

Getting Started with Informix[®] Dynamic Server[™]

Informix Dynamic Server
Informix Dynamic Server, Developer Edition
Informix Dynamic Server, Workgroup Edition

Version 7.3
February 1998
Part No. 000-4351

Published by INFORMIX® Press

Informix Software, Inc.
4100 Bohannon Drive
Menlo Park, CA 94025-1032

Copyright © 1981-1998 by Informix Software, Inc. or its 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:

Answers OnLine™; INFORMIX®; Informix®; Illustra™; C-ISAM®; DataBlade®; Dynamic Server™; Gateway™; NewEra™

All other names or marks may be registered trademarks or trademarks of their respective owners.

Documentation Team: Signe Haugen, Barbara Nomiyama, Eileen Wollam

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 of Defense use, with rights as restricted by vendor's standard license, unless superseded by negotiated vendor license as prescribed in DFAR 227.7202.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
Types of Users	4
Software Dependencies	4
Assumptions About Your Locale.	5
Demonstration Database	5
New Features	6
Documentation Conventions	6
Typographical Conventions	7
Icon Conventions	8
Additional Documentation	10
On-Line Manuals	10
Printed Manuals	10
On-Line Help	11
Error Message Files	11
Documentation Notes, Release Notes, Machine Notes	12
Related Reading	13
Compliance with Industry Standards	14
Informix Welcomes Your Comments	14

Chapter 1

Informix Dynamic Server Architecture

What Is a Relational Database Management System?	1-3
What Is Informix Dynamic Server?	1-4
Basic Database Server Architecture	1-5
The Shared-Memory Component	1-5
The Disk Component.	1-6
The Virtual Processor Component	1-7

Architectural Elements of Informix Dynamic Server	1-8
Client/Server Architecture	1-8
Scalability	1-11
High Performance	1-11
Fault Tolerance and High Availability	1-15
Multimedia Support.	1-18
Dynamic System Administration	1-19
Distributed Databases and Transactions	1-19
Database Server Security	1-21

Chapter 2 **Informix Dynamic Server Features**

Relational Database Management	2-3
System Catalog	2-4
Data Types	2-5
Data-Management Features	2-5
Data Replication	2-5
Cost-Based Optimizer	2-8
High-Performance Loader	2-9
Database Server Utilities	2-10
Data-Recovery and Data-Restore Features	2-11
The ontape Utility	2-12
On-Line Archive Recovery System (ON-Archive)	2-13
On-Line Backup and Restore (ON-Bar)	2-14
Informix Storage Manager	2-14
Tools That You Cannot Use for Recovery	2-15
Database Support	2-15
ANSI-Compliant Databases	2-15
Distributed Databases	2-16
Databases Distributed Across Multiple Vendor Servers	2-18
Dimensional Databases	2-18
Global Language Support (GLS)	2-20
Other Database Server Features	2-21
Locking	2-21
Isolation Level.	2-21
Transaction Logging.	2-22
Rolling Back Statements in a Transaction	2-22
Role Separation	2-22

Chapter 3

Getting Up and Running with the Database Server

Basic Tasks for Getting Started	3-3
Migrating to Informix Dynamic Server	3-4
Supported Interfaces and Client Products	3-4
Interfaces	3-5
Client SDK Products	3-6
Installing Informix Products	3-8
Configuring the Database Server Environment	3-9
Required Environment Variables	3-10
Connectivity Files	3-12
Preparing the ONCONFIG Configuration File	3-13
Allocating Disk Space	3-14
Initializing Disk Space	3-14
Configuring the SQL API Environment	3-15
Creating a Database	3-15
Making the Initial Connection to a Database Server	3-16
Connecting with IECC	3-16
Connecting with Other SQL APIs	3-16
Creating the Demonstration Database	3-17
Getting Error Message Information	3-18
Performing Standard Administrative Tasks	3-19

Chapter 4

Using the Documentation

Who Uses Informix Dynamic Server?	4-3
Database Users	4-3
Programmers and Application Developers	4-4
Database Administrators	4-4
Informix Dynamic Server Administrators and Operators	4-4
Task-Documentation Matrix	4-5
The Informix Dynamic Server Documentation Set	4-11
Manuals for Additional Supported Products	4-15

Index

Introduction

About This Manual.	3
Types of Users	4
Software Dependencies	4
Assumptions About Your Locale.	5
Demonstration Database	5
New Features.	6
Documentation Conventions	6
Typographical Conventions	7
Icon Conventions	8
Comment Icons	8
Cross-Reference Icons	8
Feature, Product, and Platform Icons	9
Additional Documentation	10
On-Line Manuals	10
Printed Manuals	10
On-Line Help	11
Error Message Files	11
Documentation Notes, Release Notes, Machine Notes	12
Related Reading	13
Compliance with Industry Standards	14
Informix Welcomes Your Comments	14

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

This manual provides an overview of how to work efficiently with the following Informix database servers:

Product	Platform
Informix Dynamic Server	UNIX Windows NT
Informix Dynamic Server, Workgroup and Developer Editions	Windows NT

It outlines Informix Dynamic Server architecture, introduces the major features of these database servers, and summarizes the basic tasks that are required for you to get up and running with Informix Dynamic Server and Informix Dynamic Server, Workgroup and Developer Editions. This manual also provides information to help you use the documentation that is included with this Informix Dynamic Server release.

Types of Users

This guide is for the following users:

- Database users
- Database administrators
- Database server administrators
- Database server operators
- Programmers and client application developers

This guide assumes that you have the following background:

- A working knowledge of your computer, your operating system, and the utilities that your operating system provides
- Some experience working with relational databases or exposure to database concepts
- Some experience with computer programming



If you are unfamiliar with relational database concepts or SQL, review the [Informix Guide to SQL: Tutorial](#) and the [Informix Guide to Database Design and Implementation](#). For additional titles, see [Chapter 4, “Using the Documentation.”](#)

Software Dependencies

This manual assumes that your database server is one of the following products:

- Informix Dynamic Server, Version 7.3
- Informix Dynamic Server, Developer Edition, Version 7.3.
- Informix Dynamic Server, Workgroup Edition, Version 7.3.

In this manual, Dynamic Server refers to Informix Dynamic Server, or Informix Dynamic Server, Workgroup and Developer Editions, as appropriate.

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 Global Language Support (GLS) locale.

This manual assumes that you use the U.S. 8859-1 English locale as the default locale. The default is **en_us.8859-1** (ISO 8859-1) on UNIX platforms or **en_us.1252** (Microsoft **1252**) for Windows NT environments. This locale supports U.S. English format conventions for dates, times, and currency, and also supports the ISO 8859-1 or Microsoft **1252** code set, which includes the ASCII code set plus many 8-bit characters such as é, è, and ñ.

If you plan to use non-English characters in your data or your SQL identifiers, or if you want to conform to the non-English collation rules of character data, you need to specify the appropriate nondefault locales.



For instructions on how to specify a nondefault locale, additional syntax, and other considerations related to GLS locales, see the [Informix Guide to GLS Functionality](#).

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. You can use SQL scripts provided with DB-Access to derive a second database, called **sales_demo**. This database illustrates a dimensional schema for data-warehousing applications. Sample command files are also included for creating and populating these databases.

Many examples in Informix manuals are based on the **stores7** demonstration database. The **stores7** database is described in detail and its contents are listed in the [Informix Guide to SQL: Reference](#).



The scripts that you use to install the demonstration databases reside in the **\$INFORMIXDIR/bin** directory on UNIX platforms and the **%INFORMIXDIR%\bin** directory on Windows NT platforms. For a complete explanation of how to create and populate the **stores7** demonstration database, refer to the [DB-Access User Manual](#). For an explanation of how to create and populate the **sales_demo** database, refer to the [Informix Guide to Database Design and Implementation](#).

New Features

Most of the new features for Version 7.3 of Informix Dynamic Server fall into five major areas:

- Reliability, availability, and serviceability
- Performance
- NT-specific features
- Application migration
- Manageability

Several additional features affect connectivity, replication, and the optical subsystem. For a comprehensive list of new features, see the release notes 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

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.






***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.






Cross-Reference Icons

Cross-reference icons indicate paragraphs that show where you can find more information about a topic.

Icon	Description
	Identifies paragraphs that contain cross-references to other Informix manuals that provide additional information on a topic.

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 relates to the Informix GLS feature.
	Identifies information that is specific to Dynamic Server and its editions. However, in some cases, the identified section applies only to Informix Dynamic Server and not to Informix Dynamic Server, Workgroup and Developer Editions. Such information is clearly identified.
	Identifies information that is specific to the UNIX platform.
	Identifies information that is specific to Informix Dynamic Server, Workgroup and Developer Editions.
	Identifies information that is specific to the Windows NT environment.

These icons can apply to a row in a table, one or more paragraphs, or an entire section. If an icon appears next to a section heading, the information that applies to the indicated feature, product, or platform ends at the next heading at the same or higher level. A ♦ symbol indicates the end of the feature-, product-, or platform-specific information that appears within a table or a set of paragraphs within a section.

Additional Documentation

For additional information, you can 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
- Related reading

On-Line Manuals

An Answers OnLine 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 the installation insert that accompanies Answers OnLine.

For an overview of the manuals in the Informix Dynamic Server documentation set, see [Chapter 4, “Using the Documentation.”](#)

Printed Manuals

To order printed manuals, call 1-800-331-1763 or send email to moreinfo@informix.com. Please provide the following information when you place your order.

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

For an overview of the manuals in the Informix Dynamic Server documentation set, see [Chapter 4, “Using the Documentation.”](#)

On-Line Help

The DB-Access utility provides on-line help. Press CTRL-W to display a help screen that has information about the current menu option or screen function.

WIN NT

Informix provides Help screens with each graphical user interface (GUI) that display information about the interfaces and the functions that they perform. To display these Help screens, use the Help facilities provided with each GUI. ♦

Error Message Files

Informix software products provide ASCII files that contain all of the Informix error messages and their corrective actions. For a detailed description of these error messages, refer to *Informix Error Messages* in Answers OnLine.

UNIX

To read the error messages under UNIX, you can use the following commands.

Command	Description
finderr	Displays error messages on line
rofferr	Formats error messages for printing

♦

WIN NT

To read error messages and corrective actions under Windows NT, use the **Informix Find Error** utility. To display this utility, choose **Start→Programs→Informix** from the Task Bar. ♦

UNIX

Documentation Notes, Release Notes, Machine Notes

In addition to printed documentation, the following sections describe the on-line files that supplement the information in this manual. Please examine these files before you begin using your database server. They contain vital information about application and performance issues.

On UNIX platforms, the following on-line files appear in the **SINFORMIXDIR/release/en_us/0333** directory.

On-Line File	Purpose
STARTDOC_7.3	The documentation-notes file for your version of this manual describes features that are not covered in this manual or that have been modified since publication.
SERVERS_7.3	The release-notes 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.
IDS_7.3	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.



WIN NT

The following items appear in the Informix folder. To display this folder, choose **Start→Programs→Informix** from the Task Bar.

Item	Description
Documentation Notes	This item includes additions or corrections to manuals, along with information about features that may not be covered in the manuals or that have been modified since publication.
Release Notes	This item 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.

Machine notes do not apply to Windows NT platforms. ♦

Related Reading

The following publications provide additional information about the topics that are discussed in this manual. For a list of publications that provide an introduction to database servers and operating-system platforms, refer to your *Getting Started* manual.

- *A Guide to the SQL Standard* by C. J. Date with H. Darwen (Addison-Wesley Publishing, 1993)
- *Understanding the New SQL: A Complete Guide* by J. Melton and A. Simon (Morgan Kaufmann Publishers, 1993)
- *Using SQL* by J. Groff and P. Weinberg (Osborne McGraw-Hill, 1990)

UNIX

This manual assumes that you are familiar with your computer operating system. If you have limited UNIX system experience, consult your operating-system manual or a good introductory text before you read this manual. The following texts provide a good introduction to UNIX systems:

- *Introducing the UNIX System* by H. McGilton and R. Morgan (McGraw-Hill Book Company, 1983)
- *Learning the UNIX Operating System* by G. Todino, J. Strang, and J. Peek (O'Reilly & Associates, 1993)
- *A Practical Guide to the UNIX System* by M. Sobell (Benjamin/Cummings Publishing, 1989)
- *UNIX System V: A Practical Guide* by M. Sobell (Benjamin/Cummings Publishing, 1995) ♦

WIN NT

If you are using Windows NT and have limited Windows NT system experience, consult your operating-system manual or a good introductory text before you read this manual. The following texts provide an introduction to Windows NT:

- *Using Windows NT Workstation 3.51* by Paul Sanna (Que, 1996)
- *Microsoft Windows NT Resource Kit* by Russ Blake (Microsoft Press, 1995)
- *NT Server Management and Control* by Kenneth L. Spencer (Prentice-Hall, 1995)

- *Windows NT Administration* by Marshall Brain and Shay Woodard (Prentice-Hall, 1994)
- *Windows NT Network Programming* by Ralph Davis (Addison-Wesley, 1994)
- *Inside Windows NT* by Helen Custer (Microsoft Press, 1993) ♦

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. In addition, many features of Informix database servers comply with the SQL-92 Intermediate and Full Level and X/Open SQL CAE (common applications environment) standards.

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
- The page number and part number would be helpful
- 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:

650-926-6571

We appreciate your feedback.

Informix Dynamic Server Architecture

What Is a Relational Database Management System?	1-3
What Is Informix Dynamic Server?	1-4
Basic Database Server Architecture	1-5
The Shared-Memory Component	1-5
The Disk Component.	1-6
The Virtual Processor Component	1-7
Architectural Elements of Informix Dynamic Server	1-8
Client/Server Architecture	1-8
Supported Connection Types.	1-9
Supported Client/Server Configurations	1-10
Scalability.	1-11
High Performance.	1-11
Raw (Unbuffered) Disk Management	1-12
Memory Management	1-12
Dynamic Thread Allocation	1-13
Fragmentation.	1-13
Parallelization	1-14
Fault Tolerance and High Availability	1-15
Dbpace and Logical-Log Backups of Transaction Records	1-15
Fast Recovery	1-16
Mirroring	1-16
High-Availability Data Replication.	1-17
Point-in-Time Recovery	1-18
Multimedia Support	1-18
Dynamic System Administration.	1-19
Distributed Databases and Transactions	1-19
Two-Phase Commit	1-20
Heterogeneous Commit	1-20
Database Server Security	1-21



This chapter introduces Informix Dynamic Server architecture. For information about specific features, see [Chapter 2, “Informix Dynamic Server Features.”](#)

For complete details of Informix Dynamic Server architecture, see the [Administrator’s Guide](#).

What Is a Relational Database Management System?

A *relational database management system* (RDBMS) includes all the software you need to create and maintain a relational database. A relational database consists of tables that are made up of rows and columns.

An Informix RDBMS includes the following components:

- A database server
- A database
- One or more client applications

This chapter discusses the architecture of Informix database servers. For information about Informix Dynamic Server features and supported databases, see [Chapter 2, “Informix Dynamic Server Features.”](#) For information about supported Informix products and client applications, see [“Supported Interfaces and Client Products” on page 3-4.](#)



For information about relational database management, see the [Informix Guide to Database Design and Implementation](#) and the [Informix Guide to SQL](#) set of manuals.

What Is Informix Dynamic Server?

Informix Dynamic Server is a *database server*. A database server is a software package that manages access to one or more databases for one or more client applications.

Specifically, Informix Dynamic Server is a multithreaded *relational database server* that manages data that is stored in rows and columns. It employs a single processor or *symmetric multiprocessor* (SMP) systems and dynamic scalable architecture (DSA) to deliver database scalability, manageability and performance.

Informix Dynamic Server, Version 7.3, provides a number of configuration options. This manual discusses only the following implementations:

- Informix Dynamic Server
This configuration is typically used for on-line transaction processing (OLTP), packaged applications, modest data-mart and data-warehousing applications, and web solutions.
- Informix Dynamic Server, Workgroup Edition
This configuration is typically used for branch automation and value-added reseller (VAR) solutions.
- Informix Dynamic Server, Developer Edition
This single-user workstation configuration is typically used by application developers.

W/D

Informix Dynamic Server, Workgroup and Developer Editions, generally provide the same features and functionality as the core Informix Dynamic Server, except that they do not support:

- fragmentation.
- parallel database query (PDQ).
- role separation.
- full Enterprise Replication.

In addition, the Developer Edition might be more restricted than the Workgroup Edition. ♦

Basic Database Server Architecture

Basic Informix database server architecture consists of the following three main components:

- Shared memory
- Disk
- Virtual processor

These components are described briefly in this section.

For complete descriptions of the terms and concepts discussed in this section, see the [Administrator's Guide](#).



The Shared-Memory Component

Shared memory is an operating-system feature that lets the database server threads and processes share data by sharing access to pools of memory. The database server uses shared memory for the following purposes:

- To reduce memory use and disk I/O
- To perform high-speed communication between processes

Shared memory lets the database server reduce overall memory uses because the participating processes—in this case, virtual processors—do not need to maintain individual copies of the data that is in shared memory.

Shared memory reduces disk I/O because buffers, which are managed as a common pool, are flushed on a database server-wide basis instead of on a per-process basis. Furthermore, a virtual processor can often avoid reading data from disk because the data is already in shared memory as a result of an earlier read operation. The reduction in disk I/O reduces execution time.

Shared memory provides the fastest method of interprocess communication because processes read and write messages at the speed of memory transfers.

The database server uses two types of shared memory:

- Resident
Caches data from the disk for faster access
- Virtual
Maintains and controls the resources required by the virtual processors

The Disk Component

A disk is a collection of one or more units of disk space assigned to the database server. All the data in the databases and all the system information that is necessary to maintain the database server resides within the disk component.

The database server uses the following types of disk access to store data:

- Buffered
- Unbuffered

The database server uses the following physical units to manage disk space:

- Chunk
- Page
- Blobpage
- Extent

Overlying the physical units of storage space, the database server supports the following logical units that are associated with database management:

- Dbspace
- Blobspace
- Extspace
- Database
- Table
- Tblspace

UNIX

The database server maintains the following additional disk-space storage structures to ensure physical and logical data consistency:

- Logical log
- Physical log
- Reserved pages

The UNIX operating system provides two distinct types of disk space: raw and cooked. The database server allows you to use either type of disk space or a combination of both types.

- *Raw disk space* is unformatted space where the database server manages the physical organization of the data. Raw files are more reliable than cooked files because the database server guarantees that committed data is stored on disk.
- *Cooked disk space* refers to regular operating-system files. It is space that has already been organized and that the UNIX operating system manages. Cooked files are easier to allocate than raw disk space. ♦

The Virtual Processor Component

The central component of Informix DSA is the *virtual processor*, which is a database server process that the operating system schedules for execution on the CPU. Database server processes are called virtual processors because they function similarly to a CPU in a computer. Just as a CPU runs multiple operating-system processes to service multiple users, a virtual processor runs multiple *threads* to service multiple client applications.

A thread is a task for a virtual processor in the same way that the virtual processor is a task for the CPU. How the database server processes a thread depends on the operating system. Virtual processors are *multithreaded processes* because they run multiple concurrent threads.

Architectural Elements of Informix Dynamic Server

This section describes the following architectural elements of Informix Dynamic Server:

- Client/server architecture ([page 1-8](#))
- Scalability ([page 1-11](#))
- High performance ([page 1-11](#))
- Fault tolerance and high availability ([page 1-15](#))
- Multimedia support ([page 1-18](#))
- Dynamic system administration ([page 1-19](#))
- Distributed databases and transactions ([page 1-19](#))
- Database server security ([page 1-21](#))

Client/Server Architecture

Informix client applications and database servers conform to a model of software design called *client/server*. Client/server functionality handles all the connections between the client application and the database server.

A *client* is an application program that requests or modifies information from a database, usually by issuing Structured Query Language (SQL) statements. When the client connects to the database server through an SQL statement, the client transparently accesses **sqlhosts** connectivity information that is contained in a file on UNIX or in the Windows NT registry. (See “[Preparing the sqlhosts File](#)” on [page 3-12](#).)

Client programs include the DB-Access utility, the SQL Editor utility within the Informix Enterprise Command Center (IECC), and programs that you write with one of the application-programming interfaces (API) that is included with INFORMIX-4GL, INFORMIX-NewEra, INFORMIX-Client Software Developer’s Kit, and so on. For details, see your API documentation.

Informix Dynamic Server processes requests for data from client applications, accesses the requested information from its databases, and returns the results to the client. Database-access activities include coordination of concurrent requests from multiple clients, read and write operations, and enforcement of physical and logical data consistency.

The client applications of an Informix RDBMS do not need to be modified to run on a network. The communications tools that are part of all Informix products handle the tasks of locating and attaching to the database servers. To the application, a database on a networked computer appears no different than a database on the computer where the application resides.



For information about how to define connectivity information in the `sqlhosts` file or registry, see the [Administrator's Guide](#). For information on the SQL statements that the database server supports, see the [Informix Guide to SQL: Syntax](#). For detailed instruction in how to use an SQL API, see the appropriate programmer's manual, as listed in [Chapter 4](#).

Supported Connection Types

Informix Dynamic Server supports the types of connections shown in the following table.

Communication Mechanism or Network Protocol	Interface
TCP/IP (local and remote)	Transport level interface (TLI) or sockets
IPX/SPX (local and remote)	TLI
TCP/IP	Sockets ♦
Interprocess communication (local only)	Shared memory or stream pipes Unnamed pipes ♦

Important: *Not all platforms support both the TCP/IP and IPX/SPX protocols. To check which interface/protocol combinations your database server supports for your operating system, read the Machine Notes file.*

For more information about supported connectivity protocols and how to set up local and network connections for Informix database servers and client applications, see the [Administrator's Guide](#).



Supported Client/Server Configurations

Although a typical client/server configuration places a client application on one computer and a database server on another, a client application and database server can also reside on the same computer. The client/server environment is often a multiuser environment, with several clients accessing the same database server.

Informix Dynamic Server supports the following types of client/server configurations:

- **Single user**

The application and database server reside on the same computer. You would typically find this type of configuration on a personal computer that runs Windows NT.

- **Local multiuser**

A multiuser system with a local database server is typically one you might find on a UNIX system. An example of a local multiuser system might be two or more INFORMIX-ESQL/C client applications that use the same database server.

- **Remote or network**

The client application resides on one computer and the database server and its associated databases reside on another computer on the network.

- **Local loopback**

This configuration behaves as if multiple sites reside on one computer. All the components are on the same computer, but the connections are made as if they were connected through a network. Because the connections use network software, the database server appears as a remote site to the client application.

Each type of client/server configuration has its advantages and disadvantages. Considerations include ease of use, data availability, distributed transaction processing, shared memory, response time, and so on.

For more information about the client/server configurations that the database server supports, see the [*Administrator's Guide*](#).



Scalability

Informix Dynamic Server lets you scale resources in relation to the demands that applications place on the database server. Dynamic scalable architecture provides the following performance advantages for both single-processor and multiprocessor platforms:

- A small number of database server processes can service a large number of client application processes, producing the following benefits:
 - Reduced operating-system overhead (fewer processes to run)
 - Reduced overall memory requirements
 - Reduced contention for resources within the RDBMS
- DSA provides more control over setting priorities and scheduling database tasks than the operating system does.

Informix Dynamic Server employs single-processor or symmetric multiprocessor computer systems. In an SMP computer system, multiple central processing units (CPUs) or processors all run a single copy of the operating system, sharing memory and communicating with each other as necessary.



For detailed information on dynamic scalable architecture, see the [Administrator's Guide](#). For information on performance tuning, see the [Performance Guide](#).

High Performance

Informix Dynamic Server achieves high performance through the following mechanisms:

- Raw (unbuffered) disk management
- Memory management
- Dynamic thread allocation
- Fragmentation
- Parallelization

Informix Dynamic Server, Workgroup and Developer Editions, do not support fragmentation and parallelization. ♦

Raw (Unbuffered) Disk Management

Informix Dynamic Server can use both file-system disk space and raw disk space in UNIX and Windows NT environments. When the database server uses raw disk space, it performs its own disk management using raw devices. By storing tables on one or more raw devices instead of in a standard operating-system file system, the database server can manage the physical organization of data and minimize disk I/O. Doing so results in three performance advantages:

- No restrictions due to operating-system limits on the number of tables that can be accessed concurrently.
- Optimization of table access by guaranteeing that rows are stored contiguously.
- Elimination of operating-system I/O overhead by performing direct data transfer between disk and shared memory.

If these issues are not a primary concern, you can also configure the database server to use regular operating-system files to store data. In this case, Informix Dynamic Server manages the file contents, but the operating system manages the I/O.



For information on disk management, see the [Administrator's Guide](#).

Memory Management

Informix Dynamic Server provides two main options to help you manage memory to optimize performance.

Dynamically Sharing Memory

All applications that use a single instance of Informix Dynamic Server share data in the memory space of the database server. After one application reads data from a table, other applications can access whatever data is already in memory. Disk access, and the corresponding degradation in performance, might not occur because the database server adds memory dynamically as it needs it. The database server administrator can also add segments to shared memory when necessary.

Buffering Transactions

You can determine where transactions are logged in Informix Dynamic Server. Your logs can be buffered or unbuffered. Buffered logging holds transactions in memory until the buffer is full, regardless of when the transaction is committed.



For information on managing the various aspects of memory to increase performance, see the [Performance Guide](#).

Dynamic Thread Allocation

Informix Dynamic Server supports multiple client applications using a relatively low number of processes called *virtual processors*. A virtual processor is a multithreaded process that can serve multiple clients and, where necessary, run multiple threads to work in parallel for a single query. In this way, the database server provides a flexible architecture that is well suited for both on-line transaction processing (OLTP) and decision-support system (DSS) applications.



For information on how virtual processors can dynamically allocate threads, see the [Administrator's Guide](#).

Fragmentation

Informix Dynamic Server supports table and index *fragmentation* over multiple disks. Fragmentation lets you group rows within a table according to a distribution scheme and improve performance on very large databases.

W/D

Informix Dynamic Server, Workgroup and Developer Editions, do not support fragmentation. ♦

The database server stores the rows in separate database spaces (*dbspaces*) that you specify in a fragmentation strategy. A dbspace is a logical collection of one or more database server *chunks*. Chunks represent specific regions of disk space.

Informix Dynamic Server supports two types of distribution scheme.

- *Round-robin fragmentation* places rows one after another in fragments, rotating through the series of fragments to distribute the rows evenly.
- *Expression-based fragmentation* puts rows that contain specified values in the same fragment. You specify a fragmentation expression that defines criteria for assigning a set of rows to each fragment, either as a range rule or some arbitrary rule.

Use the CREATE TABLE statement to create a fragmented table and the CREATE INDEX statement to create a fragmentation strategy for an index.

Table fragmentation can improve:

- single-user response time.
- concurrency.
- availability.
- backups and restores.



For information on how to formulate a fragmentation strategy and manage fragmentation, see the [Administrator's Guide](#) and the [Informix Guide to Database Design and Implementation](#). For information on how fragmentation can enhance performance, see the [Performance Guide](#). For information about the CREATE TABLE and CREATE INDEX statements, see the [Informix Guide to SQL: Syntax](#).

Parallelization

Informix Dynamic Server can allocate multiple threads to work in parallel on a single query. This feature is known as the parallel database query (PDQ) feature.

Informix Dynamic Server, Workgroup and Developer Editions, do not support parallelization. ♦

PDQ can improve performance dramatically when the database server processes queries that are initiated by decision-support applications. PDQ lets the database server distribute the work for one aspect of a query among several processors.

W/D



Use the SET PDQPRIORITY statement to establish PDQ. PDQ is most effective when you use it with fragmentation.

For information on how to implement PDQ, see the [Administrator's Guide](#). For information on how parallelization can enhance performance, see the [Performance Guide](#). For a description of the SET PDQPRIORITY statement, see the [Informix Guide to SQL: Syntax](#).

Fault Tolerance and High Availability

Informix Dynamic Server uses the following logging and recovery mechanisms to protect data integrity and consistency in the event of an operating-system or media failure:

- Dbspace and logical-log backups of transaction records
- Fast recovery
- Mirroring
- High-availability data replication
- Point-in-time recovery



For information on fault tolerance and high availability, see the [Administrator's Guide](#).

Dbspace and Logical-Log Backups of Transaction Records

Informix Dynamic Server lets you back up the data that it manages and also store changes to the database server and data since the backup was performed. The changes are stored in *logical-log files*.

You can create backup tapes and logical-log backup tapes while users are accessing the database server. You can also use on-line archiving to create incremental backups. Incremental backups let you back up only data that has changed since the last backup, which reduces the amount of time that a backup would otherwise require.

After a media failure, if critical data was not damaged (and Informix Dynamic Server remains on-line), you can restore only the data that was on the failed media, leaving other data available during the restore.



For information about logical and physical logs, see the [Administrator's Guide](#). For information about backing up data, see the [Archive and Backup Guide](#) if you use ON-Archive, or the [Backup and Restore Guide](#) if you use ON-Bar.

Fast Recovery

When the database server starts up, it checks the *physical log*, which contains pages that have not yet been written to disk. If the physical log is empty, that implies that the database server was shut down in a controlled fashion. If the physical log is *not* empty, Informix Dynamic Server automatically performs an operation called *fast recovery*.

Fast recovery automatically restores Informix Dynamic Server databases to a state of physical and logical consistency after a system failure that might have left one or more transactions uncommitted. During fast recovery, the database server spawns multiple threads to work in parallel and uses its logical log and physical log to perform the following operations:

- Restore the databases to their state at the last checkpoint
- Roll forward all committed transactions since the last checkpoint
- Roll back any uncommitted transactions



For information about fast recovery, see the [Administrator's Guide](#).

Mirroring

When you use disk mirroring, the database server writes data to two locations. Mirroring eliminates data losses due to media (hardware) failures. If mirrored data becomes unavailable for any reason, the mirror of the data is made available immediately and transparently to users.

Mirroring is a strategy that pairs a *primary chunk* of one defined dbspace with an equal-sized *mirrored chunk*. Every write to the primary chunk is automatically accompanied by an identical write to the mirrored chunk. If a failure occurs on the primary chunk, mirroring lets you read from and write to the mirrored chunk until you can recover the primary chunk, all without interrupting user access to data.

Ideally, you should mirror all of your data. However, if cost is a problem, you should select certain critical chunks to mirror. For example:

- The logical-log files contain critical information and should be mirrored for maximum data protection.
- If you move logical-log files to a different dbspace, start mirroring that dbspace. If you do not mirror your dbspaces, you have to restore much more frequently from a dbspace backup in the event of a media failure.

The operating system or hardware that you use might provide an alternative form of mirroring. If you consider a mirroring option provided by your operating system instead of by the database server, compare the implementation of both options before you decide which to use. Operating-system mirroring options that do not use parallel mirror writes and split reads might provide inferior performance to mirroring by the database server.

Database server mirroring and operating-system mirroring run independently and can run at the same time. For example, you might have both database server data and nondatabase server data on a single disk drive. You could use operating-system mirroring to mirror nondatabase server data and database server mirroring to mirror database server data.

Logical-volume managers and hardware mirroring are other alternative mirroring solutions. Saving data to more than two disks with logical-volume managers gives you added protection from media failure, but the additional writes have a performance cost. Hardware mirroring such as RAID (redundant array of independent disks) has the advantage of requiring less disk space to store the same amount of data than does Informix database server mirroring, but it is slower for write operations.

For information about mirroring, see the [*Administrator's Guide*](#).



High-Availability Data Replication

If your organization requires a high degree of availability, you can replicate Informix Dynamic Server and its databases, running simultaneously on a second computer. Replication of the database server and its databases provides a backup system in case of a catastrophic failure. If one site experiences a disaster, you can immediately direct applications to use the second database server in the pair.



Running data replication also allows you to balance read-only applications (such as DSS applications) across both database servers in the data-replication pair.

For information about high-availability data replication, see the [Administrator's Guide](#).

Point-in-Time Recovery

Use point-in-time recovery after a catastrophic event to restore the data in a database to a specific time, perhaps immediately preceding the catastrophic event. A point-in-time recovery can undo mistakes, such as dropping a table, that might not be fixable otherwise.

When you restore the database server to a specific time, some transactions might be lost even though they are included in an existing logical-log backup. Because the database server can only be restored to the last known global point of consistency across all database servers, transactions that were committed after the specified recovery point cannot be retrieved.



For information about point-in-time recovery and data restoration, see the [Archive and Backup Guide](#) if you use ON-Archive, or the [Backup and Restore Guide](#) if you use ON-Bar.

Multimedia Support

Informix Dynamic Server supports the TEXT and BYTE data types that place no practical limit on the size of the stored data item. This type of data is stored either with other database data or in specially designated portions of the disk called blobspaces.



For information about supported data types and how to use them, see the [Informix Guide to SQL: Reference](#) and the [Informix Guide to Database Design and Implementation](#). For information about storing TEXT and BYTE data on write-once-read-many (WORM) optical devices with Informix Dynamic Server, see the [Guide to the Optical Subsystem](#). For information about additional multimedia data types such as image, audio, video, and user defined, see the documentation for Informix Dynamic Server with Universal Data Option.

Dynamic System Administration

Distributed databases require dynamic system administration tools that monitor and fine tune the following system parameters:

- CPU and memory utilization
- Asynchronous I/O
- Decision support and batch queuing
- Available disk space
- Efficient partitioning schemes

The database server provides various administration tools to help you perform these tasks. Tools include the system monitoring interface (SMI), memory grant manager, **onperf** utility, and Informix Enterprise Command Center (IECC).



For information on SMI, see the [Administrator's Guide](#). For information on the memory grant manager and the **onperf** utility, see the [Performance Guide](#). For information on IECC, see the [Informix Enterprise Command Center User Guide](#) or use the IECC help feature.

Distributed Databases and Transactions

Informix Dynamic Server lets you query (and update) more than one database across multiple database servers within a single transaction. This ability is called *distributed database processing*.

Distributed databases are useful because operations that access databases are often distributed into separate pieces, either organizationally, geographically, or both. Distributed databases provide the following advantages:

- Local data can be kept where it is more easily maintained and is more frequently used.
- Data from remote sites is available to all users.

Informix Dynamic Server lets you perform transactions on data from databases that reside on different database servers connected across a network. The database server supports two multiphase protocols, two-phase commit and heterogeneous commit, to process transactions that span multiple database servers.

Two-Phase Commit

A two-phase commit protocol ensures that transactions are uniformly committed or rolled back across the multiple database servers. This protocol governs the order in which a two-phase commit transaction is performed and provides an automatic recovery mechanism in case a system or media failure occurs during execution of the transaction.

A database server automatically uses the two-phase commit protocol for any transaction that performs modifications to data on more than one database server. The database server uses logical-log records to implement the two-phase commit protocol. You can use these logical-log records to detect heuristic decisions and, if necessary, to help you perform a manual recovery.

Heterogeneous Commit

In the context of Informix database servers, the term *heterogeneous environment* refers to a group of database servers in which at least one is not an Informix database server. Heterogeneous commit is a database server feature that ensures the all-or-nothing basis of distributed transactions in a heterogeneous environment.

Unlike the two-phase commit protocol, the heterogeneous commit protocol supports the participation of a non-Informix participant. The non-Informix participant, called a gateway participant, must communicate with the coordinator through an INFORMIX-Enterprise Gateway product.



For information about two-phase commit and heterogeneous commit protocols and concepts and the use of logical-log records in distributed databases and transactions, see the [Administrator's Guide](#). For information about a specific INFORMIX-Enterprise Gateway product, see the appropriate *INFORMIX-Enterprise Gateway User Manual*.

Database Server Security

Informix Dynamic Server provides the following security features:

- Database-level security
- Table-level security
- Role creation

W/D

Informix Dynamic Server, Workgroup and Developer Editions, do not support role creation and the CREATE ROLE statement. ♦

The databases and tables that the database server manages enforce access based on a set of database and table privileges. You can use the following SQL statements to manage these privileges:

- Use the GRANT and REVOKE statements to give or deny access to a database or specific tables and to control the kinds of database uses.
- Use the CREATE PROCEDURE statement to write and compile a stored procedure that controls and monitors access to tables.
- Use the CREATE VIEW statement to prepare a restricted or modified view of the data.
- Combine the GRANT and CREATE VIEW statements to precisely control the parts of a table that a user can modify.
- Use the CREATE ROLE statement to set up classifications with privileges on database objects granted to a specific role.

Informix Dynamic Server also lets you audit database events on a database server-wide basis. Auditing allows you to track which users performed specific actions to particular objects at distinct times. You can use this information to monitor database activity for suspicious use, deter unscrupulous users, or even act as evidence of database server abuse.

UNIX

Informix database servers follow UNIX security requirements for making connections. Thus, the UNIX system administrator might need to make modifications to the `/etc/passwd`, `/etc/hosts`, `~/rhosts`, and other related files. ♦



For information about database and table privileges, see the [Informix Guide to SQL: Tutorial](#). For the syntax and description of SQL statements, see the [Informix Guide to SQL: Syntax](#). For information about controlling access to databases, see the [Informix Guide to Database Design and Implementation](#). For information on database server security, see the [Administrator's Guide](#). For information on auditing, see the [Trusted Facility Manual](#).

Informix Dynamic Server Features

Relational Database Management	2-3
System Catalog	2-4
Data Types	2-5
Data-Management Features	2-5
Data Replication	2-5
High-Availability Data Replication.	2-6
Enterprise Replication	2-7
Cost-Based Optimizer	2-8
High-Performance Loader	2-9
Database Server Utilities	2-10
Data-Recovery and Data-Restore Features	2-11
The ontape Utility	2-12
On-Line Archive Recovery System (ON-Archive)	2-13
On-Line Backup and Restore (ON-Bar)	2-14
Informix Storage Manager	2-14
Tools That You Cannot Use for Recovery	2-15
Database Support	2-15
ANSI-Compliant Databases	2-15
Distributed Databases	2-16
Databases Distributed Across Multiple Vendor Servers	2-18
Dimensional Databases	2-18
Global Language Support (GLS)	2-20
Other Database Server Features	2-21
Locking	2-21
Isolation Level	2-21
Transaction Logging	2-22
Rolling Back Statements in a Transaction	2-22
Role Separation.	2-22



This chapter provides a brief overview of relational database management and touches on the significant features of Informix Dynamic Server.

For more information about database server features, see the [Administrator's Guide](#).

Relational Database Management

An Informix RDBMS consists of a database server, a database, and one or more client applications. This chapter discusses the first two components. For information on the third, see “[Supported Interfaces and Client Products](#)” on [page 3-4](#).

Informix Dynamic Server works with *relational databases*. A relational database lets you store data so that the data is perceived as a series of rows and columns. This series of rows and columns is called a table, and a group of tables is called a database. (An edition of the database server that is not discussed in this manual works with object relational databases.)

SQL statements direct all operations on a database. The client application interacts with you, prepares and formats data, and uses SQL statements to send data requests to the database server. The database server interprets and executes SQL statements to manage the database and return data to the client application. You can use SQL statements to retrieve, insert, update, and delete data from a database.

To retrieve data from a database, you perform a query. A query is a `SELECT` statement that specifies the rows and columns to be retrieved from the database.



An RDBMS permits high-speed, short-running queries and transactions on the following types of data:

- Integer
- Floating-point number
- Character string, fixed or variable length
- Date and time, time interval
- Numeric and decimal
- Complex data stored in objects

For information about relational database management and choosing data types for your database, see the [Informix Guide to Database Design and Implementation](#). For information about SQL, see the [Informix Guide to SQL: Tutorial](#) and [Informix Guide to SQL: Syntax](#). For information about supported data types, see the [Informix Guide to SQL: Reference](#).

System Catalog

The *system catalog* consists of tables that describe the structure of the relational database. The system catalog tables are generated automatically when you create a database. Each system catalog table contains specific information about an element in the database.

System catalog tables track the following objects:

- Tables and constraints
- Views
- Triggers
- Authorized users and privileges that are associated with tables
- Stored procedures



For information about the structure and use of the system catalog, see the [Informix Guide to SQL: Reference](#).

Data Types

Every column in a table in a relational database is assigned a data type, which precisely defines the kinds of values that you can store in that column. Informix database servers support numerous data types and various kinds of data type conversion.



For a description of the data types and data type conversions that Informix database servers support, see the [Informix Guide to SQL: Reference](#).

Data-Management Features

Informix Dynamic Server includes or works with the following data management tools and utilities:

- Data replication
- Cost-based optimizer
- High-Performance Loader
- Administration utilities

Data Replication

Data replication is the process of representing database objects at more than one distinct site. For example, one way to replicate data is to copy a database to a database server installed on a different computer. This copy allows reports to access the data without disturbing client applications that use the original database.

Informix Dynamic Server, Version 7.3, supports two kinds of data replication:

- High-availability data replication
- Enterprise replication

Advantages of data replication are as follows:

- Clients at the site to which the data is replicated experience improved performance because those clients can access data locally rather than connecting to a remote database server over a network.
- Clients at all sites experience improved availability of replicated data. If the local copy of the replicated data is unavailable, clients can still access the remote copy of the data.

These advantages do not come without a cost. Data replication obviously requires more storage, and updating replicated data can take more processing time than updating a single object.

High-Availability Data Replication

As introduced on [page 1-17](#), Informix high-availability data replication provides a way to maintain a backup copy of an entire database server instance that applications can access quickly in the event of a catastrophic failure. The database server implements nearly transparent data replication of entire database servers. All the data managed by one database server is replicated and dynamically updated on another database server, often at a separate geographical location.

You can configure high-availability data replication to be synchronous or asynchronous. Both database servers must be on the same hardware platform and have the same database layout.

For more information about high-availability data replication, see the [Administrator's Guide](#).



Enterprise Replication

The Informix Enterprise Replication feature of Informix Dynamic Server, Version 7.3, provides a cost-effective, efficient means to replicate data at the table level throughout an open-systems enterprise. Use Enterprise Replication to replicate a table or subset of a table (such as a set of rows or columns) instead of an entire database server instance.

Enterprise Replication is a client/server application that lets you asynchronously replicate data between an unlimited number of database servers and between heterogeneous hardware platforms throughout your enterprise, including UNIX and Windows NT. A graphical user interface (GUI) helps you easily define, monitor, and control your replication system.

W/D

Informix Dynamic Server, Workgroup and Developer Editions, do not provide full Enterprise Replication functionality because the GUI offers fewer options. For example, these editions support the publish and subscribe features, but not full update-anywhere. ♦

The key advantages of Enterprise Replication are as follows:

- **High performance**
Log-based transaction capture and a parallel distribution mechanism ensure that the replication system does not burden the data source and that it uses networks and all resources efficiently.
- **High availability of data**
Replication of a central database on a single secondary database server maximizes availability and fault tolerance.
- **Consistent information delivery**
Enterprise Replication maintains multiple copies of data at different sites in a distributed environment. Transactions are stored to maintain transactional consistency and ordering.
- **Flexible architecture**
Enterprise Replication supports a full range of ownership models to meet a wide spectrum of business and application requirements.



- Centralized administration

A single integrated point of administration lets the DBA graphically configure and monitor the replicated system.

- Heterogeneous data-source support

Enterprise Replication provides bidirectional heterogeneous replication from non-Informix sources to Informix Dynamic Server.

For more information about Enterprise Replication, see the [Guide to Informix Enterprise Replication](#).

Cost-Based Optimizer

Before a query is executed, the query optimizer formulates an execution plan based on the lowest resource-use cost to fetch the data rows that are required to process a query.

The optimizer evaluates the different ways in which a query might be performed. For example, the optimizer determines whether one or multiple indexes should be used. If the query includes a join, the optimizer determines the join method (hash, sort-merge, or nested loop) and the order in which tables are evaluated or joined. The optimizer assesses each query plan on a cost basis and selects the plan with the lowest execution cost.

Hash joins can improve query execution speed when tables are joined on arbitrary, unindexed columns. In a hash join, one table is scanned to create a hash table based on the join key. Subsequent tables are read once to look up each row in the hash table to find out if a join can be made. Hash joins are more efficient than sort-merge joins because nothing is sorted. Eliminating sorting makes hash joins particularly efficient for large tables.



For more information on the cost-based optimizer, see the [Performance Guide](#).

W/D

High-Performance Loader

The High-Performance Loader (HPL) is an Informix Dynamic Server feature that lets you efficiently load and unload very large quantities of data to or from an Informix database. Use the HPL to exchange data with tapes, data files, and programs and convert data from these sources into a format compatible with an Informix database. The HPL also lets you manipulate and filter the data as you perform load and unload operations.

Informix Dynamic Server, Developer Edition, does not support the High-Performance Loader (HPL) although the Workgroup Edition does. ♦

The HPL has the following components:

- The **ipload** utility (a Motif GUI)
- The **onpload** utility
- The **onpload** database
- A special IECC utility (a Windows 95 and Windows NT GUI)

The HPL provides the following key features:

- The HPL supports COBOL, ASCII, multibyte, delimited, or binary data. You can add custom drivers to support other data types.
- The HPL can load and unload data that has a different GLS locale than that of the database server.
- The client/server architecture of the HPL lets you use a graphical user interface on any computer on your network.
- Data can be loaded from or unloaded to files, tapes, or application pipes, or to any combination of these three device types.
- The HPL provides context-sensitive on-line help that includes a glossary.
- Any database server on your network can use the database, which allows centralized management of your load and unload controls.



For complete details of how to access and use the HPL, including tutorial examples that take you through the process of loading and unloading data, see the [Guide to the High-Performance Loader](#).

Database Server Utilities

Informix Dynamic Server includes utilities that let you execute administrative tasks directly from the command line or from a windowing environment. The following table lists the supported utilities.

Utility	Use	Where Described
dbexport	Unload a database into text files for importation into another database with dbimport	Informix Migration Guide
dbimport	Create and populate a database from text files created by dbexport	Informix Migration Guide
dbload	Load data into databases or tables that were created with Informix products	Informix Migration Guide
dbschema	Create a file that contains SQL statements needed to replicate a specified table, view, or database	Informix Migration Guide
onaudit	Manage audit masks and auditing configurations	Trusted Facility Manual
oncheck	Check specified disk structures for inconsistencies, repair inconsistent index structures, and display information about disk structures	Administrator's Guide
ondblog	Change logging mode for one or more databases	Administrator's Guide
oninit	Initialize shared memory or shared memory and disk space to bring the database server on-line	Administrator's Guide
onload	Load data that was created with onunload into the database server	Informix Migration Guide
onlog	Display the contents of logical-log files	Administrator's Guide
onmode	Change the database server operating mode and perform various other operations on shared memory, sessions, transactions, parameters, and segments	Administrator's Guide

(1 of 2)

Utility	Use	Where Described
onparams	Modify log-configuration parameters to add or drop a logical-log file or change the size or location of the physical log	Administrator's Guide
onperf	Monitor database server performance through a windowing environment; it has some advantages over onstat	Performance Guide
onshowaudit	Extract information from an audit trail for specified users and database servers	Trusted Facility Manual
onspaces	Modify blobspaces or dbspaces to create or drop a blobspace or dbspace, add or drop a chunk, and start or end mirroring	Administrator's Guide
onstat	Monitor the operation of the database server	Administrator's Guide
ontape	Log, archive, and restore data as outlined in " The ontape Utility " on page 2-12	Administrator's Guide
onunload	Unload data from the database server	Informix Migration Guide

(2 of 2)

W/D

Informix Dynamic Server, Workgroup and Developer Editions, do not support the **oninit** and **onperf** utilities. ♦

Data-Recovery and Data-Restore Features

Informix Dynamic Server includes the following features that help you recover and restore data:

- The **ontape** utility
- ON-Archive ♦
- ON-Bar
- Informix Storage Manager

UNIX

To use these features, you must understand the following terms.

- An *archive* is a copy of one or more database server *dbspaces* (*database spaces*) and any supporting data that you might need to restore them.
- A *logical-log backup* is a copy to tape or disk of logical-log files that have become full and eligible for backup. The logical-log files store a record of database server activity that occurs between archives.
- A *restore* recreates data, particularly *dbspaces*, from an archive and backed-up logical-log files. You must restore the data in two operations:
 - A *physical* restore of *dbspace* or *blob*space data from an archive
 - A *logical* restore that accesses a logical-log backup to re-create in the restored *dbspaces* any transactions that were generated after the archive



For more information about these and other terms, see the [Administrator's Guide](#) and the Glossary in the [Informix Guide to SQL: Reference](#).

The *ontape* Utility

Use the **ontape** utility options of your database server to perform the following tasks:

- Create archives and archive data
- Back up logical-log files
- Perform continuous backups of logical-log files
- Restore data from an archive or logical-log backup
- Change the logging status of a database

The **ontape** utility is interactive and prompts you to mount new tapes when necessary or provide input from the terminal or window.

Important: Do not start **ontape** in background mode (that is, using the UNIX **&** operator on the command line), because you might miss prompts and delay an operation.



For information about the **ontape** utility, see the [Administrator's Guide](#).

UNIX

On-Line Archive Recovery System (ON-Archive)

The ON-Archive recovery system lets you perform the same archive and recovery tasks as the **ontape** utility but provides the following additional features:

- Scheduling and tracking of archives and backups
- Tape reliability and storage options
- Multiple ways of securing data and access to ON-Archive
- Use of multiple tape drives simultaneously for archives and restores
- Unattended operations
- Disaster provision

ON-Archive has the following main components:

- Data to be archived, backed up, or restored
- ON-Archive commands and requests
- ON-Archive programs
- ON-Archive catalog
- Devices to write to and read from
- Media to store the archive or backup data

Important: *The archive tapes produced by **ontape** and ON-Archive are not compatible. You cannot create an archive with **ontape** and restore it with ON-Archive. Also, you must not try to restore some logical logs created by one tool and then restore other logical logs from a tape created by the other tool.*

For complete information about ON-Archive, see the [Archive and Backup Guide](#). For a detailed example of how to use ON-Archive, see the [ON-Archive Quick Start Guide](#).



On-Line Backup and Restore (ON-Bar)

ON-Bar provides a single-system view for backing up and restoring logs and data. Use ON-Bar to make a copy of your data and logical logs as insurance against lost or corrupted data. Data might be lost or corrupted for reasons that range from a program error to a disk crash to a disaster that damages the facility in which your computer resides. To recover data, you first restore the backup copy of the data and then restore the logical logs to bring data as close as possible to the most recent state.

ON-Bar provides backup and restore capability for all dbspaces and logical logs. To prevent performance problems caused by overfull logical-log space, you can have logical logs backed up automatically as soon as they fill. You can schedule backups of dbspaces appropriately as insurance against system failure.

ON-Bar uses the X/Open Backup Services Application Programmer's Interface (XBSA). It connects to either the Informix Storage Manager or another XBSA-compliant storage manager that provides device and storage management functionality.

For complete information about ON-Bar, see the [Backup and Restore Guide](#).



Informix Storage Manager

The Informix Storage Manager (ISM) lets you connect an Informix database server to storage devices for backup and restore operations. ISM also manages backup media. ISM has two main components: the ISM server for data backup and recovery, and the ISM administrator program for management and configuration of the ISM server, storage media, and devices.

ISM works with ON-Bar to perform the following tasks:

- Provide complete media management for all ON-Bar backups and restores
- Track the location of all backup data and move backup data through a managed life cycle
- Provide complete disaster recovery protection for a database server instance



For complete information about ISM, see the [Informix Storage Manager Administrator's Guide](#).

Tools That You Cannot Use for Recovery

You cannot use data-migration utilities such as **onunload** or **dbexport** as a substitute for a database server archive. None of the data-migration utilities are coordinated with the information stored in the logical-log files and, unlike archives, they do not save a copy of system-overhead information important to the database server.



For information about data-migration utilities, see the [Informix Migration Guide](#).

Database Support

Informix Dynamic Server supports the following types of databases:

- ANSI compliant
- Distributed
- Distributed on multiple vendor servers
- Dimensional (data warehouse)

ANSI-Compliant Databases

Informix Dynamic Server supports ANSI-compliant databases. An ANSI-compliant database enforces ANSI requirements, such as implicit transactions and required ownership, that are not enforced in databases that are not ANSI compliant.

You must decide whether you want any of the databases to be ANSI compliant before you connect to a database server. ANSI-compliant databases and databases that are not ANSI-compliant differ in the following areas:

- Transaction processing
- Transaction logging
- Owner naming



- Privileges on and access to objects
- Default isolation level
- Character and decimal data types
- Escape characters
- Cursor behavior
- SQLCODE of the SQL Communications Area (SQLCA)
- Valid SQL statements
- Synonym behavior
- Data recovery
- Portability

For information about ANSI-compliant databases, see the [Informix Guide to Database Design and Implementation](#) and the *Informix Guide to SQL* set of manuals.

Distributed Databases

Informix Dynamic Server provides distributed database processing capabilities that let you query and update more than one database across multiple database servers within a single transaction. A *transaction* is a collection of SQL statements that is treated as a single unit of work.

A *distributed database* has information in several databases. The data can be maintained by a variety of database servers and located on computers that use different operating systems and communication networks. [Figure 2-1 on page 2-17](#) shows the configuration of a distributed database, which includes databases maintained by multiple Informix database servers.

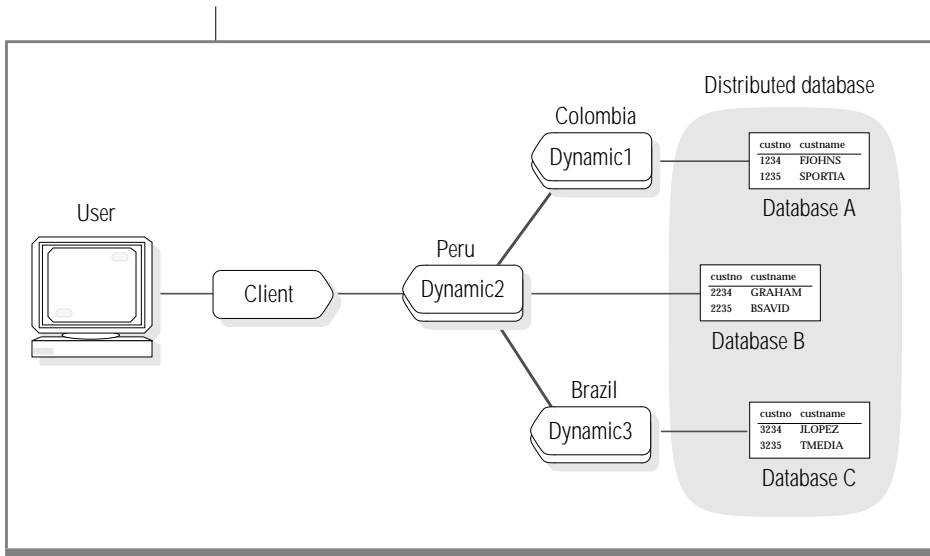


Figure 2-1
Configuration of a
Distributed
Database with
Multiple Instances
of Database Servers

Distributed databases are useful because operations that use databases are often distributed into separate pieces, either organizationally, geographically, or both. A distributed database provides the following advantages:

- Local data can be kept where it is most easily maintained and most frequently used.
- Data from remote sites is available to all users.
- Duplicate copies can be maintained in case one database server crashes.

A distributed database system has the following disadvantages:

- Management of the distributed database server is more involved than management of a single-host system.
- Network access is often slower than local access.

For information on transaction concepts, see the [Informix Guide to SQL: Tutorial](#).



UNIX

Databases Distributed Across Multiple Vendor Servers

TP/XA is a library of functions that lets an Informix database server act as a resource manager in a distributed transaction-processing (DTP) environment. The TP/XA library is installed as part of INFORMIX-ESQL/C to facilitate communication between a third-party transaction manager and Informix Dynamic Server.

You can use Informix database servers with the TP/XA library to create global transactions that span multiple computers and even multiple XA-compliant database servers from different vendors. TP/XA lets you use database-management systems from multiple vendors to store and access your data.

You might want to use the TP/XA feature when your database environment has the following characteristics:

- Data is distributed across multivendor databases.
- Transactions include Informix and non-Informix data.

The TP/XA library and DTP environment adhere to the XA interface specification developed by the X/Open Company to support large-scale, high-performance, on-line transaction-processing applications.

For more information, see the [TP/XA Programmer's Manual](#).



Dimensional Databases

Informix Dynamic Server supports the concept of *data warehousing*. This typically involves a dimensional database that contains large stores of historical data.

W/D

Informix Dynamic Server, Workgroup and Developer Editions, do not support dimensional databases or data warehousing. ♦

A dimensional database is optimized for data retrieval and analysis. The data is stored as a series of snapshots, in which each record represents data at a specific point in time.

A data warehouse integrates and transforms the data that it retrieves before it is loaded into the warehouse. A primary advantage of a data warehouse is that it provides easy access to, and analysis of, vast stores of information.

A data-warehousing environment can store data in one of the following forms:

- Data warehouse

A data warehouse is a database that is optimized for data retrieval. Data is not stored at the transaction level; some level of data is summarized. A data warehouse provides a decision-support environment in which you can evaluate the performance of an entire enterprise over time.

- Data mart

A data mart is a subset of a data warehouse that is stored in a smaller database. It is oriented toward a specific purpose or subject rather than enterprise-wide strategic planning. It can contain operational data, summarized data, spatial data, or metadata.

- Operational data store

An operational data store is a subject-oriented system for decision making that is optimized for looking up one or two records at a time. It is a hybrid form of data warehouse that contains timely, current, integrated information that typically is of a higher level granularity than the transaction. This data can serve as the common source of data for data warehouses.

- Repository

A repository is a system that combines multiple data sources into one normalized database. The records in a repository are updated frequently. Data stored in a repository is operational rather than historical and might be used for specific decision-support queries.



For details of how to plan a database, build and implement a traditional relational database or dimensional database (data warehousing), and work with data types and fragmentation, see the [Informix Guide to Database Design and Implementation](#).

Global Language Support (GLS)

Informix Version 7.3 products include the Global Language Support (GLS) feature. The GLS feature lets the database server handle different languages, cultural conventions, and code sets through the use of different locales. A GLS locale is an environment that has defined conventions for a particular language or culture. See “[Assumptions About Your Locale](#)” on page 5 of the Introduction.

GLS provides support for the following language-related items:

- Collation order of characters
- Definition of uppercase and lowercase conventions
- Non-ASCII characters, including multibyte characters
- Culture-specific formatting for numeric, monetary, date, and time values

With GLS support, the database server does not need to specify how to process culture-specific information directly because this information resides in a GLS locale. When the database server needs culture-specific information, it makes a call to the GLS library. The GLS library, in turn, accesses the GLS locale and returns the information to the Informix product.

For complete information about the GLS feature, see the [Informix Guide to GLS Functionality](#).



Other Database Server Features

This section outlines a number of other features that Informix database servers share. Most of the features discussed involve the use of SQL statements.

Locking

Locking is the process of temporarily limiting access to an object (database, table, page, or row) to prevent conflicting interactions among concurrent processes. When you write applications, be aware of how the database server handles the following locking issues:

- Database locking
- Table locking
- Page and row locking
- Lock mode
- Shared locks
- Waiting for locks



For information on the SQL statements that you use for locking, see the [Informix Guide to SQL: Syntax](#). For general locking information, see the [Informix Guide to SQL: Tutorial](#).

Isolation Level

Isolation level refers to the level of independence among multiple users when they attempt to access common data. The isolation level relates specifically to the locking strategy that you can use for read-only SQL requests. The default isolation level is Read Committed, unless the database is ANSI-compliant, in which case the default is Serializable.

Informix database servers support the SET TRANSACTION statement for databases that use transaction logging. When you use this option, you can set the isolation level to Read Uncommitted, Read Committed, Repeatable Read, and Serializable.



For information about isolation levels, see the description of the SET TRANSACTION statement in the [Informix Guide to SQL: Syntax](#) and the [Informix Guide to SQL: Tutorial](#).

Transaction Logging

The database server supports buffered logging and lets you switch between buffered and unbuffered logging with the SET LOG statement. *Buffered logging* holds transactions in memory until the buffer is full, regardless of when the transaction is committed or rolled back. You can also choose to log or not to log data.



For information on the SET LOG statement, see the [Informix Guide to SQL: Syntax](#). For information about buffered and unbuffered logging, see the [Administrator's Guide](#).

Rolling Back Statements in a Transaction

A *roll back* is a process that reverses an action or series of actions on a database. If you have not yet issued a COMMIT WORK statement for a transaction, the ROLLBACK WORK statement lets you undo any changes that occurred since the beginning of that transaction. You can roll back any statement.



For a list of data definition statements, see the [Informix Guide to SQL: Syntax](#). For a general discussion of transactions, see the [Informix Guide to SQL: Tutorial](#). For a description of the COMMIT WORK and ROLLBACK WORK statements, see the [Informix Guide to SQL: Syntax](#).

Role Separation

Role separation is an option that lets users perform different administrative tasks. Role separation is based on the principle of separation of duties, which can reduce security risks by providing a checks-and-balances mechanism in the system. For example, the person who determines what to audit should be different from the person who monitors the audit trail, and both of these users should be different from the person who is responsible for the operations of the database server.

W/D



Informix Dynamic Server, Workgroup and Developer Editions, do not support role separation. ♦

For information on role separation, see the [Trusted Facility Manual](#) and the description of the CREATE ROLE statement in the [Informix Guide to SQL: Syntax](#).

Getting Up and Running with the Database Server

Basic Tasks for Getting Started	3-3
Migrating to Informix Dynamic Server	3-4
Supported Interfaces and Client Products	3-4
Interfaces	3-5
Informix Enterprise Command Center	3-5
DB-Access	3-5
Optical Subsystem	3-6
Informix SNMP Subagent	3-6
Client SDK Products	3-6
INFORMIX-ESQL/C	3-7
INFORMIX-GLS	3-7
INFORMIX-CLI	3-8
Installing Informix Products.	3-8
Configuring the Database Server Environment	3-9
Required Environment Variables.	3-10
Other Useful Environment Variables	3-11
Setting UNIX Environment Variables	3-11
Setting Windows NT Environment Variables	3-11
Connectivity Files	3-12
Preparing the sqlhosts File	3-12
Network-Configuration Files.	3-13
Network-Security Files.	3-13
Preparing the ONCONFIG Configuration File	3-13
Allocating Disk Space	3-14
Initializing Disk Space	3-14

Configuring the SQL API Environment	3-15
Creating a Database.	3-15
Making the Initial Connection to a Database Server.	3-16
Connecting with IECC	3-16
Connecting with Other SQL APIs	3-16
Creating the Demonstration Database	3-17
Getting Error Message Information	3-18
Performing Standard Administrative Tasks	3-19

This chapter provides information to help you get Informix Dynamic Server and Informix Dynamic Server, Workgroup and Developer Editions, up and running.

Basic Tasks for Getting Started

Perform the following tasks to bring up Informix Dynamic Server successfully:

- Migrate to Informix Dynamic Server, Version 7.3, (if needed) ([page 3-4](#))
- Check client/server compatibility ([page 3-4](#))
- Install Informix products ([page 3-8](#))
- Configure the environment
 - Set required environment variables ([page 3-10](#))
 - Prepare connectivity files ([page 3-12](#))
 - Prepare the configuration file ([page 3-13](#))
 - Allocate and initialize disk space ([page 3-14](#))
- Configure a client (SQL API) environment ([page 3-15](#))
- Choose a database type ([page 3-15](#))
- Connect to the database server ([page 3-16](#))
- Create the demonstration database ([page 3-17](#))
- Get error message information ([page 3-18](#))
- Perform administrative tasks ([page 3-19](#))



Migrating to Informix Dynamic Server

If you migrate to Informix Dynamic Server, Version 7.3, from an earlier version of the database server, start with the information provided in the [Informix Migration Guide](#).

Supported Interfaces and Client Products

This section discusses the following Informix interfaces and client products that Informix Dynamic Server, Version 7.3, supports:

- Interfaces
 - ❑ Informix Enterprise Command Center
 - ❑ DB-Access
 - ❑ Optical Subsystem
 - ❑ Informix SNMP subagent
- SQL API and client products
 - ❑ INFORMIX-ESQL/C
 - ❑ INFORMIX-GLS
 - ❑ INFORMIX-CLI

Other Informix products such as the High-Performance Loader (HPL), Informix Storage Manager (ISM), and database server utilities, are discussed in [Chapter 2](#). Informix Dynamic Server and Informix Dynamic Server, Workgroup and Developer Editions, also might support additional development tools that are not described in this manual.



For information about specific versions of (or platforms for) products that Informix Dynamic Server supports, see the on-line Release Notes for Windows NT environments, or the on-line Release Notes and Machine Notes files for UNIX environments. See also the manual for the client product or SQL API that you want to use, or the INFORMIX-Client Software Developer's Kit or Admin Kit documentation for the specific group of products.

Interfaces

Informix Dynamic Server supports the interfaces that are listed in this section.

Informix Enterprise Command Center

Informix Enterprise Command Center (IECC) runs on Windows 95 or Windows NT. IECC provides a graphical interface that allows the administrator to configure, connect to, control, and monitor the status of the database server. IECC simplifies the process of database server administration and automates common administrative functions. Administrators can perform the following tasks with IECC:

- Configure and bring on-line new database servers
- Create storage areas for new database servers
- Change the mode of database servers
- Back up and restore data
- Monitor query performance
- Monitor database server status



For complete information on how to install and use IECC, see the IECC on-line Help feature. See also the [Informix Enterprise Command Center Installation Guide](#) and the [Informix Enterprise Command Center User Guide](#).

DB-Access

DB-Access is a menu-driven utility that is included with Informix database servers. DB-Access lets you connect to the database server and access, modify, and retrieve information. Use DB-Access menus and screens to access and manipulate the data in a relational database and perform a variety of data-management tasks such as organizing, storing, retrieving, and viewing data.



For information about DB-Access, see the [DB-Access User Manual](#).

W/D



Optical Subsystem

The optical storage subsystem supports the storage of TEXT and BYTE data on optical platters known as WORM optical media. It includes a specific set of SQL statements that support the storage and retrieval of data to and from the optical storage subsystem.

Informix Dynamic Server, Workgroup and Developer Editions, do not support the Optical Subsystem. ♦

For information about the Optical Subsystem, see the [Guide to the Optical Subsystem](#).

Informix SNMP Subagent

Simple Network Management Protocol (SNMP) is a published, open standard for network management. The Informix SNMP subagent lets hardware and software components on networks provide information to network administrators. The administrators use the information to manage and monitor applications, database servers, and systems on networks.

For complete information, see the [Informix SNMP Subagent Guide](#).

Client SDK Products

The Informix Client SDK provides several application-programming interfaces that you can use to develop applications for Informix database servers. These APIs let developers write applications in the language with which they are familiar, such as ESQL, C, C++, and Java. INFORMIX-Connect contains the runtime libraries of the APIs in the Client SDK.

INFORMIX-ESQL/C

INFORMIX-ESQL/C lets programmers embed SQL statements directly into a C program. ESQL/C contains:

- ESQL/C libraries of C functions, which provide access to the database server
- ESQL/C header files, which provide definitions for the data structures, constants, and macros useful to the ESQL/C program
- **esql**, a command that manages the source-code processing to convert a C file that contains SQL statements into an object file



The [INFORMIX-ESQL/C Programmer's Manual](#) is a complete guide to the features that make up the Informix implementation of embedded SQL for C.

INFORMIX-GLS

The INFORMIX-GLS application-programming interface lets ESQL/C programmers develop internationalized applications with a C-language interface. It accesses GLS locales to obtain culture-specific information. Use INFORMIX-GLS to write or change programs to handle different languages, cultural conventions, and code sets. For information on the GLS feature, see [“Global Language Support \(GLS\)” on page 2-20](#).

INFORMIX-GLS provides procedures, macros, and functions to:

- process single-byte, multibyte, and wide characters and strings.
 - String-processing operations include string traversal, concatenation, copying, and character searching.
 - Character-processing operations include character classification, case conversion, code-set conversion, and character comparison.
- convert date, time, monetary, and number values from and to locale-specific data formats.

The INFORMIX-GLS library supports conversion of a locale-specific string to its internal database representation and formatting of an internal database representation to a locale-specific string.



The [INFORMIX-GLS Programmer's Manual](#) is a complete guide to the features that make up the INFORMIX-GLS API.

INFORMIX-CLI

INFORMIX-CLI is the Informix implementation of the Microsoft Open Database Connectivity (ODBC) standard. INFORMIX-CLI is a Call Level Interface that supports SQL statements with a library of C functions. An application calls these functions to implement ODBC functionality. Use the INFORMIX-CLI application programming interface (API) to access an Informix database and interact with an Informix database server.

INFORMIX-CLI consists of the following parts:

- The INFORMIX-CLI libraries
These libraries provide the functions and values for the INFORMIX-CLI API.
- A driver manager
A driver manager provides an interface between an INFORMIX-CLI application and the INFORMIX-CLI driver. It also checks parameters and transitions.
- The INFORMIX-CLI driver
The driver provides an interface between a data source and either a driver manager or an application, depending on whether the system architecture includes a driver manager. If a driver manager is not included, the INFORMIX-CLI driver performs the driver manager functions.



The [INFORMIX-CLI Programmer's Manual](#) is a user guide and reference manual for the features that make up the API.

Installing Informix Products

You can install Informix database servers and client products on a hardware configuration that runs on a UNIX operating system or in a Windows environment.



For information about platform-specific requirements, refer to the on-line Machine Notes for your UNIX platform. For information about installing a database server or other Informix product in a UNIX or Windows environment, see the [Installation Guide](#) for the product or environment.

To install a database server or other Informix product, you must prepare the UNIX or Windows environment, load the product files supplied by Informix onto your computer, and run an installation script (UNIX) or follow instructions in the windowing environment (such as Windows NT) to correctly set up the product files.

If you install more than one Informix product, complete all installation procedures for one product before you start to install the next product. Do not load the files from another Informix product onto your computer until you complete the current installation.

If you want to preserve product files of earlier versions, you must create separate directories for each version of your Informix products. You must set the **INFORMIXDIR** environment variable to the appropriate directory name for the version that you want to access.



For information about the **INFORMIXDIR** environment variable, see the [Administrator's Guide](#) and the [Informix Guide to SQL: Reference](#). For information on memory and disk space requirements for installation as well as the product installation order, see the [Installation Guide](#).

Configuring the Database Server Environment

After you install the database server, you can configure the database server environment. *Configuration* refers to setting specific parameters that affect data processing in the client/server environment.

You must configure your database server environment before you can connect a client application with a database server. The following list identifies the basic configuration requirements:

- Set environment variables
- Prepare the **sqlhosts** information in the file or registry
- Prepare the ONCONFIG file
- Allocate disk space
- Initialize disk space



For a full discussion of requirements, see the [Administrator's Guide](#).

Required Environment Variables

The following table shows the minimum environment variables that Informix database servers use. The environment variables described in this section must be set before an Informix client application can connect with the database server.

Environment Variable	Description
INFORMIXDIR	The INFORMIXDIR environment variable must be set to the full pathname of the directory where the Informix product has been installed.
INFORMIXSERVER	The INFORMIXSERVER environment variable specifies the name of the default database server. The sqlhosts file must contain a database server name entry that matches the value for INFORMIXSERVER . INFORMIXSERVER must be set to the same value that you assign to the DBSERVERNAME parameter in the ONCONFIG configuration file.
PATH	The PATH environment variable must include the full pathname of the directory where the executable files for the database server reside.
ONCONFIG	The ONCONFIG environment variable specifies the name of the active ONCONFIG configuration file that the database server uses. If the ONCONFIG environment variable is not set, the database server uses values from the file \$INFORMIXDIR/etc/onconfig .
TERM	The TERM environment variable specifies the type of terminal interface. If the TERM environment variable has not been set on your computer system, you might need the assistance of your UNIX system administrator because this variable is system and terminal dependent.

UNIX



The **PATH** and **TERM** environment variables are standard UNIX environment variables that must be set to run Informix products. ♦

For detailed information about how these and other environment variables affect the database server, see the [Informix Guide to SQL: Reference](#) and the [Administrator's Guide](#).

UNIX



UNIX



WIN NT

Other Useful Environment Variables

Additional environment variables let you work with the Global Language Support (GLS) feature. These include:

- ★ **CLIENT_LOCALE**
- ★ **DB_LOCALE**
- ★ **SERVER_LOCALE**

Other environment variables let you work with the client:

- ★ **INFORMIXTERM**
- ★ **TERMINFO** ♦

For detailed information about GLS environment variables, see the [Informix Guide to GLS Functionality](#). For information on how setting certain environment variables can affect performance, see the [Performance Guide](#).

Setting UNIX Environment Variables

Set UNIX environment variables in one of the following ways:

- At the system prompt
- In an environment-configuration file
- In your **.profile** or **.login** file

You can check the validity of environment variable settings with the **chkenv** utility.

***Tip:** Informix recommends that you set the environment variables in the appropriate start-up file for your shell.*

Setting Windows NT Environment Variables

Depending on the application, you can set Windows NT environment variables in one of the following ways:

- In the Windows NT registry
- In an **autoexec.bat** file
- With a system applet



Modify entries in the registry with the **regedt32.exe** Registry Editor.

The installation procedure prepares a file, **setenv.cmd**, that sets the environment variables to their correct values. The **setenv.cmd** file is stored in the %INFORMIXDIR% directory. You must execute **setenv.cmd** before you can use any of the command-line utilities. If you choose **Command Line** utilities from the Informix Enterprise Command Center, the file executes for you automatically. You can also execute **setenv.cmd** from the command line.

***Tip:** Informix recommends that you set the environment variables in the appropriate start-up file for your windowing environment.*

Connectivity Files

The connectivity files contain information that enables client/server communication. The following connectivity configuration files must be present for client/server connections to succeed:

- **\$INFORMIXDIR/etc/sqlhosts** file on UNIX systems (an ASCII file), or **%INFORMIXDIR%\etc\sqlhosts** in Windows NT environments (in the registry)
- Network-configuration files
- Network-security files

Of these files, the database server administrator manages only the Informix **sqlhosts** file. The other files are UNIX operating-system files that the UNIX system (or network) administrator manages.

Preparing the sqlhosts File

An **sqlhosts** file must exist on each computer that has either an Informix client application or database server. If no database server resides on the computer where the client program runs, an **sqlhosts** file is required on the host computers of both the client application and the database server.

The **sqlhosts** file specifies the location of the database server and the type of connection (interface/protocol combinations) for a connection to that database server. A client application uses information in the **sqlhosts** file to establish a connection to a database server. The **sqlhosts** file must contain an entry for each type of connection to each database server on the network.



UNIX



UNIX



You can use the **INFORMIXSQLHOSTS** environment variable to change the location or the name of the **sqlhosts** file from the default: **\$INFORMIXDIR/etc** on UNIX and **%INFORMIXDIR%\etc** in Windows NT environments.

For information about the **INFORMIXSQLHOSTS** environment variable, see the [Informix Guide to SQL: Reference](#). For detailed information about how to prepare the **sqlhosts** file, refer to the [Administrator's Guide](#).

Network-Configuration Files

In addition to the **sqlhosts** files, TCP/IP connections require entries in the **/etc/hosts** and **/etc/services** UNIX systems files. IPX/SPX connections also require auxiliary files. However, unlike TCP/IP, the names of the auxiliary files depend on the hardware vendor.

The network-configuration files are described in the [Administrator's Guide](#) and in operating-system manuals.

Network-Security Files

Informix database servers follow UNIX security requirements for making connections. Thus, the UNIX system administrator might need to make modifications to the **/etc/passwd**, **etc/hosts**, **~/.rhosts**, and other related files.

The network-security files are described in the [Administrator's Guide](#) and in operating-system manuals.

Preparing the ONCONFIG Configuration File

The ONCONFIG configuration file contains values for global and database server-specific configuration parameters that describe the database server environment. The location of the ONCONFIG file is **\$INFORMIXDIR/etc/\$ONCONFIG** on UNIX systems or **%INFORMIXDIR%\etc\%ONCONFIG%** in Windows NT environments.

An **onconfig.std** file contains initial settings for the configuration parameters and serves as a template for customized configuration files. During installation, the file is loaded into **\$INFORMIXDIR/etc/onconfig.std** on UNIX systems or **%INFORMIXDIR%\etc\onconfig.std** in Windows NT environments.



You customize an ONCONFIG file to describe a specific environment. You can have several ONCONFIG files to describe different environments, such as learning, production, and development.

For detailed information about all configuration parameters and how to prepare the ONCONFIG file, see the [Administrator's Guide](#). For information on how configuration parameter settings can affect performance, see the [Performance Guide](#).

Allocating Disk Space

Before you can initialize the database server, you must decide which type of disk space to allocate for the database server. There are two distinct types of disk space:

- Cooked file space, in which the operating system manages physical disk I/O
- Raw disk space, in which the database server manages physical disk I/O

For more information, see “[The Disk Component](#)” on page 1-6 and “[Raw \(Unbuffered\) Disk Management](#)” on page 1-12.



To learn more about cooked and raw space and for information about how to allocate disk space, see the [Administrator's Guide](#).

Initializing Disk Space

In most cases, you initialize disk space just once for a database server. You initialize disk space when you bring the database server on-line for the first time. When the database server is in on-line mode, you can connect with it and perform all database activities.



For information about how to initialize disk space, see the [Administrator's Guide](#).

GLS



Configuring the SQL API Environment

After you install an SQL API or client product, you can configure the product environment. You must set the following environment variables on the computer on which you have installed the SQL API or client product before you can compile, link, and run the product:

- ★ **INFORMIXDIR**
- ★ **INFORMIXSERVER**
- ★ **PATH**

If your client application uses a nondefault client locale or accesses a database that has a nondefault database locale, you must set the **CLIENT_LOCALE** and **DB_LOCALE** environment variables. ♦

For information about environment variables that affect SQL API programs, see the [Informix Guide to SQL: Reference](#). For information about GLS environment variables and locales, see the [Informix Guide to GLS Functionality](#).

Creating a Database

When the database server is on-line, you can connect client applications and SQL APIs to it and begin to create databases. However, before any users can connect to the database server, you must take the following actions:

- Set environment variables that specify the location of the software and the name of the database server, as outlined in “[Required Environment Variables](#)” on page 3-10.
- Decide whether you want any of the databases to be ANSI compliant. See “[ANSI-Compliant Databases](#)” on page 2-15.



For information about ANSI-compliant databases, see the [Informix Guide to Database Design and Implementation](#). For information about ANSI syntax in SQL statements, see the [Informix Guide to SQL: Syntax](#).

Making the Initial Connection to a Database Server

Before you can access information in a database, the client application must connect to the database server environment. You can connect a client application to Informix Dynamic Server in two main ways:

- Specify a connection option on the Informix Enterprise Command Center graphical interface.
- Issue an SQL statement in your client application.

Connecting with IECC

The Informix Enterprise Command Center interface displays Informix database servers and administration tools. Select the **Informix Neighborhood** icon and then display a selection of database server icons in the window.

Before you can connect to one or more database servers, you must select the database server in the IECC window, have Informix administrative privileges on the database server, and configure the database server definitions.

Use IECC windows to specify connection options (such as how long or how often to try to connect), locate new database servers in the network, and disconnect from a database server. In addition, the IECC SQL Editor accepts the CONNECT and DISCONNECT statements.



For information on how to use IECC to connect to, disconnect from, and configure definitions for database servers, see the [Informix Enterprise Command Center User Guide](#).

Connecting with Other SQL APIs

To connect to and disconnect from a database server, issue SQL statements from DB-Access or another SQL API such as INFORMIX-ESQL/C or INFORMIX-CLI. You also can use DB-Access menu and screen options to execute an existing command file or connect to a listed database server.

The client application executes a `CONNECT` statement to connect to a database server or a `DATABASE` statement to open the database. (The `CONNECT` statement is preferred because it conforms with ANSI and X/Open standards, which attempt to specify uniform syntax.)

The `CONNECT` statement connects the application to a *database environment*, which is a database server or a database server and a database. For example, the following command connects a client application to a database server named **alaska** and opens the database named **rivers**:

```
CONNECT TO rivers@alaska
```

Use the `DISCONNECT` statement to terminate the connection between your client application and the database server. With `ESQL/C` applications, you can use the `SET CONNECTION` statement to reestablish an active connection and make it current. You can also use the `SET CONNECTION` statement to make the current connection dormant. (DB-Access does not support the `SET CONNECTION` statement.)



The complete syntax for the `CONNECT`, `DATABASE`, `DISCONNECT`, and `SET CONNECTION` statements is covered in the [Informix Guide to SQL: Syntax](#). For information about how to use DB-Access, `INFORMIX-ESQL/C`, or `INFORMIX-CLI`, see the [DB-Access User Manual](#), [INFORMIX-ESQL/C Programmer's Manual](#), or [INFORMIX-CLI Programmer's Manual](#), respectively.

Creating the Demonstration Database

The DB-Access utility, which is provided with your Informix database server products, includes a script that creates a demonstration database called **stores7**. Many examples that are included in your Informix documentation are based on the **stores7** demonstration database. Install the demonstration database with the **dbaccessdemo7** script that is located in the `$INFORMIXDIR/bin` directory on UNIX systems or the `%INFORMIXDIR%\bin` directory in Windows NT environments.

The `INFORMIX-ESQL/C` product also includes a script that creates the **stores7** demonstration database. This script, called **esqlcdemo7**, is also located in one of the preceding directories, depending on your platform.

A second demonstration database, named **sales_demo**, provides a way to work with data-warehousing concepts.



For an explanation of how to create and populate the **stores7** demonstration database, refer to the [DB-Access User Manual](#). For a description of the structure and contents of **stores7**, see the [Informix Guide to SQL: Reference](#). For information on the permissions required to use the command files, see the [Administrator's Guide](#). For information on how to work with the **sales_demo** database, see the [Informix Guide to Database Design and Implementation](#).

UNIX

WIN NT



Getting Error Message Information

Informix software products provide text files that contain all the Informix error messages and their corrective actions. To read the error messages in the text file, Informix provides scripts that let you display error messages on the screen (**finderr**) or print formatted error messages (**rofferr**). ♦

To read error messages and corrective actions under Windows NT, use the **Informix Find Error** utility. To display this utility, choose **Start→Programs→Informix** from the Task Bar. ♦

For information on the error message files and a detailed description of the **finderr** and **rofferr** scripts, see the Introduction to *Informix Error Messages* in Answers Online.

Performing Standard Administrative Tasks

The database server administrator should routinely perform the following tasks after the database server is initialized:

- Check that users have set the correct environment variables.
Every user of an Informix product must have the correct environment variables set to work with the database server and client applications and, if needed, to use GLS locales.
- Review the database server configuration parameters.
You can examine the configuration of a database server in various ways.
- Transfer data that was created on other Informix database servers.
Use a supported unload utility and import the data to the database server. Use the High-Performance Loader if your database server supports it.
- Warn the UNIX system administrator about **cron** jobs.
Some UNIX systems run **cron** jobs that routinely delete all files from the **/tmp** directory. ♦

UNIX



For complete information about these and other administrative tasks, see the [Administrator's Guide](#). For information about the High-Performance Loader, see the [Guide to the High-Performance Loader](#).

Using the Documentation

Who Uses Informix Dynamic Server?	4-3
Database Users	4-3
Programmers and Application Developers	4-4
Database Administrators	4-4
Informix Dynamic Server Administrators and Operators	4-4
Task-Documentation Matrix.	4-5
The Informix Dynamic Server Documentation Set	4-11
Manuals for Additional Supported Products	4-15



This chapter provides an overview of Informix Dynamic Server users and the tasks that each type of user is likely to perform. It includes a matrix of those tasks and where to find information throughout the Informix Dynamic Server, Version 7.3, documentation set.

This chapter also contains an alphabetical list of the manuals that are provided with the database server software and a list of the documentation for some related Informix client products.

***Tip:** See the Release Notes for current information about what products Informix Dynamic Server, Version 7.3, supports. See the Documentation Notes for additional information that postdates the manuals.*

Who Uses Informix Dynamic Server?

The following major groups use Informix Dynamic Server:

- Database users
- Programmers and application developers
- Database administrators
- Informix Dynamic Server administrators and operators

Database Users

If you are a database user, you access, insert, update, and manage information in databases with SQL, which is often embedded in a client application.

In [Figure 4-1 on page 4-6](#), look for the term **USER** in the Role column to identify database user tasks and the books that will help you complete your tasks.

Programmers and Application Developers

If you develop client applications, Informix Dynamic Server offers a number of possibilities for data management, multimedia, isolation levels, and so on. The database server integrates information objects such as scanned and digitized images, voice, graphs, facsimiles, and word-processing documents into an SQL-based relational database.

In [Figure 4-1 on page 4-6](#), look for the term **DEV** in the Role column to identify application developer tasks and the manuals that will help you complete your tasks.

Database Administrators

If you are a database administrator (DBA), you are primarily responsible for creating and managing access control for databases. You use SQL statements to grant and revoke privileges to ensure that the correct individuals are able to perform the actions they need to and that untrained or unscrupulous users are kept from performing potentially damaging or inappropriate resource-intensive activities.

In [Figure 4-1 on page 4-6](#), look for the term **DBA** in the Role column to identify database administrator tasks and the manuals that will help you complete your tasks.

Informix Dynamic Server Administrators and Operators

If you are an Informix Dynamic Server administrator, you are responsible for the installation, maintenance, administration, and operation of the entire database server that might manage many individual databases.

If you are an Informix Dynamic Server operator, you are responsible for backing up and restoring databases and for carrying out similar routine tasks associated with database server administration.

In [Figure 4-1 on page 4-6](#), look for the term **ADMIN** in the Role column to identify Informix Dynamic Server administrator tasks and the books that will help you complete your tasks. Operators will perform a subset of these tasks.

Task-Documentation Matrix

The Informix Dynamic Server task-documentation matrix in [Figure 4-1 on page 4-6](#) provides a quick reference to all the documentation that is provided with Version 7.3 Informix Dynamic Server and Informix Dynamic Server, Workgroup and Developer Editions. The matrix includes the following three columns:

- **If You Want To.** A task you might want to perform.
- **Manual.** The primary book that contains information to help you perform the task.
- **Role.** The person most likely to perform the task.

Use the matrix to associate specific tasks with the manuals that will help you perform the tasks. For a list of all the Informix Dynamic Server documentation, see [“The Informix Dynamic Server Documentation Set” on page 4-11](#).

Figure 4-1
Informix Dynamic Server Task-Documentation Matrix

If You Want To:	Manual	Role
Install the database server in different environments	Installation Guide	ADMIN
Configure the database server	Administrator's Guide	ADMIN
Identify sources of information for monitoring the database server (such as the message log and the sysmaster database)		
Initialize the database server		
Manage database-logging status		
Manage the logical and physical log		
Manage modes		
Manage virtual processors, shared memory, and disk space		
Perform consistency checking		
Perform distributed transactions		
Perform mirroring operations		
Recover from a failed two-phase commit		
Use utilities such as oncheck , onlog , and onstat to perform administrative tasks		
Work with these utilities: oninit , onmode , onparams , onspaces		
Manage fragmentation (not in Editions)		
Perform a parallel database query (PDQ) (not in Editions)		
Back up the logical log	Archive and Backup Guide	ADMIN
Configure, use, and administer ON-Archive		
Configure and use the ontape utility		
Create an archive		
Restore data		
Refer to examples to set up, back up, and restore data for ON-Archive	ON-Archive Quick Start Guide	ADMIN

(1 of 5)

If You Want To:	Manual	Role
Configure and use the ON-Bar backup and restore system Copy your data and logical logs as insurance against lost or corrupted data Restore data	Backup and Restore Guide	ADMIN
Connect your Informix database server to storage devices for backup and restore operations Manage backup media for all ON-Bar backups and restores Track the location of all backup data Move backup data through a managed life cycle Provide disaster recovery for a database server instance	Informix Storage Manager Administrator's Guide	ADMIN
Migrate to Informix Dynamic Server from an earlier version of an Informix database server Move data between different physical equipment (computer and storage devices) and different operating systems Move data between database servers with different language support Work with these utilities: dbexport , dbimport , dbload , dbschema , onmode -b Revert from Informix Dynamic Server to an earlier version of an Informix database server	Informix Migration Guide	ADMIN
Eliminate performance bottlenecks; for example, by balancing load on system resources or adjusting configuration parameters Identify the database activities that affect critical resources Improve the performance of a query Manage indexes Monitor system resources (CPU, memory, disk) that are critical to performance Use utilities such as onperf for performance monitoring and tuning Use secondary access methods such as B-trees Control placement and size of tables and table extents	Performance Guide	ADMIN

(2 of 5)

If You Want To:	Manual	Role
Control placement and size of fragments (not in Editions) Manage resources for parallel database queries (PDQ) (not in Editions)	Performance Guide	ADMIN
Use the secure-auditing utilities (onaudit , onshowaudit) to set up, administer, and interpret audit trails	Trusted Facility Manual	ADMIN
Use the Optical Subsystem interface for an optical storage subsystem that supports the storage of TEXT and BYTE data on optical platters (WORM optical media) Use the exclusive set of SQL statements that support the storage and retrieval of data to and from the optical storage subsystem	Guide to the Optical Subsystem	ADMIN
Use the ipload and onpload utilities to load or unload large quantities of data to or from an Informix database Use the graphical user interface that prepares the onpload database Move data to a different computer or configuration Alter the schema of a table	Guide to the High-Performance Loader	ADMIN
Create databases and manage access	Informix Guide to SQL: Syntax	DBA
Understand the impact of database design and use on performance	Performance Guide	DBA
Design and implement database for Informix database servers Define data for Informix databases Define a fragmentation strategy Design different data models Use supported data types and SQL to implement a database Work with the sales_demo demonstration database	Informix Guide to Database Design and Implementation	DBA

(3 of 5)

If You Want To:	Manual	Role
<p>Use the Informix SNMP subagent to extract information from an Informix database server and pass that information to a network manager</p> <p>Perform low-impact sampling of database server performance</p> <p>Interpret the contents of application, Informix, and RDBMS Management Information Bases (MIBs)</p> <p>Use SNMP applications to remotely monitor the status of all Informix database servers, operating systems, routers, printers, and other devices on a network</p>	Informix SNMP Subagent Guide	ADMIN
<p>Understand the concepts of data replication</p> <p>Design, define, monitor, and control your own Enterprise Replication system</p> <p>Install the Enterprise Replication client/server application</p> <p>Administer and manage data replication throughout your open-system enterprise</p>	Guide to Informix Enterprise Replication	ADMIN
Use a primary access method	Informix Guide to SQL: Syntax	DEV
Assign data types to columns	Informix Guide to SQL: Tutorial Informix Guide to Database Design and Implementation Informix Guide to SQL: Syntax	DEV
Embed SQL statements directly into C programs	INFORMIX-ESQL/C Programmer's Manual	DEV
Install client products that the database server supports	INFORMIX-Client Products Installation Guide for UNIX Informix Client Products Installation and Configuration Guide for Microsoft Windows Environments	DEV
<p>Use the GLS features that let Informix SQL APIs and database servers handle different languages, cultural conventions, and code sets</p> <p>Specify a nondefault locale</p>	Informix Guide to GLS Functionality INFORMIX-GLS Programmer's Manual	DEV

(4 of 5)

If You Want To:	Manual	Role
Create custom application with the INFORMIX-CLI API for database access	INFORMIX-CLI Programmer's Manual	DEV
Work with the TP/XA library in a distributed transaction-processing (DTP) environment	TP/XA Programmer's Manual	DEV
Learn how GLS affects database server migration	Informix Migration Guide	DEV
Find a list of all the environment variables that Informix database servers support	Informix Guide to SQL: Reference Informix Migration Guide	DEV
Acquaint yourself with terms in an Informix Glossary	Informix Guide to SQL: Reference	DEV
Start and stop Enterprise Replication and perform other basic replication tasks at remote sites	Guide to Informix Enterprise Replication	USER
Invoke the DB-Access utility Access, retrieve, and modify data from a database Work with a demonstration database	DB-Access User Manual	USER
Compose correct SQL statements	Informix Guide to SQL: Syntax	USER
Find corrective actions to error messages	on-line Informix Error Messages	USER
Use the Informix Enterprise Command Center graphical utility to work with the database server	Informix Enterprise Command Center User Guide or IECC on-line Help	ALL
		(5 of 5)

The Informix Dynamic Server Documentation Set

This section summarizes the documentation that is distributed with Informix Dynamic Server, Version 7.3. Figure 4-2 lists manuals alphabetically.

Figure 4-2
Manuals in the Documentation Set

Book Title	Description
<i>Administrator's Guide for Informix Dynamic Server</i>	This combination user guide and reference manual describes the features of Informix Dynamic Server and Informix Dynamic Server, Workgroup and Developer Editions, in detail. It is intended to help you understand, configure, and use the database server.
<i>Archive and Backup Guide for Informix Dynamic Server</i>	This guide describes the components, features, and tasks that make up the recovery system provided with the database server. It provides detailed information on the ON-Archive and ontape facilities.
<i>Backup and Restore Guide for Informix Dynamic Server</i>	This user guide and reference manual explains the concepts and methods required to back up and restore data with the ON-Bar backup and restore system.
<i>DB-Access User Manual</i>	This user guide and reference manual describes how to use the DB-Access utility to access, modify, and retrieve information from Informix database server products. On-line Help is available within DB-Access as well.
<i>Getting Started with Informix Dynamic Server</i>	This guide provides an overview of the relational database management system, summarizes important features of Informix products, and provides information to help you use the documentation that is included with Informix Dynamic Server products.
<i>Guide to Informix Enterprise Replication</i>	This guide contains information to help you understand the concepts of data replication, design your own Enterprise Replication client/server application system, install Enterprise Replication, and administer and manage data replication throughout your enterprise.

(1 of 4)

Book Title	Description
Guide to the High-Performance Loader	This guide describes how to use the High-Performance Loader (HPL) to efficiently load an unload large quantities of data to or from an Informix database.
Guide to the Optical Subsystem	This guide provides details of how to use the Optical Subsystem feature.
Informix Enterprise Command Center Installation Guide	This guide describes how to install the Informix Enterprise Command Center (IECC), which provides a graphical interface for database servers in Windows NT environments.
Informix Enterprise Command Center User Guide	This guide describes the Informix Enterprise Command Center, which runs in a Windows NT environment. It tells the database server administrator how to use IECC to configure, control, and monitor the status of the database server.
Informix Error Messages	This Answers OnLine product provides a complete list of Informix-specific error messages and describes their corrective actions for current Informix products and earlier Informix products that are still supported. It also describes how to use the finderr and rofferr scripts to read error messages on-line.
Informix Guide to Database Design and Implementation	This guide documents how to design and implement databases for Informix database servers. It describes tasks usually performed by the DBA and addresses data definition with Informix databases. The guide describes how to design different data models and use the supported data types and SQL to implement a database.
Informix Guide to GLS Functionality	This manual describes the Global Language Support (GLS) feature that is available in Informix products. The GLS feature allows Informix client products and Informix database servers to handle different languages, cultural conventions, and code sets. This manual describes only the language-related topics that are unique to GLS.
Informix Guide to SQL: Reference	This manual describes the Informix system catalog tables, Informix and other environment variables that you might need to set, and the data types supported by Informix database servers. It also contains a complete description of the demonstration database and a glossary of terms used in Informix manuals.

Book Title	Description
<i>Informix Guide to SQL: Syntax</i>	This manual contains the complete syntax descriptions for SQL and Stored Procedure Language (SPL) statements.
<i>Informix Guide to SQL: Tutorial</i>	This tutorial includes instructions for using SQL. It also describes the fundamental ideas and terminology for planning, implementing, and using a relational database management system.
<i>Informix Migration Guide</i>	This manual describes the tasks that you perform when you move data from one location to another and when you migrate existing Informix databases to various Informix database servers. It discusses such database server utilities as dbexport , dbimport , dbload , dbschema , and onmode .
<i>Informix SNMP Subagent Guide</i>	This manual introduces Simple Network Management Protocol (SNMP) and describes the subagent that provides information about Informix database servers to network-management tools. The guide also documents the Management Information Bases (MIBs) that specify the information that the onSNMP program provides to the network-management tools.
<i>Informix Storage Manager Administrator's Guide</i>	This guide describes how to connect your Informix database server to storage devices for backup and restore operations, and how to manage backup media with the Informix Storage Manager.
<i>Installation Guide for Informix Dynamic Server on UNIX</i>	This guide contains instructions for installing Informix Version 7.3 database server products on computers that run the UNIX operating system. It also describes common installation problems and indicates how to solve them.
<i>Installation Guide for Informix Dynamic Server on Windows NT</i>	This guide contains instructions for installing Informix Version 7.3 database server products on computers that run in the Windows NT environment. It also describes common installation problems and indicates how to solve them.

(3 of 4)

Book Title	Description
<i>ON-Archive Quick Start Guide</i>	This guide presents a complete example of setting up, backing up, and restoring data for a system. It introduces you to the concepts, terminology, and basic tasks involved in using ON-Archive as your recovery system. It includes a section on troubleshooting common problems and a scripted sample session.
<i>Performance Guide for Informix Dynamic Server</i>	This manual explains how to configure and operate Informix Dynamic Server and Informix Dynamic Server, Workgroup and Developer Editions, to improve overall system throughput and how to improve the performance of SQL queries.
<i>Trusted Facility Manual for Informix Dynamic Server</i>	This guide describes the secure auditing facility and includes information on how to set up and administer audit trails, extract and interpret audit records, and use utilities and SQL statements for audit analysis.

(4 of 4)

Manuals for Additional Supported Products

This section summarizes additional Informix manuals that you can use when you work with the Informix Dynamic Server, Version 7.3, software. Figure 4-3 lists manuals for Informix client products alphabetically.

Figure 4-3
Supplementary Manuals for Informix Dynamic Server

Book Title	Description
<i>INFORMIX-CLI Programmer's Manual</i>	This user guide and reference manual discusses the features that make up the Informix implementation of the Microsoft Open Database Connectivity (ODBC) standard. It explains how to use the INFORMIX-CLI application-programming interface (API) to access Informix databases, manipulate the data in your program, interact with the database server, and check for errors.
<i>INFORMIX-ESQL/C Programmer's Manual</i>	This manual explains how to use INFORMIX-ESQL/C, the Informix implementation of embedded Structured Query Language for C, to create client applications with database-management capabilities. It is a complete guide to the features of ESQL/C that enable programmers to interact with the database server, access databases, manipulate the data in programs, and check for errors.
<i>INFORMIX-GLS Programmer's Manual</i>	This manual describes the INFORMIX-GLS application-programming interface in Informix products. INFORMIX-GLS allows ESQL/C programmers to write or change programs to work with different locales.
<i>INFORMIX-Client Products Installation Guide for UNIX</i>	This guide describes how to install Informix client products on computers that run the UNIX operating system. It also lists GLS support files and describes how to solve installation problems.
<i>Informix Client Products Installation and Configuration Guide for Microsoft Windows Environments</i>	This guide describes how to install Informix client products on computers that run in Windows environments (Windows NT and Windows 95) and discusses how to solve common installation problems.
<i>TP/XA Programmer's Manual</i>	This guide describes how to use the TP/XA library to program in an X/Open DTP environment when data is distributed across multivendor or non-Informix databases.

Index

A

Administrative tasks
 of database administrator 4-4
 role separation for 2-22
 standard, listed 3-19
 Allocating disk space 3-14
 ANSI compliance Intro-14, 2-15
 Architecture
 client/server 1-9
 distributed databases 1-19
 dynamic thread allocation 1-13
 fault tolerance and high
 availability 1-15
 high performance 1-11
 memory management 1-12
 parallelization 1-14
 raw disk management 1-12
 scalability 1-11
 Archive, and backups 1-15
 Asynchronous data replication 2-6
 Auditing 1-21

B

Backup
 Informix Storage Manager 2-14
 ON-Bar 2-14
 transaction records 1-15
 Buffered logging, description
 of 2-22

C

chkenv utility 3-11
 Client
 configuring environment for 3-15
 description of 1-8
 improving performance and
 availability 2-6
 supported products 3-4
 Client application
 description of 2-3
 running on a network 1-9
 Client SDK products 3-6
 Client/server
 architecture, description of 1-8
 configurations 1-10
 definition of 1-8
 Code set, ISO 8859-1 Intro-5
 Comment icons Intro-8
 Commit
 heterogeneous 1-20
 two-phase 1-20
 COMMIT WORK statement 2-22
 Communication
 client to database server. *See*
 Connectivity.
 network. *See* Network
 communication.
 Compliance with industry
 standards Intro-14
 Configuration
 files for network 3-13
 ONCONFIG file 3-10
 requirements 3-9
 Configuring the database server
 environment 3-9

- Configuring the SQL API environment 3-15
- CONNECT statement 3-16, 3-17
- Connection
 - to database server with IECC 3-16
 - to database server with SQL statement 3-16
 - to database server, initial 3-16
- Connectivity
 - files, overview of 3-12
 - sqlhosts file 3-12
- Cooked disk space, description of 1-7
- Cost-based optimizer 2-8
- CPU, relationship to virtual processor 1-7
- CREATE INDEX statement 1-14
- CREATE ROLE statement 2-23
- CREATE TABLE statement 1-14
- Creating a database 3-15
- Creating a fragmentation strategy 1-14
- Creating a fragmented table 1-14
- Creating the demonstration database 3-17
- Cross-reference icons Intro-8

D

- Data
 - manipulating with DB-Access 3-5
 - storage in data warehouse environment 2-19
- Data mart, described 2-19
- Data migration tools, and database server recovery 2-15
- Data recovery mechanisms
 - data replication 1-17
 - mirroring 1-16
- Data replication
 - advantages of 2-6
 - defined 2-5
 - description of 1-17
 - enterprise 2-7
 - high-availability 2-6
 - two types 2-6
- Data storage, for database servers 1-7

- Data warehouse, described 2-19
- Database
 - ANSI compliance 2-15
 - creating 3-15
 - demonstration Intro-5, 3-17
 - dimensional 2-18
 - distributed, description of 2-16
 - onload for high-performance loader 2-9
 - relational 3-5
 - relational, description of 2-3
 - support 2-15
- Database environment, definition of 3-17
- Database management systems, overview of relational 2-4
- Database server
 - administrator, description of 4-4
 - available data types 2-5
 - client/server architecture 1-8
 - configuring 3-9
 - data storage 1-7
 - description of 2-3
 - distributed database processing 2-16
 - fault-tolerant features 1-15
 - high performance of 1-11
 - initial connection to 3-16
 - ON-Bar backup and restore 2-14
 - parallel database query feature 1-14
 - raw-disk management 1-12
 - scalability of 1-11
 - security 1-21
 - system catalog for 2-4
 - utilities, listed 2-10
- Database server feature
 - isolation level 2-21
 - locking 2-21
 - role separation 2-22
 - rolling back transactions 2-22
 - transaction logging 2-22
- DATABASE statement 3-17
- DB-Access utility
 - connecting with 3-16
 - creating the demonstration database 3-17
 - described 3-5
 - documentation 4-10, 4-11

- dbexport utility 2-10, 2-15, 4-7
- dbimport utility 2-10, 4-7
- dbload utility 2-10, 4-7
- DBMS. *See* Database management system.
- dbschema utility 2-10, 4-7
- Dbspaces
 - archiving 2-12
 - restoring 2-12
- Default locale Intro-5
- Definitions
 - data mart 2-19
 - data warehouse 2-19
- Demonstration database Intro-5
 - creating 3-17
 - overview of 3-17
- DISCONNECT
 - statement 3-16, 3-17
- Disk access
 - two types for data storage 1-6
- Disk failure, recovering from 1-17
- Disk I/O, role of shared memory in reducing 1-5
- Disk space
 - allocating 3-14
 - cooked 1-7
 - initializing 3-14
 - raw 1-7
 - types of physical units 1-6
- Distributed database
 - advantages of 2-17
 - and transactions 1-19
 - description of 2-16
 - disadvantages of 2-17
 - multiple vendor servers 2-18
- Distributed transaction-processing environment 2-18
- Documentation
 - for other Informix products, listed 4-15
 - for specific tasks 4-5
 - listed 4-11
 - on-line notes Intro-12
- Documentation conventions
 - icon Intro-8
 - typographical Intro-7
- Documentation notes Intro-12
- Documentation notes, program item Intro-12

Documentation set, listed 4-11
Documentation, types of
 documentation notes Intro-12
 error message files Intro-11
 machine notes Intro-12
 on-line help Intro-11
 on-line manuals Intro-10
 printed manuals Intro-10
 related reading Intro-13
 release notes Intro-12
Driver manager, described 3-8
Driver, INFORMIX-CLI 3-8
DTP environment. *See* Distributed
 transaction-processing
 environment.

E

Enterprise Replication,
 described 2-7
Environment variable
 CLIENT_LOCALE 3-11
 DB_LOCALE 3-11
 for client 3-11
 how to set 3-11
 INFORMIXDIR 3-10
 INFORMIXSERVER 3-10
 INFORMIXTERM 3-11
 ONCONFIG 3-10
 PATH 3-10
 required 3-10
 SERVER_LOCALE 3-11
 TERM 3-10
 TERMINFO 3-11
 useful 3-11
en_us.8859-1 locale Intro-5
Error messages
 files Intro-11
 getting information 3-18
Expression-based
 fragmentation 1-14

F

Failure
 disk, recovering from 1-17
 media, recovering from 1-17
 system, and ON-Bar 2-14
Fast recovery, description of 1-16
Fault tolerance
 archives and backups 1-15
 data replication 1-17, 2-6
 fast recovery 1-16
 mirroring 1-16
Feature icons Intro-9
File
 configuration 3-11
 connectivity 3-12
 database server security 1-21
 error message 3-18
 logical-log 2-12
 network configuration 3-13
 network security 3-13
 ONCONFIG 3-10
 ONCONFIG, preparing 3-13
 sqlhosts 3-12
 UNIX operating system 1-7
finderr script Intro-11, 3-18
Fragmentation
 described 1-13
 expression-based 1-14
 round-robin 1-14

G

Getting error message
 information 3-18
Global Language Support (GLS)
 defined Intro-5
 described 2-20
 GLS library 2-20
 locales 2-20
 using nondefault locales 3-15
GLS. *See* Global Language Support.

H

Hash joins 2-8
Heterogeneous commit protocol,
 described 1-20
High-availability data replication,
 described 2-6
High-performance loader
 described 2-9
 utilities for 2-9
hosts file 3-13

I

Icons
 comment Intro-8
 cross-reference Intro-8
 feature Intro-9
 platform Intro-9
 product Intro-9
IECC. *See* Informix Enterprise
 Command Center.
Industry standards, compliance
 with Intro-14
Informix Enterprise Command
 Center
 connecting with 3-16
 described 3-5
Informix Find Error
 utility Intro-11, 3-18
Informix SNMP subagent,
 described 3-6
Informix Storage Manager (ISM),
 described 2-14
INFORMIX-CLI, described 3-8
INFORMIXDIR environment
 variable 3-10
INFORMIXDIR/bin
 directory Intro-5
INFORMIX-ESQL/C,
 described 3-7
INFORMIX-GLS
 library 3-7
INFORMIX-GLS, described 3-7
INFORMIX-OnLine Dynamic
 Server
 See also Database server.

INFORMIXSERVER environment
variable 3-10
INFORMIXSQLHOSTS
environment variable 3-13
Initializing disk space 3-14
Installing Informix products 3-8
Integrity. *See* Data integrity.
Interfaces supported 3-5
Interprocess communication (IPC)
connection types 1-9
shared memory for 1-5
IPC. *See* Interprocess
communication.
ipload utility 4-8
IPX/SPX. *See* Network
communication.
ISO 8859-1 code set Intro-5
Isolation level
default 2-21
definition of 2-21

J

Join, hash 2-8

L

Library
GLS 2-20
INFORMIX-CLI 3-8
INFORMIX-GLS 3-7
TP/XA 2-18
Locale
assumptions about Intro-5
default Intro-5
en_us.8859-1 Intro-5
GLS 2-20
Locking, definition of 2-21
Logging. *See* Transaction logging.
Logical log, purpose of 1-15
Logical units of storage, list of 1-6
Logical volume manager as
mirroring alternative 1-17
Logical-log files, description
of 2-12

M

Machine notes Intro-12
Manuals
for additional Informix products,
listed 4-15
listed 4-11
Manuals. *See* Documentation.
Media failure
recovering from 1-17
restoring data 1-15
Memory management
buffering transactions 1-13
dynamically sharing
memory 1-12
Migration, information on 3-4
Mirroring
description of 1-16
logical volume manager
alternative 1-17
RAID alternative 1-17
Mirroring, description of 1-16
Multithreaded processes,
description of 1-7

N

Network
configuration files 3-13
connection information 3-12
security files 3-13
Network communication
using IPX/SPX 1-9
using TCP/IP 1-9
New features, Version 7.3 Intro-6

O

ON-Archive
description of 2-13
ontape compatibility 2-13
onaudit utility 2-10, 4-8
ON-Bar, described 2-14
oncheck utility 2-10
oncheck utility 4-6
ONCONFIG configuration file 3-10
ONCONFIG environment
variable 3-10

ondblog utility 2-10
oninit utility 2-10, 2-11, 4-6
On-line help Intro-11
On-line manuals Intro-10
onload utility 2-10
onlog utility 2-10, 4-6
onmode utility 2-10, 4-6, 4-7
onparams utility 2-11, 4-6
onperf utility 1-19, 2-11, 4-7
onpload utility 2-9, 4-8
onshowaudit utility 2-11, 4-8
onspaces utility 2-11, 4-6
onstat utility 2-11, 4-6
ontape utility 2-11, 4-6
compatibility with
ON-Archive 2-13
described 2-12
onunload utility 2-11, 2-15
Operating system files, UNIX 1-7
Operational data store,
described 2-19
Optical Subsystem, described 3-6
Optimizer, cost-based 2-8

P

Parallel processing,
mentioned 1-13
Parallelization, described 1-14
passwd file 3-13
PATH environment variable 3-10
PDQ (Parallel Database Query),
mentioned 1-14
Performance
advantages of raw-disk
management 1-12
and shared memory 1-5
Performance tuning,
mechanisms 1-11
Physical units of storage, list of 1-6
Pipes
stream 1-9
unnamed 1-9
Platform icons Intro-9
Point-in-time recovery,
described 1-18
Preparing the ONCONFIG file 3-13
Preparing the sqlhosts file 3-12

- Printed manuals Intro-10, 4-11
- Processes, compared to threads 1-7
- Product icons Intro-9
- Product installation 3-8
- Products, supported 3-4
- Program group
 - documentation notes Intro-12
 - release notes Intro-12
- Programmer, role of 4-4
- Protocol. *See* Network communication.

Q

- Query, definition of 2-3

R

- RAID as mirroring alternative 1-17
- Raw disk
 - management 1-12
 - space, description of 1-7
- RDBMS. *See* Relational database management system.
- Recovery
 - from media failure 1-17
 - ON-Bar 2-14
 - point-in-time 1-18
- Registry
 - configuration files in 3-12
 - setting environment variables in 3-11
- Related reading Intro-13
- Relational database
 - components of 1-3, 2-3
 - definition of 2-3
- Relational database management system (RDBMS) 1-3
- Release notes Intro-12
- Release notes, program item Intro-12
- Repository, described 2-19
- Requirements
 - ANSI 2-15
 - basic configuration 3-9
 - UNIX security 1-21
- Resource manager, in DTP environment 2-18

- Restore
 - description of 2-12
 - Informix Storage Manager 2-14
 - ON-Bar 2-14
 - point-in-time 1-18
- .rhosts file 3-13
- rofferr script Intro-11, 3-18
- Role separation, description of 2-22
- Roll back, definition of 2-22
- Roll back, mentioned 1-16
- Roll forward, mentioned 1-16
- ROLLBACK WORK statement 2-22
- Round-robin fragmentation 1-14

S

- Salvaging logical-log files 2-12
- Script
 - finderr 3-18
 - installation 3-9
 - rofferr 3-18
- SDK products supported 3-6
- Security
 - database server 1-21
 - files for network 3-13
 - role separation 2-22
 - with stored procedures 1-21
- Server. *See* Database server.
- services file 3-13
- SET CONNECTION
 - statement 3-17
- SET LOG statement 2-22
- SET PDQPRIORITY statement 1-15
- SET TRANSACTION
 - statement 2-21
- Setting environment variables 3-11
- Shared data 1-5
- Shared memory
 - dynamic, management of 1-12
 - for interprocess communication 1-5
 - interface 1-9
 - performance 1-5
 - purposes of 1-5
- SNMP, described 3-6
- Sockets 1-9
- Software dependencies Intro-4
- SQL API
 - compatibility with database servers 3-4
 - configuring environment for 3-15
- SQL statement
 - CONNECT 3-17
 - DATABASE 3-17
 - DISCONNECT 3-17
 - for database server security 1-21
 - SET CONNECTION 3-17
- sqlhosts file
 - changing location or name of 3-13
 - description of 3-12
- Stored procedure, and security 1-21
- stores7 database Intro-5
 - overview 3-17
- Stream pipes 1-9
- Structured Query Language (SQL) and client applications 1-8
 - and initial connection to database server 3-16
 - description of 2-3
- Support
 - buffered logging 2-22
 - client/server configurations 1-10
 - connection types 1-9
 - data replication 2-6
 - data types 2-4, 2-5
 - data warehouse 2-18
 - database 2-15
 - global language
 - feature Intro-5, 2-20
 - high-performance loader 2-9
 - interface and client product 3-4
 - logical units 1-6
 - manuals for products 4-15
 - multimedia 1-18
 - physical units 1-6
 - TEXT and BYTE on optical platter 3-6
 - utilities 2-10
 - utilities for database server 2-10
- Symmetric multiprocessing, description of 1-11
- Synchronous data replication 2-6
- System catalog, description of 2-4
- System failure, and ON-Bar 2-14

T

Table fragmentation. *See* Fragmentation.
Task-documentation matrix 4-5
Tasks
 administrative 3-19
 basic, listed 3-3
 documentation for 4-5
 for application developers 4-4
 for database administrators 4-4
 for database server administrators 4-4
 for database server operators 4-4
 for database users 4-3
 for programmers 4-4
TCP/IP. *See* Network communication.
TERM environment variable 3-10
Thread
 description of 1-7
 dynamic allocation of 1-13
 for client applications 1-7
TLI. *See* Transport level interface.
TP/XA
 described 2-18
 library 2-18
 when to use 2-18
Transaction
 definition of 2-16
 distributed 2-16
 rolling back 2-22
Transaction logging
 buffered 2-22
 effect on database server type 2-22
 using SET LOG statement 2-22
Transport level interface (TLI) 1-9

U

UNIX operating system
 default locale for Intro-5
 directory for on-line files Intro-12
 files 1-7
 reading error messages Intro-11
 security requirements 1-21
 standard environment variables 3-10
Unnamed pipes 1-9
Utilities
 for auditing 4-14
 for migration 4-13
 listed 2-10
 not supported by editions 2-11
 not to use for recovery 2-15
Utility
 chkenv 3-11
 DB-Access 3-5, 3-17, 4-10, 4-11
 dbexport 2-10, 2-15, 4-7
 dbimport 2-10, 4-7
 dbload 2-10, 4-7
 dbschema 2-10, 4-7
 finderr Intro-11
 Informix Find Error Intro-11, 3-18
 ipload 2-9, 4-8
 onaudit 2-10, 4-8
 oncheck 2-10, 4-6
 ondblog 2-10
 oninit 2-10, 2-11, 4-6
 onload 2-10
 onlog 2-10, 4-6
 onmode 2-10, 4-6, 4-7
 onparams 2-11, 4-6
 onperf 1-19, 2-11, 4-7
 onpload 2-9, 4-8
 onshowaudit 2-11, 4-8
 onspaces 2-11, 4-6
 onstat 2-11, 4-6
 ontape 2-11, 2-12, 2-13, 4-6
 onunload 2-11, 2-15
 rofferr Intro-11

V

Virtual processor, description of 1-7

W

Windows NT
 default locale for Intro-5
 program groups for on-line notes Intro-12
 reading error messages Intro-11

X

XA interface standard, adherence to 2-18
X/Open compliance Intro-14