DB2 IBM Informix

Version 10.0

IBM Informix Dynamic Server Getting Started Guide

G251-2281-02

DB2 IBM Informix

IBM

Version 10.0

IBM Informix Dynamic Server Getting Started Guide

G251-2281-02

Note:
Before using this information and the product it supports, read the information in "Notices" on page C-1.

Third Edition (December 2005)

This document contains proprietary information of IBM. It is provided under a license agreement and is protected by copyright law. The information contained in this publication does not include any product warranties, and any statements provided in this manual should not be interpreted as such.

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

© Copyright International Business Machines Corporation 1996, 2005. All rights reserved.
US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Introduction vii
In This Introduction vii
About This Manual vii
Types of Users vii
Software Dependencies viii
Assumptions About Your Locale viii
Demonstration Database viii
New Features in Version 10.0 ix
Organization of This Manual ix
Documentation Conventions x
Typographical Conventions x
Feature, Product, and Platform Markup x
Syntax Diagrams xi
Example Code Conventions xiv
Additional Documentation xv
IBM Informix Information Center xv
Installation Guides xv
Online Notes xv
Informix Error Messages xvii
Manuals xvii
Online Help xvii
Accessibility xviii
Compliance with Industry Standards xviii
IBM Welcomes Your Comments xviii

Chapter 1. Introducing Dynamic Server and Client Products 1-1
In This Chapter 1-1
IBM Informix Dynamic Server 1-1
3 Edition Comparison 1-2
Installation and Migration 1-2
Products Bundled with the Database Server 1-2
BladeManager 1-2
DataBlade API 1-3
DataBlade Developers Kit 1-3
DB-Access 1-3
High-Performance Loader 1-3
IBM Informix Connect 1-4
IBM Informix Server Administrator (ISA) 1-4
IBM Informix SNMP Subagent 1-4
Optical Subsystem 1-4
Related IBM Informix Products 1-4
Client SDK Products 1-4
IBM Informix MaxConnect (UNIX) 1-6
IBM Office Connect 1-6
Server Studio Java Edition by AGS 1-6
IBM Informix Data Director for Web 1-7
DataBlade Modules 1-7
Related IBM Products 1-7
Global Language Support 1-9

Chapter 2. Using New Features in Dynamic Server 2-1
In This Chapter 2-4
New Features in Version 10.00 of IBM Informix Dynamic Server 2-4
4 New Features in Version 10.00.xC4 2-4
3 New Features in Version 10.00.xC3 2-5

© Copyright IBM Corp. 1996, 2005

New Features in Version 10.00.xC1 2-8
New Features in Version 9.4 2-18
Security Enhancement 2-18
Database Server Usability Enhancements 2-19
Performance Enhancements 2-21
Enterprise Replication Enhancements 2-22
Extensibility Enhancements 2-23
SQL Enhancements 2-25
GLS Enhancements 2-28
Reliability, Availability, and Supportability Features 2-29
DataBlade API Enhancements. 2-29
High-Performance Loader Enhancements 2-30
Backup and Restore Enhancements 2-31
Installation Enhancements 2-31
Changed or New URLs 2-32
New Features in Version 9.3 2-32
UNIX Bundle Installer 2-33
Database Server Usability Enhancements 2-33
DataBlade API Enhancements. 2-34
Enterprise Replication Enhancements 2-35
Extensibility Enhancements 2-38
J/Foundation Enhancements 2-39
Performance Enhancements 2-40
SQL Enhancements 2-41
Features from Dynamic Server, Version 9.21 2-41
ANSI Join Syntax 2-41
Rename Index Statement 2-42
Nonlogging (RAW) Tables 2-42
onpladm Utility 2-42
The onbar -b -l Command 2-42
9.x DB-Access to 7.x Synonyms 2-42
SQL Statement Cache Improvements 2-42
DataBlade API Features. 2-43
Java Features in 9.21 2-44
MaxConnect Support 2-45

Chapter 3. Using Existing Dynamic Server Features 3-1
In This Chapter 3-2
Dynamic Scalable Architecture 3-2
The Shared-Memory Component 3-2
The Disk Component 3-3
The Virtual Processor Component. 3-4
Client/Server Connections 3-4
High Performance 3-5
Memory Management 3-5
Parallelization 3-6
Query Optimizer 3-6
Fault Tolerance and High Availability 3-6
Backup and Restore 3-6
Fast Recovery 3-8
Mirroring 3-8
Data Replication 3-8
Database Server Security. 3-9
Informix RDBMS Features 3-10
Structured Query Language (SQL) 3-10
Stored Procedure Language (SPL) 3-10
System Catalog Tables 3-10
Data Types 3-11
Application Types 3-13
OLTP Applications 3-13
DSS Applications 3-13

Database Support 3-14
Relational Databases 3-14
ANSI-Compliant Databases 3-14
Object-Relational Databases 3-14
Dimensional Databases 3-18
Distributed Databases and Queries 3-19
Access Methods 3-19
Primary Access Methods 3-19
Secondary Access Methods 3-19
User-Defined Primary Access Methods 3-20
User-Defined Secondary Access Methods 3-20

Chapter 4. Installing, Administering, and Tuning the Database Server 4-1
In This Chapter 4-1
Database Server Users 4-1
Planning, Installing, and Configuring the Database Server 4-1
Administering the Database Server 4-3
Monitoring Performance 4-6
Troubleshooting the Database Server 4-7

Chapter 5. Designing, Maintaining, and Extending the Database 5-1
In This Chapter 5-1
Designing, Developing, and Extending the Database 5-1
Developing Application Programs that Access the Database 5-3

Chapter 6. Using the Documentation 6-1
In This Chapter 6-1
The IBM Informix Documentation Set 6-1
IBM Informix Dynamic Server Manuals 6-1
Client SDK and Connectivity Manuals 6-3
DataBlade Manuals 6-4

Appendix A. Database Server Utilities A-1

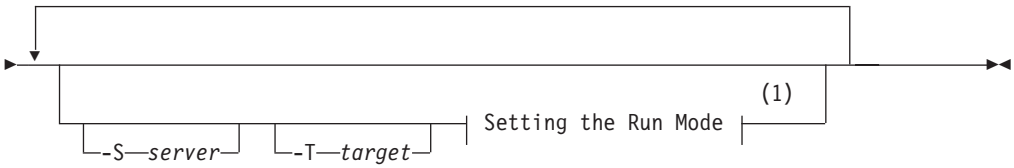
Appendix B. Accessibility B-1

Notices C-1

Index X-1

Introduction

In This Introduction vii
About This Manual vii
Types of Users vii
Software Dependencies viii
Assumptions About Your Locale viii
Demonstration Database ix

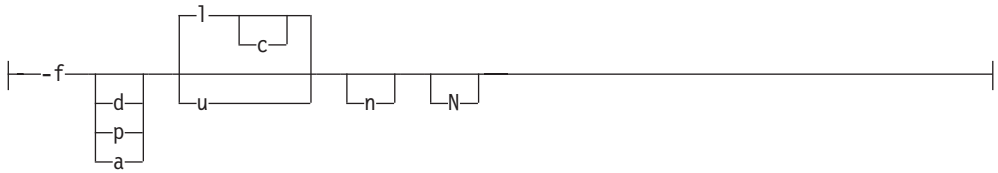


Notes:

1 See page Z-1

The second line in this diagram has a segment named "Setting the Run Mode," which according to the diagram footnote, is on page Z-1. If this was an actual cross-reference, you would find this segment in on the first page of Appendix Z. Instead, this segment is shown in the following segment diagram. Notice that the diagram uses segment start and end components.

Setting the Run Mode:



To see how to construct a command correctly, start at the top left of the main diagram. Follow the diagram to the right, including the elements that you want. The elements in this diagram are case sensitive because the illustrates utility syntax. Other types of syntax, such as SQL, are not case sensitive.

The Creating a No-Conversion Job diagram illustrates the following steps:

1. Type **onpladm create job** and then the name of the job.
2. Optionally, type **-p** and then the name of the project.
3. Type the following required elements:
   - **-n**
   - **-d** and the name of the device
   - **-D** and the name of the database
   - **-t** and the name of the table
4. Optionally, you can choose one or more of the following elements and repeat them an arbitrary number of times:
   - **-S** and the server name
   - **-T** and the target server name
   - The run mode. To set the run mode, follow the Setting the Run Mode segment diagram to type **-f**, optionally type **d**, **p**, or **a**, and then optionally type **l** or **u**.
5. Follow the diagram to the terminator.

Your diagram is complete.

Keywords and Punctuation

Keywords are words reserved for statements and all commands except system-level commands. When a keyword appears in a syntax diagram, it is

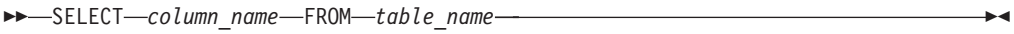
shown in uppercase letters. When you use a keyword in a command, you can write it in uppercase or lowercase letters, but you must spell the keyword exactly as it appears in the syntax diagram.

You must also use any punctuation in your statements and commands exactly as shown in the syntax diagrams.

Identifiers and Names

Variables serve as placeholders for identifiers and names in the syntax diagrams and examples. You can replace a variable with an arbitrary name, identifier, or literal, depending on the context. Variables are also used to represent complex syntax elements that are expanded in additional syntax diagrams. When a variable appears in a syntax diagram, an example, or text, it is shown in *lowercase italic*.

The following syntax diagram uses variables to illustrate the general form of a simple SELECT statement.



When you write a SELECT statement of this form, you replace the variables *column_name* and *table_name* with the name of a specific column and table.

Example Code Conventions

Examples of SQL code occur throughout this manual. Except as noted, the code is not specific to any single IBM Informix application development tool.

If only SQL statements are listed in the example, they are not delimited by semicolons. For instance, you might see the code in the following example:

```
CONNECT TO stores_demo
...

DELETE FROM customer
   WHERE customer_num = 121
...

COMMIT WORK
DISCONNECT CURRENT
```

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

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

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

Additional Documentation

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

- Installation guides
- Online notes
- Informix error messages
- Manuals
- Online help

IBM Informix Information Center

The Informix Dynamic Server Information Center integrates the entire IBM Informix Dynamic Server 10.0 and IBM Informix Client SDK (CSDK) 2.90 documentation sets in both HTML and PDF formats. The Information Center provides full text search, a master index, logical categories, easy navigation, and links to troubleshooting and support files.

The IBM Informix Information Center site is located at http://publib.boulder.ibm.com/infocenter/ids9help/index.jsp.

Installation Guides

Installation guides are located in the **/doc** directory of the product CD or in the **/doc** directory of the product's compressed file if you downloaded it from the IBM Web site. Alternatively, you can obtain installation guides from the IBM Informix Online Documentation site at http://www.ibm.com/software/data/informix/pubs/library/ or the IBM Informix Information Center at http://publib.boulder.ibm.com/infocenter/ids9help/index.jsp.

Online Notes

The following sections describe the online files that supplement the information in this manual. Please examine these files before you begin using your IBM Informix product. They contain vital information about application and performance issues.

Online File	Description	Format
TOC Notes	The TOC (Table of Contents) notes file provides a comprehensive directory of hyperlinks to the release notes, the fixed and known defects file, and all the documentation notes files for individual manual titles.	HTML
Documentation Notes	The documentation notes file for each manual contains important information and corrections that supplement the information in the manual or information that was modified since publication.	HTML, text
Release Notes	The release notes file describes feature differences from earlier versions of IBM Informix products and how these differences might affect current products. For some products, this file also contains information about any known problems and their workarounds.	HTML, text

Online File	Description	Format
Machine Notes	(Non-Windows platforms only) The machine notes file describes any platform-specific actions that you must take to configure and use IBM Informix products on your computer.	text
Fixed and Known Defects File	This text file lists issues that have been identified with the current version. It also lists customer-reported defects that have been fixed in both the current version and in previous versions.	text

Locating Online Notes

Online notes are available from the IBM Informix Online Documentation site at http://www.ibm.com/software/data/informix/pubs/library/ and in the IBM Informix Information Center at http://publib.boulder.ibm.com/infocenter/ids9help/index.jsp. Additionally you can locate these files before or after installation as described below.

Before Installation

All online notes are located in the **/doc** directory of the product CD. The easiest way to access the documentation notes, the release notes, and the fixed and known defects file is through the hyperlinks from the TOC notes file.

The machine notes file and the fixed and known defects file are only provided in text format.

After Installation

On UNIX platforms in the default locale, the documentation notes, release notes, and machine notes files appear under the **$INFORMIXDIR/release/en_us/0333** directory.

Dynamic Server

On Windows the documentation and release notes files appear in the **Informix** folder. To display this folder, choose **Start > Programs > IBM *product name version* > Documentation Notes** or **Release Notes** from the taskbar.

Machine notes do not apply to Windows platforms.

End of Dynamic Server

Online Notes Filenames

Online notes have the following file formats:

Online File	File Format	Examples
TOC Notes	*prod_os*_toc_*version*.html	**ids_win_toc_10.0.html**
Documentation Notes	*prod_bookname*_docnotes_*version*.html/txt	**ids_hpl_docnotes_10.0.html**
Release Notes	*prod_os*_relnotes_*version*.html/txt	**ids_unix_relnotes_10.0.txt**
Machine Notes	*prod*_machine_notes_*version*.txt	**ids_machine_notes_10.0.txt**

Online File	File Format	Examples
Fixed and Known Defects File	*prod*_defects_*version*.txt	**ids_defects_10.0.txt** **client_defects_2.90.txt**
	ids_win_fixed_and_known _defects_*version*.txt	**ids_win_fixed_and_known _defects_10.0.txt**

Informix Error Messages

This file is a comprehensive index of error messages and their corrective actions for the Informix products and version numbers.

On UNIX platforms, use the **finderr** command to read the error messages and their corrective actions.

Dynamic Server

On Windows, use the Informix Error Messages utility to read error messages and their corrective actions. To display this utility, choose **Start > Programs > IBM *product name version* > Informix Error Messages** from the taskbar.

End of Dynamic Server

You can also access these files from the IBM Informix Online Documentation site at http://www.ibm.com/software/data/informix/pubs/library/ or in the IBM Informix Information Center at http://publib.boulder.ibm.com/infocenter/ids9help/index.jsp.

Manuals

Online Manuals

A CD that contains your manuals in electronic format is provided with your IBM Informix products. You can install the documentation or access it directly from the CD. For information about how to install, read, and print online manuals, see the installation insert that accompanies your CD. You can also obtain the same online manuals from the IBM Informix Online Documentation site at http://www.ibm.com/software/data/informix/pubs/library/ or in the IBM Informix Information Center at http://publib.boulder.ibm.com/infocenter/ids9help/index.jsp.

Printed Manuals

To order hardcopy manuals, contact your sales representative or visit the IBM Publications Center Web site at http://www.ibm.com/software/howtobuy/data.html.

Online Help

IBM Informix online help, provided with each graphical user interface (GUI), displays information about those interfaces and the functions that they perform. Use the help facilities that each GUI provides to display the online help.

Accessibility

IBM is committed to making our documentation accessible to persons with disabilities. Our books are available in HTML format so that they can be accessed with assistive technology such as screen reader software. The syntax diagrams in our manuals are available in dotted decimal format, which is an accessible format that is available only if you are using a screen reader. For more information about the dotted decimal format, see the Accessibility appendix.

Compliance with Industry Standards

The American National Standards Institute (ANSI) and the International Organization of Standardization (ISO) have jointly established a set of industry standards for the Structured Query Language (SQL). IBM 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 IBM Informix database servers comply with the SQL-92 Intermediate and Full Level and X/Open SQL Common Applications Environment (CAE) standards.

IBM Welcomes Your Comments

We want to know about any corrections or clarifications that you would find useful in our manuals, which will help us improve future versions. Include the following information:

- The name and version of the manual that you are using
- Section and page number
- Your suggestions about the manual

Send your comments to us at the following email address:

docinf@us.ibm.com

This email address is reserved for reporting errors and omissions in our documentation. For immediate help with a technical problem, contact IBM Technical Support. For instructions, see the IBM Informix Technical Support website at http://www-306.ibm.com/software/data/informix/support/contact.html.

We appreciate your suggestions.

Chapter 1. Introducing Dynamic Server and Client Products

In This Chapter 1-1
IBM Informix Dynamic Server 1-1
3 Edition Comparison 1-2
Installation and Migration 1-2
Products Bundled with the Database Server 1-2
BladeManager 1-2
DataBlade API 1-3
DataBlade Developers Kit 1-3
DB-Access 1-3
High-Performance Loader 1-3
IBM Informix Connect 1-4
IBM Informix Server Administrator (ISA) 1-4
IBM Informix SNMP Subagent 1-4
Optical Subsystem 1-4
Related IBM Informix Products 1-4
Client SDK Products 1-4
IBM Informix ESQL/C 1-4
IBM Informix ESQL/J Pre-Processor 1-5
IBM Informix GLS 1-5
IBM Informix JDBC Driver 1-5
IBM Informix Object Interface for C++ 1-5
IBM Informix ODBC Driver. 1-6
IBM Informix OLE DB Provider (Windows) 1-6
IBM Informix .NET Provider (Windows) 1-6
TP/XA 1-6
IBM Informix MaxConnect (UNIX) 1-6
IBM Office Connect 1-6
Server Studio Java Edition by AGS 1-6
IBM Informix Data Director for Web 1-7
DataBlade Modules 1-7
Related IBM Products. 1-7
Global Language Support 1-9

In This Chapter

This chapter provides an overview of IBM Informix Dynamic Server, Version 10.0, IBM Informix Client Software Developer's Kit, and related products. For a list of the manuals and description of each product, see "The IBM Informix Documentation Set" on page 6-1.

IBM Informix Dynamic Server

A *database server* is a software package that manages access to one or more databases for one or more client applications. Dynamic Server is a fast and scalable database server that manages traditional relational, object-relational, and web-based databases. Dynamic Server supports alphanumeric and rich data, such as graphics, multimedia, geospatial, HTML, and user-defined types. You can use Dynamic Server on UNIX, Linux, or Windows with online transaction processing (OLTP), data marts, data warehouses, and e-business applications.

You can write *user-defined routines* (UDRs) in Java, C, and stored procedure language (SPL). A UDR is a routine that an SQL statement, user-defined function, or user-defined procedure can invoke.

© Copyright IBM Corp. 1996, 2005

Edition Comparison

IBM Informix Dynamic Server is available in the following different editions:

- Enterprise Edition, for any business that needs the full functionality of Dynamic Server
- Workgroup Edition, for small and medium businesses
- Express Edition, for small and medium businesses that only need to use two CPUs for their database needs

The following table compares the features of each edition.

Table 1-1. Comparison of Dynamic Server Editions

Feature	Express Edition	Workgroup Edition	Enterprise Edition
CPUs	Limited to two CPUs	Limited to four CPUs More available with special per-CPU licensing	No limit to number of CPUs
High-Availability Data Replication	Not available	Available with special per-CPU licensing	Available feature
Enterprise Replication	Not available	Limited functionality: available as leaf node (target server) only	Available feature
Price point	Lowest price point	Variable price point	Highest price point

Installation and Migration

For information on how to install the database server products, see the *IBM Informix Dynamic Server Installation Guide for UNIX and Linux*, the *IBM Informix Dynamic Server Installation Guide for Microsoft Windows*, or the *Quick Beginnings for IBM Informix Dynamic Server Express Edition*.

If you migrate to Dynamic Server, Version 10.0, from an earlier version of the database server, start with the information provided in the *IBM Informix Migration Guide*.

Products Bundled with the Database Server

In addition, several products are included with the database server. This section discusses the IBM Informix products that help you manage the database server. The following products are bundled with Dynamic Server:

- BladeManager
- DataBlade Developer's Kit
- IBM Informix Connect
- IBM Informix Client SDK
- IBM Informix Server Administrator
- JDBC 3.0 with ESQL J 1.01
- Server Studio Java Edition by AGS

BladeManager

Use the BladeManager to register new DataBlade modules in Informix databases. BladeManager runs on client computers.

For more information, see the *IBM Informix DataBlade Module Installation and Registration Guide*.

DataBlade API

The DataBlade API is a C-language application programming interface that is provided with Dynamic Server. Experienced C programmers can use API functions in DataBlade modules to develop client and database server applications that access data stored in a database. The DataBlade API contains public data structures, public functions, and header files for DataBlade modules, ESQL/C, GLS, and so on.

For more information, see the *IBM Informix DataBlade API Programmer's Guide* and the *IBM Informix DataBlade API Function Reference*.

DataBlade Developers Kit

The DataBlade Developer's Kit includes the following tools for developing and packaging DataBlade modules:

- BladeSmith (organizes a DataBlade development project)
- DBDK Visual C++ Add-In and Ifx Query (debugs DataBlade modules)
- BladePack (creates a DataBlade package)
- BladeManager (registers and unregisters DataBlade modules)

For more information, see the *DataBlade Module Development Overview* and *IBM Informix DataBlade Developer's Kit User's Guide*.

DB-Access

DB–Access is a client tool that is included with the database server. DB–Access lets you connect to the database server and access, modify, and retrieve database data. To use DB–Access, type **dbaccess** from the command line.

For more information, see the *IBM Informix DB–Access User's Guide*.

High-Performance Loader

The High-Performance Loader (HPL) lets you efficiently load and unload large quantities of data to or from an Informix database. Use the HPL to exchange data with tapes, data files, and programs and to 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.

Use the **ipload** graphical user interface (GUI) to load and unload data, create, modify, and delete HPL objects on UNIX.

Use the **onpladm** utility to perform the same tasks from the command line on either UNIX or Windows platforms. The **ipload** and **onpladm** utilities are included with the database server.

For information on how to use the HPL, including tutorial examples, see the *IBM Informix High-Performance Loader User's Guide*. For information on other migration utilities, such as **dbexport** and **dbload**, see the *IBM Informix Migration Guide*.

IBM Informix Connect

IBM Informix Connect allows applications that run on client computers to access the database server.

IBM Informix Server Administrator (ISA)

IBM Informix Server Administrator (ISA) is a browser-based tool that provides web-based system administration for Informix database servers. ISA provides access to database server command-line functions and presents the output in an easy-to-read format.

For information on how to install ISA, see your *IBM Informix Installation Guide*. For information on how to use ISA, see the ISA online help and onscreen instructions.

IBM Informix SNMP Subagent

Simple Network Management Protocol (SNMP) is a published, open standard for network management that is included with the database server. The IBM Informix SNMP subagent provides information about Informix database servers to SNMP-compliant applications.

For more information, see the *IBM Informix SNMP Subagent Guide*.

Optical Subsystem

The Optical Subsystem is included with the database server and stores *simple large objects* (TEXT and BYTE data) on optical platters known as WORM optical media. The optical media are the removable optical platters that contain data. The Optical Subsystem includes a specific set of SQL statements that allow you to store and retrieve data stored on optical disks.

The Optical Subsystem does not store character large object (CLOB) and binary large object (BLOB) data types, also known as *smart large objects*.

For more information, see the *IBM Informix Optical Subsystem Guide*.

Related IBM Informix Products

This section discusses the related products that you can use with Dynamic Server. For information on ordering these products, contact your IBM sales representative.

Client SDK Products

The IBM Informix Client Software Developer's Kit (Client SDK) includes several application-programming interfaces (APIs) that developers can use to write applications for Informix database servers in ESQL, C, and Java. You can also write Informix ESQL/C applications for the DB2 database server. IBM Informix Connect contains the runtime libraries of the APIs in the Client SDK.

For more information, see your *IBM Informix Client Products Installation Guide*.

IBM Informix ESQL/C

ESQL/C is an SQL application programming interface (API) that lets programmers embed SQL statements directly into a C program to interact with the database server, access databases, manipulate the data in a program, and check for errors.

IBM Informix ESQL/C consists of the following components:

- ESQL/C libraries of C functions for accessing the database server
- ESQL/C header files, which provide definitions for the data structures, constants, and macros
- **esql**, a command that manages the source-code processing to convert a C file that contains SQL statements into an object file
- ESQL client-interface *dynamic link libraries* (DLLs), which let an ESQL/C application run on Windows

For more information, see the *IBM Informix ESQL/C Programmer's Manual*.

IBM Informix ESQL/J Pre-Processor

IBM Informix Embedded SQLJ enables you to embed SQL statements in your Java programs. It consists of the SQLJ translator, which translates SQLJ code into Java code, and a set of Java classes that provide runtime support for SQLJ programs. When you run an SQLJ program, it uses IBM Informix JDBC Driver to connect to an Informix database.

For more information, see the *IBM Informix Embedded SQLJ User's Guide* and "IBM Informix JDBC Driver" on page 1-5.

IBM Informix GLS

Global Language Support (GLS) allows you to create databases that use the diacritics, collating sequence, and monetary and time conventions of the language that you select. The IBM Informix GLS library contains APIs that let programmers develop internationalized ESQL/C and DataBlade module client applications. IBM Informix GLS is separately orderable, but the GLS libraries are shipped with the database server and Client SDK.

IBM Informix GLS provides procedures, macros, and functions to:

- Process single-byte and multibyte characters and strings.
- Convert date, time, monetary, and number values from and to locale-specific data formats.

For more information, see the *IBM Informix GLS User's Guide*. IBM Informix GLS provides an HTML reference that you can access with a web browser. The URL must include the full pathname of the directory that your **INFORMIXDIR** environment variable designates:
$INFORMIXDIR/doc/gls_api/en_us/0333/index.htm on UNIX or
%INFORMIXDIR%\doc\gls_api\en_us\04e4\index.htm on Windows.

IBM Informix JDBC Driver

IBM Informix JDBC Driver lets Java programmers access Informix databases from within Java applications or applets. Programmers can create client applications that use JDBC to connect to Dynamic Server, query and retrieve data from a database or column, handle errors, and write UDRs. The IBM Informix JDBC Driver is compatible with the JavaSoft JDBC specifications. It maps standard Java data types and Informix database server data types.

For more information, see the *IBM Informix JDBC Driver Programmer's Guide*.

IBM Informix Object Interface for C++

Use the IBM Informix Object Interface for C++ to develop IBM Informix client applications using the C++ programming language.

For more information, see the *IBM Informix Object Interface for C++ Programmer's Guide*.

IBM Informix ODBC Driver

IBM Informix ODBC Driver is the Informix implementation of the Microsoft Open Database Connectivity (ODBC) standard. It supports SQL statements with a library of C functions that an application calls to access Informix databases.

For more information, see the *IBM Informix ODBC Driver Programmer's Manual*.

IBM Informix OLE DB Provider (Windows)

IBM Informix OLE DB Provider enables OLE DB applications, such as Active Data Objects (ADO) applications and web pages, to access the database server.

For more information, see the *IBM Informix OLE DB Provider Programmer's Guide*.

IBM Informix .NET Provider (Windows)

The IBM Informix .NET Provider is a .NET assembly that lets .NET applications access and manipulate data in IBM Informix databases. It does this by implementing several interfaces in the Microsoft .NET Framework that are used to access data from a database.

For more information, see the *IBM Informix .NET Provider Reference Guide*.

TP/XA

The TP/XA library facilitates communication between a third-party transaction manager and your database server. TP/XA is supplied with IBM Informix ESQL/C. Use TP/XA for distributed transaction processing in a multivendor database setting.

For more information, see the *IBM Informix TP/XA Programmer's Manual*.

IBM Informix MaxConnect (UNIX)

IBM Informix MaxConnect is a networking product for Informix database servers on UNIX. Two protocols for multiplexing connections, **ontliimc** and **onsocimc**, are available for MaxConnect users. MaxConnect manages large numbers (from several hundred to tens of thousands) of client/server connections. The ratio of client connections to database connections can be 100:1 or higher. MaxConnect increases system scalability to many thousands of connections and saves system resources, reducing response times and CPU requirements. You can install MaxConnect on the client application server, on a dedicated server, or on the database server computer.

For more information, see the *IBM Informix MaxConnect User's Guide*.

IBM Office Connect

IBM Office Connect enables your Excel worksheets to access, display, and modify data from Informix and other ODBC databases.

For more information, see the *IBM Informix Connect User's Guide*.

Server Studio Java Edition by AGS

Server Studio Java Edition by AGS (formerly known as Server Studio JE) is a stand-alone, Java-based Integrated Development Environment (IDE) for the 7.3x, 9.x, and 10.x database servers. Server Studio contains the following modules:

- Database Object Explorer
- Properties Inspector
- SQL Editor
- Table Editor

These modules are provided free of charge. Additional modules are provided with the distribution of Server Studio Java Edition by AGS on a try and buy basis. Contact Advanced Global Systems Ltd. (AGS) to obtain a license for the additional modules at www.agsltd.com.

Server Studio replaces IBM Informix Database Administrator.

IBM Informix Data Director for Web

IBM Informix Data Director for Web provides a model-driven development environment designed explicitly for creating powerful database applications that can grow with your business, addressing both evolving enterprise needs and increasingly diverse technical requirements. Data Director for Web has the following capabilities:

- Automates all of the data access operations of the client application
- Eliminates the task of writing data-access code
- Allows developers to easily incorporate sophisticated functionality without having to be database programming experts
- Helps project teams improve time to market with scalable applications that solve real business problems
- Enables interactive Web sites with Informix Web Datablade module

DataBlade Modules

DataBlade modules extend the capabilities of Dynamic Server with user-defined objects. Available DataBlade modules include:

- IBM Informix Image Foundation DataBlade module
- IBM Informix Excalibur Text Search DataBlade module
- IBM Informix Geodetic DataBlade module
- IBM Informix Spatial DataBlade module
- IBM Informix TimeSeries DataBlade module
- IBM Informix TimeSeries Real Time Loader DataBlade module
- IBM Informix Video Foundation DataBlade module
- IBM Informix Web DataBlade module

For a brief description of each of these, see “DataBlade Manuals” on page 6-4.

Related IBM Products

The following IBM products are among those that you can use with IBM Informix Dynamic Server:

- WebSphere Application Server

  A J2EE and Web Services application server that controls interactions with enterprise information systems, including IBM Informix Dynamic Server.
- WebSphere Portal

Enables you to develop and maintain B2C, B2B and B2E portals. Portals provide a mechanism for aggregating information and accessing enterprise services via a single consolidated view for Web usage.

- WebSphere Studio tools

  Provides a flexible, portal-like integration of multi-language, multi-platform and multi-device application development tools for building, testing and deploying dynamic applications.

- WebSphere MQ

  Connects applications and Web Services by exchanging information, including XML documents and SOAP messages, across different platforms. Supports the latest Java Messaging Service interface standard, JMS v1.1.

- Enterprise Generation Language (EGL)

  A development environment and programming language that lets you write full-function applications quickly.

- WORF

  Provides an environment to create XML-based Web services that access DB2.

- Tivoli Storage Manager

  Tivoli Storage Manager for Databases for Multiplatforms (Informix Interface for Tivoli Storage Manager) provides backup capabilities for restoring and managing Informix databases and logical logs. The Informix ON-Bar utility uses the X/Open Backup Services Application Program Interface (XBSA) to communicate with the Tivoli Storage Manager, where backups are stored.

- Tivoli TME 10 NetView

  Acts as an SNMP Network Manager with IBM Informix Dynamic Server.

- Tivoli Monitoring for Databases

  Tivoli Monitoring for Databases helps increase database availability, providing proactive alerts to potential problems before they arise.

- DB2 Web Query Tool

  Connects users directly to multiple enterprise databases, including IBM Informix Dynamic Server.

- DB2 Table Editor

  Enables you to edit DB2 tables from within Java-enabled Web browsers, Java-interfaces launched from IBM DB2 Control Center or from Microsoft Windows, or the ISPF interface in z/OS.

- DB2 Information Integrator

  Uses federated servers to provide a single abstract view of diverse backend information sources. You can integrate data in real time from these different backends, update that data, and have the updates reflected in the backend.

- DB2 Information Integrator for Content

  DB2 Information Integrator for Content is an information integration infrastructure that allows users to access data and content from multiple sources.

- DB2 DataJoiner

  DB2 DataJoiner makes diverse and distributed relational databases appear as a single database, and supports replication among mixed relational databases.

- DB2 Life Sciences Data Connect

  DB2 Life Sciences Data Connect integrates genetic, chemical, biological and other data from distributed sources.

- DB2 Relational Connect

Queries and retrieves information from IBM Informix Dynamic Server and other database servers.

- DB2 Content Manager

  Provides transaction, security, process integration, and life cycle services to manage digitized content, including HTML and XML Web content, document images, electronic office documents, printed output, and digital audio and video.

- Lotus Notes

  Lotus Notes combines enterprise-class messaging, calendaring and scheduling capabilities with a robust platform for collaborative applications.

- Lotus Domino

  Lotus Domino provides a multi-platform foundation for collaboration and e-business, driving solutions from corporate messaging to Web-based transactions.

- Rational Unified Process

  Rational Unified Process combines a core set of industry-proven best practices with optional process plug-ins to support development projects of any size or scope.

- Rational XDET

  Rational XDET, a complete visual design and development environment, offers software developers a model-driven development approach to building applications, including Web-centric solutions.

- Rational ProjectConsole

  Rational ProjectConsole, an integral part of every IBM Rational Suite, helps monitor project progress, identify trends and improve project predictability by creating a Web site with assets and metrics automatically collected from your development environment.

- Rational Suite AnalystStudio

  Rational Suite AnalystStudio is a powerful, integrated set of tools for business, application and data modeling requirements.

- Rational RequisitePro

  Rational RequisitePro, a powerful, easy-to-use, integrated product for requirements and use-case management, promotes better communication, enhances teamwork and reduces project time.

- Rational Rapid Developer

  Rational Rapid Developer is a single, integrated application development environment that combines model-driven development, architected RAD techniques and automated construction to develop, integrate, deploy and maintain high-quality n-tier business applications.

Global Language Support

The Global Language Support (GLS) feature lets the database server handle different languages, cultural conventions, and code sets using 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 viii of the Introduction.

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 IBM Informix product.

For more information about the GLS feature, see the *IBM Informix GLS User's Guide*.

Chapter 2. Using New Features in Dynamic Server

In This Chapter 2-4
New Features in Version 10.00 of IBM Informix Dynamic Server 2-4
4 New Features in Version 10.00.xC4 2-4
4 TRUNCATE Table support 2-4
4 Enterprise Replication Direct Synchronization 2-4
4 Enterprise Replication Consistency Checking 2-4
4 Enhanced Support for IPv6 2-4
4 Secure Local Connections 2-4
4 Secure DataBlade Module Paths 2-5
4 Parallel Backup and Restore is More Efficient 2-5
4 DB-Access Stops a Process After the First Error 2-5
4 Informix Interface for Tivoli Storage Manager Supports HP-UX (Itanium) 2-5
4 New Default Value for IFX_EXTEND_ROLE Configuration Parameter 2-5
3 New Features in Version 10.00.xC3 2-5
3 ANSI-Joins in Distributed Queries 2-6
3 Transaction Support for XA-Compliant External Data Sources 2-6
3 MQ DataBlade Module 2-6
3 Restricting Database Creation 2-6
3 Secure DUMPDIR Configuration Parameter Default Directory 2-6
3 Table-Level Restore Support for Smart Large Objects 2-6
3 AES Ciphers for Network Encryption 2-6
3 Enterprise Replication Statistics Commands 2-7
3 CSDK Included in Dynamic Server Installation 2-7
3 Enhanced Support for Retrieving Subsets of Query Results 2-7
3 Returning Subsets of Query Results as Collection-Derived Tables 2-7
3 J/Foundation Upgraded to JRE 1.4.2 2-7
3 Secure ADTPATH Configuration Parameter Default Directory 2-7
3 UNSECURE_ONSTAT Configuration Parameter 2-7
New Features in Version 10.00.xC1 2-8
Security Enhancements 2-8
Server Usability Enhancements 2-9
Performance Enhancements 2-12
SQL Enhancements 2-14
Enterprise Replication Enhancements 2-14
Backup and Restore Enhancements 2-16
Storage Enhancements 2-16
Extensibility Enhancements 2-17
Installation Enhancements 2-17
New installation program on UNIX and Linux 2-18
Interoperability Enhancements 2-18
New Features in Version 9.4 2-18
Security Enhancement 2-18
Database Server Usability Enhancements 2-19
Increase Size of Chunks, Chunk Offsets, and Number of Allowable Chunks 2-19
Configurable Event Alarms 2-19
Increased Database Server Aliases 2-19
Increased File Size Limit 2-19
Full Use of Storage Media 2-20
Increased Default Values for Tape Block Size Configuration Parameters 2-20
Chunk Reserve Pages in Non-Root Chunks 2-20
Restartable Fast Recovery 2-20
Microsoft Transaction Server/XA Support 2-21
Performance Enhancements 2-21
PDQ is Enabled for Hold Cursors 2-21
Improved Transaction Processing with the B-tree Scanner 2-21

© Copyright IBM Corp. 1996, 2005

Improved Priority Management for the Buffer Manager . . . 2-21
Spatial Query Costing . . . 2-21
More Precise LRU Maximum and Minimum Settings . . . 2-22
Enterprise Replication Enhancements . . . 2-22
Enterprise Replication Security . . . 2-22
Support for ROW and Collection Data types . . . 2-22
Faster Queue Recovery . . . 2-22
Replication During Queue Recovery . . . 2-22
Large Transactions Support . . . 2-22
Improved Availability with HDR . . . 2-22
Dynamic Log File . . . 2-22
New Commands . . . 2-23
New and Altered Configuration Parameters . . . 2-23
New Environment Variables . . . 2-23
Extensibility Enhancements . . . 2-23
Enhanced HDR Support for Extensibility Features . . . 2-23
Using an Iterator Function in the FROM Clause of a SELECT Statement . . . 2-24
Enhanced CREATE FUNCTION and CREATE PROCEDURE Syntax . . . 2-24
SQL Enhancements . . . 2-25
INSTEAD OF Triggers on Views . . . 2-25
Enhanced SELECT Statement Syntax . . . 2-25
Functional Indexes on More Than 16 Columns . . . 2-25
Enhanced Dynamic Query Support . . . 2-25
Session-Level Non-Default Collation . . . 2-26
LOAD TO and UNLOAD FROM with Large Files . . . 2-26
SET Residency Statements No Longer Needed . . . 2-26
Multiple OUT Parameters . . . 2-27
Sequence Objects . . . 2-27
ANSI Join Syntax . . . 2-27
Unions in Subqueries of SELECT Statements . . . 2-27
LVARCHAR Data Types Greater Than 2048 Bytes . . . 2-27
New SQL Reserved Words . . . 2-28
New Environment Variables . . . 2-28
GLS Enhancements . . . 2-28
Support for Unicode . . . 2-28
Support for Unicode Collation . . . 2-29
Full Support for Chinese GB18030-2000 Locale . . . 2-29
Reliability, Availability, and Supportability Features . . . 2-29
Dynamically Monitor Queries . . . 2-29
Print the Session Control Block Address . . . 2-29
Display Environment Variable Settings . . . 2-29
Print Online Chunk Pages . . . 2-29
Display Stored Procedure Information . . . 2-29
DataBlade API Enhancements . . . 2-29
New mi_get_db_locale() Function . . . 2-29
New mi_get_transaction_id() Function . . . 2-29
New mi_realloc() function . . . 2-30
New mi_stack_limit() Function . . . 2-30
New mi_system() Function . . . 2-30
Enhanced Stream Support . . . 2-30
High-Performance Loader Enhancements . . . 2-30
Full Use of Storage Media . . . 2-30
New Location for the Custom-Code Shared Library File . . . 2-31
Custom-Code Function Input and Output Length . . . 2-31
Backup and Restore Enhancements . . . 2-31
Renaming Chunks During a Cold Restore . . . 2-31
Full Use of Storage Media and Increased File Size Limit . . . 2-31
Installation Enhancements . . . 2-31
No Files Installed in the /usr/lib Directory . . . 2-31
More Recent Client and GLS Files Are Not Overwritten . . . 2-31
Serial Number and Key are No Longer Needed . . . 2-31

Changed or New URLs 2-32
New Features in Version 9.3 2-32
UNIX Bundle Installer 2-33
Database Server Usability Enhancements 2-33
Ability to Display the Maximum Number of Connections 2-33
Changes to onconfig.std File 2-33
Database Server Administration Utilities (Windows) 2-33
High-Availability Data Replication Failover Scripts 2-34
DataBlade API Enhancements. 2-34
New PER_STMT_EXEC and PER_STMT_PREP Memory Durations 2-34
NULL Connections for mi_lo() Functions 2-34
New mi_collection_card() Function to Obtain Cardinality for Collections 2-34
Access to Files on a Client Computer One Buffer at a Time 2-35
New Callbacks to Handle Transactions 2-35
New Function for Determining the Transaction State for DataBlades 2-35
Enterprise Replication Enhancements 2-35
Replication of Extensible Data Types 2-36
Support Functions for Replication of User-Defined Types 2-36
Performance Enhancements to Enterprise Replication 2-36
SERIAL Column Primary Keys 2-37
Replicate Sets and Exclusive Replicate Sets 2-37
Replicating Only Changed Columns 2-37
Spooling of Replicate Data to Nonlogging Smart Large Objects 2-37
In-Place Alters to Add or Drop Shadow Columns (CRCOLS) 2-38
New onstat Options for Enterprise Replication 2-38
The cdr finderr Utility 2-38
Extensibility Enhancements 2-38
DeepCopy Function for Multirepresentational Data Types 2-38
Nearest Neighbor Queries in R-Trees 2-39
Temporary Sbspaces and Smart Large Objects 2-39
Improved Space Allocation of User Data and Metadata in Sbspaces 2-39
J/Foundation Enhancements 2-39
JVM 1.3 Support in J/Foundation 2-40
Performance Enhancements 2-40
Configurable Default Lock Modes 2-40
The onstat -g stm Option 2-40
The Ability to Display the Query Plan Without Executing the Query 2-40
Dynamic Addition of Logical Logs 2-41
SQL Enhancements 2-41
Optional FROM in the DELETE Statement 2-41
REVOKE AS User 2-41
Features from Dynamic Server, Version 9.21 2-41
ANSI Join Syntax 2-41
Rename Index Statement 2-42
Nonlogging (RAW) Tables 2-42
onpladm Utility 2-42
The onbar -b -l Command 2-42
9.x DB-Access to 7.x Synonyms 2-42
SQL Statement Cache Improvements 2-42
DataBlade API Features. 2-43
Functions for Controlling the Virtual-Processor Environment 2-43
Functions for Getting Information About a UDR 2-43
Java Features in 9.21 2-44
JVM 1.2 Support in J/Foundation 2-44
Default Values of Java Configuration Parameters 2-44
JDBC 2.0 Support 2-44
GLS Support for J/Foundation 2-44
update_jars.sql Script 2-44
Java Runtime Environment Variables 2-44
Partial Support for Variable-Length Opaque-Types 2-44
References to J/Foundation Features 2-45

MaxConnect Support 2-45

In This Chapter

This chapter describes the new features in Dynamic Server, Version 10.0, 9.4, 9.3, and 9.21.

Important: See your release notes and documentation notes for the latest information on new features.

New Features in Version 10.00 of IBM Informix Dynamic Server

New Features in Version 10.00.xC4

IBM Informix Dynamic Server, Version 10.00.xC4 contains the following new features:

- TRUNCATE table support
- Enterprise Replication direct synchronization
- Enterprise Replication consistency checking
- Enhanced support for IPv6
- Secure local connections
- Secure DataBlade module paths
- Parallel backup and restore is more efficient
- DB-Access stops a process after the first error
- Informix Interface for TSM supports HP-UX (Itanium)
- New default value for IFX_EXTEND_ROLE configuration parameter

TRUNCATE Table support

TRUNCATE is a SQL keyword that quickly deletes active rows from a table and the b-tree structures of its indexes. The active rows are deleted without dropping the table or its schema, access privileges, triggers, constraints, and other attributes. With TRUNCATE TABLE, you can de-populate a local table and release (or reuse for the same table) the storage space that formerly held its data rows and b-tree structures. For more information, see the *Guide to SQL: Syntax*.

Enterprise Replication Direct Synchronization

The **cdr sync replicate** and **cdr sync replicateset** commands perform direct synchronization between a reference server and one or more target servers. For more information, see the *Enterprise Replication Guide*.

Enterprise Replication Consistency Checking

The **cdr check replicate** and **cdr check replicateset** commands check for consistency within replicates, and optionally, repair inconsistent rows. For more information, see the *Enterprise Replication Guide*.

Enhanced Support for IPv6

IPv6 support now includes automatic discovery of whether or not a platform supports IPv6.

Secure Local Connections

The SECURITY_LOCALCONNECTION configuration parameter enables a database server administrator (DBSA) to set up security checking for local connections with the same host. For more information, see the *Administrator's Reference*.

Secure DataBlade Module Paths

The DB_LIBRARY_PATH configuration parameter allows you to specify a comma-separated list of valid directory prefix locations from which the database server can load external modules, such as DataBlade modules. You can use DB_LIBRARY_PATH to control the location from which shared objects can be loaded, and it allows you to enforce policies and standards on the formats of the EXTERNAL NAME clause of the CREATE FUNCTION, CREATE PROCEDURE, and CREATE ROUTINE statements. For more information, see the *Administrator's Reference*.

Parallel Backup and Restore is More Efficient

The new BAR_SORT_DBS environment variable can improve the efficiency of backup and restore operations whose scope is not the whole system. This environment variable implements new sorting methods for parallel dbspace backup and restore. BAR_SORT_DBS takes effect if it is set to any value. You can preserve the (less efficient) legacy behavior of backup and restore operations by not setting BAR_SORT_DBS.

DB-Access Stops a Process After the First Error

In DB-Access, using the **-a** command-line option will stop a process directly after the first error is encountered. Stopping a process from continuing after the first error ensures greater database consistency. For more information on DB-Access, see the *DB-Access User's Guide*.

Informix Interface for Tivoli Storage Manager Supports HP-UX (Itanium)

You can now use the ON-Bar utility with Tivoli Storage Manager on the HP-UX (Itanium) platform. The Informix Interface for TSM supports HP-UX.

New Default Value for IFX_EXTEND_ROLE Configuration Parameter

The default value of the IFX_EXTEND_ROLE configuration parameter is changing from 0 to 1. The new default is more restrictive so that only administrators or users who have been granted extend role by administrators can register external routines.

New Features in Version 10.00.xC3

IBM Informix Dynamic Server, Version 10.00.xC3 contains the following new features:

- ANSI-joins in distributed queries
- Transaction support for XA-compliant external data sources
- MQ DataBlade module
- New DBCREATE_PERMISSION configuration parameter to restrict the ability to create databases
- New secure default directory for the DUMPDIR configuration parameter
- Table-level restore for smart large object columns
- AES cipher support for network encryption
- New Enterprise Replication commands to show statistics information
- Client SDK included in Dynamic Server installation process
- Returning subsets of query results as collection-derived tables
- Ordering subsets of query results in collection-derived tables
- J/Foundation upgrade to JRE 1.4.2

- New default directory for ADTPATH configuration parameter
- New UNSECURE_ONSTAT configuration parameter

ANSI-Joins in Distributed Queries

Distributed queries that use ANSI-compliant JOIN syntax for specifying joined tables, execute more efficiently in Informix Dynamic Server 10.00.xC3 than in earlier releases, this is achieved by sending the remote joined tables as one query instead of multiple queries for each remote table, to each participating database server for operations on local tables of those servers. If you revert from Dynamic Server 10.00.xC3 to an earlier release that does not support this implementation of the ANSI-compliant syntax, such queries might show reduced performance, because the Dynamic Server instance from which the query originates performs the joins locally.

Transaction Support for XA-Compliant External Data Sources

The Dynamic Server Transaction Manager recognizes XA-compliant external data sources, which can participate in two-phase commit transactions. The transaction manager can invoke support routines for each XA-compliant external data source that participates in a distributed transaction at a particular transactional event, such as prepare, commit, or rollback. This interaction conforms to X/Open XA interface standards. For more information, see the *IBM Informix DataBlade API Programmer's Guide* and the *IBM Informix DataBlade API Function Reference*.

MQ DataBlade Module

The MQ DataBlade module provides mechanisms for data exchange between Dynamic Server and queues managed by IBM WebSphere MQ. For more information, see the *IBM Informix Built-In DataBlade Modules User's Guide*.

Restricting Database Creation

You can restrict which users can create Informix databases by setting the DBCREATE_PERMISSION configuration parameter in the ONCONFIG file. If you do not set DBCREATE_PERMISSION, all valid users can create databases. When you set DBCREATE_PERMISSION, only user **informix** and the users whom you specify are allowed to create databases. For more information, see the chapter on security in the *IBM Informix Dynamic Server Administrator's Guide* and the *Administrator's Reference* documentation notes.

Secure DUMPDIR Configuration Parameter Default Directory

The previous default value of the DUMPDIR configuration parameter was the **/tmp** directory. The new default value of the DUMPDIR configuration parameter in the **onconfig.std** file is the more secure **/usr/informix/tmp** on UNIX and **%INFORMIXDIR%\tmp** on Windows. If the DUMPDIR configuration parameter is not in the ONCONFIG file, then the **$INFORMIXDIR/tmp** directory is used.

Table-Level Restore Support for Smart Large Objects

For Dynamic Server version 10.00.xC1, table-level restore supported all built-in, BOOLEAN, and LVARCHAR data types for physical and logical restore. For Dynamic Server version 10.00.xC3, table-level restore also supports smart large objects for physical restore only (restore from level-0 archive). For more information, see the *Backup and Restore Guide* documentation notes.

AES Ciphers for Network Encryption

The following AES ciphers are supported for network encryption:

aes AES 128bit key

aes128 AES 128bit key

aes192 AES 192bit key

aes256 aes 256bit key

Network encryption is described in the chapter on security in the *IBM Informix Dynamic Server Administrator's Guide*.

Enterprise Replication Statistics Commands

The **cdr stats rqm** command displays information about the reliable queue manager (RQM) queues used for Enterprise Replication. The **cdr stats recv** command displays receiver parallelism statistics and latency statistics by source node. For more information, see the *Enterprise Replication Guide* documentation notes.

CSDK Included in Dynamic Server Installation

You can install IBM Informix CSDK with the Dynamic Server installation program. CSDK version 2.90.xC3 is included as part of the Dynamic Server version 10.00.xC3 installation package. You can choose whether to install CSDK simultaneously with Dynamic Server or separately. For more information, see the *IBM Informix Dynamic Server Installation Guide for UNIX and Linux*.

Enhanced Support for Retrieving Subsets of Query Results

New syntax enhancements support FIRST (and its new keyword synonym LIMIT) in the Projection clause, as well as the new SKIP keyword, which can exclude a specified number of the first qualifying rows before saving the result set as a collection-derived table. You can also use host variables or local SPL variables to specify the SKIP or FIRST parameters of queries in prepared objects. Before applying the FIRST and SKIP clause, the rows in the result set are numbered starting from 1. This feature provides greater flexibility in SELECT and INSERT statements, and in web applications whose pages display disjunct subsets of qualifying rows. For more information, see the *IBM Informix Guide to SQL: Syntax*.

Returning Subsets of Query Results as Collection-Derived Tables

Previous Dynamic Server releases did not support the ORDER BY clause in collection subqueries nor in conjunction with the FIRST keyword in the Projection clause of collection subquery. Syntax enhancements now support the ORDER BY clause in the FROM clause of a SELECT statements that defines a collection-derived tables. If the SELECT statement also includes any of the FIRST, LIMIT, or SKIP keywords in its Projection clause, the ORDER BY clause is applied to qualifying rows before the SKIP and FIRST specifications are applied to the result set. For more information, see the *IBM Informix Guide to SQL: Syntax*.

J/Foundation Upgraded to JRE 1.4.2

The J/Foundation component of Dynamic Server now includes JRE 1.4.2.

Secure ADTPATH Configuration Parameter Default Directory

The default value of the ADTPATH configuration parameter for Dynamic Server 10.00.xC3 and later on UNIX is the secure **$INFORMIXDIR/aaodir** directory, instead of the **/tmp** directory.

UNSECURE_ONSTAT Configuration Parameter

The **onstat** commands that show SQL statement text being executed by a session are normally restricted to DBA users. This restriction can be removed by setting the UNSECURE_ONSTAT configuration parameter to 1. For example, the **onstat -ses**, **onstat -stm**, **onstat -ssc**, and **onstat -sql** commands show SQL statement text.

onconfig.std value

not set

possible values

1

takes effect

when the database server is shut down and restarted

New Features in Version 10.00.xC1

IBM Informix Dynamic Server, Version 10.00.xC1 contains new features in the following areas:

- Security enhancements
- Server usability enhancements
- Performance enhancements
- SQL enhancements
- Enterprise Replication enhancements
- Backup and restore enhancements
- Storage enhancements
- Extensibility enhancements
- Installation enhancements
- Interoperability enhancements

Security Enhancements

IBM Informix Dynamic Server 10.0 provides significant advances in database server security, encryption, authentication, and availability.

Column-Level Encryption: You can use the new SQL statement, SET ENCRYPTION PASSWORD, to implement column-level encryption to improve the confidentiality of the data. New built-in functions provide methods for encrypting and decrypting data. The system catalog does not identify which columns contain encrypted data, and the same column can include both encrypted and unencrypted values. This security enhancement feature supports data confidentiality and data integrity.

This feature is primarily documented in the *IBM Informix Dynamic Server Administrator's Guide* and *IBM Informix Guide to SQL: Syntax*.

Server Utilities Check for Secure Environment Before Starting: This feature was first available with version 9.40.xC3. Server utilities on UNIX now check if the environment is secure by testing for the following conditions before starting:

- The permissions on **$INFORMIXDIR** and some directories under it are correct. For each directory, check that the directory exists, is owned by user **informix** and the correct group, and that its permissions do not include write permissions for the group or other users.
- The permissions on the ONCONFIG file are correct. The file must belong to the DBSA group. If the DBSA group is group informix (default), then the ONCONFIG file should be owned by user **informix** too; otherwise, the ownership is not constrained. The file must not have write permissions for others.
- The permissions on the **sqlhosts** file are correct. Under a default configuration, the **sqlhosts** file is **$INFORMIXDIR/etc/sqlhosts**; the owner should be user **informix**, the group should be either the **informix** group or the DBSA group, and there should be no public write permissions. If the file is specified by setting the **INFORMIXSQLHOSTS** environment variable, then the owner and group are not checked, but public write permissions are not permitted.

- The length of both the file specifications **$INFORMIXDIR/etc/onconfig.std** and **$INFORMIXDIR/etc/$ONCONFIG** must each be less than 256 characters.

If the tests for any of these conditions fail, the utilities exit with an error message.

This feature is primarily documented in the *IBM Informix Dynamic Server Administrator's Guide*.

Restricting Registration of External Routines: The DBSA (Database Server Administrator) can use a new built-in role, called EXTEND, to specify which users can register UDRs that include the EXTERNAL NAME clause. User-defined routines use shared-object files that are external to the database server and that could potentially contain harmful code. The DBSA can use the GRANT statement to confer the EXTEND role on a user (typically the DBA of a local database), or can use REVOKE to withdraw that role from a user. The DBSA can disable this feature by setting to "off" a new IFX_EXTEND_ROLE configuration parameter. This feature is intended to improve security and to control accessibility.

This feature is primarily documented in *IBM Informix Guide to SQL: Syntax*, *IBM Informix Dynamic Server Administrator's Guide*, and *IBM Informix Dynamic Server Administrator's Reference*.

Preventing Denial-of-Service Attacks: Dynamic Server provides multiple listener threads available to handle connections and imposes limits on the availability of the listener VP for incomplete connections. This feature reduces the risk of a hostile denial-of-service attack by making it more difficult to overwhelm the listener VP that handles connections. The default incomplete connection timeout period is reduced from 60 to 10 seconds. The default maximum number of incomplete connections is 1024.

You can customize this feature with the following two new configuration parameters:

- LISTEN_TIMEOUT. Sets the incomplete connection timeout period.
- MAX_INCOMPLETE_CONNECTION. Restricts the number of incomplete requests for connections.

You can change the value of these configuration parameters with the **onmode** utility while the database server is running.

This feature is primarily documented in the *IBM Informix Dynamic Server Administrator's Guide* and *IBM Informix Dynamic Server Administrator's Reference*.

Server Usability Enhancements

The server usability enhancements include features for ease of administration, scalability, and high availability.

Configuring Page Size: This feature adds support for specifying the page size for a standard or temporary dbspace when you create the dbspace. You

Informix Interface for Tivoli Storage Manager: The Informix Interface for Tivoli Storage Manager (formerly known as Tivoli Data Protection for Informix) is bundled with the Dynamic Server installation. This feature eases installation. You can use Informix Interface for TSM with ON-Bar to store data. Informix Interface for TSM stores Dynamic Server databases and logical logs on the Tivoli Storage Manager.

Extensibility Enhancements

The extensibility enhancements improve distributed transactions, obtaining information from trigger executions, and Java support.

Manipulating Built-in Opaque Data Types In External Tables: You can use built-in opaque data types in remote queries involving databases residing on the same database server. The built-in opaque data types supported by this feature are BOOLEAN, BLOB, CLOB, and LVARCHAR. User-defined types that can be cast explicitly to a built-in data type are also supported. You can use these data types in DML operations and as parameters and returned data types of UDRs between all databases of the same Dynamic Server instance. The target data type of the explicit cast must be a built-in opaque type or other built-in SQL data type, and all the casts and all the UDTs must be defined in all of the databases participating in the query. The following types of operations are supported:

- DML operations such as SELECT, UPDATE, DELETE, and INSERT on tables across databases having built-in opaque data type columns. The table can be a table, view, or synonym.
- DDL operations such as CREATE VIEW and CREATE SYNONYM at the local database on cross-database tables having built-in opaque data type columns. The view or synonym can be created only at the local database.
- Implicit and explicit execution of user-defined routines (written in SPL, C, or Java) with built-in opaque data type parameters and return types. This is applicable for both functions and procedures.

Obtaining Information From Trigger Executions: You can create user-defined routines that are invoked in trigger action statements to obtain information about the triggers, triggering tables, views, statements, and the values of rows involved in the trigger actions. Using the new DataBlade API routines, you can write a general purpose user-defined routine that can you can use to audit any table and any trigger event.

This feature is documented in the *IBM Informix DataBlade API User's Guide*.

Support for JRE Version 1.4: J/Foundation supports JRE Version 1.4 and the JDBC 3.0 specification.

Installation Enhancements

The installation enhancements improve usability. These enhancements are documented in the *IBM Informix Dynamic Server Installation Guide for UNIX and Linux*.

IBM Software Electronic Licensing: Dynamic Server, CSDK, JDBC, and ISA installation programs now include the display and required acceptance of a License Agreement. The License Agreement and License Information files are installed with the product and available for customer review at any time.

New doc Directory Before Installation: The following files are now available in a **/doc** directory that is available before installing Dynamic Server:

- Release notes

- Machine notes
- Documentation notes
- *IBM Informix Dynamic Server Installation Guide* (in PDF format)

New installation program on UNIX and Linux

You can install Dynamic Server on UNIX or Linux using the new installation application in console, GUI, or silent mode. For detailed instructions, see the *IBM Informix Dynamic Server Installation Guide for UNIX and Linux*.

Interoperability Enhancements

The interoperability enhancements improve communication between Informix and DB2 products.

Running Informix ESQL/C Applications with DB2: You can run Informix ESQL/C applications with DB2 servers and databases. The Informix ESQL/C product provides new libraries that are called when you use the esql command to preprocess your files to work with DB2. Informix ESQL/C runs with DB2 Version 8.2, or later, running on Linux, UNIX, and Windows operating systems.

New Features in Version 9.4

The new features for Dynamic Server, Version 9.4, fall into the following major areas:

- Security Enhancement
- Database Server Usability Enhancements
- Performance Enhancements
- Enterprise Replication Enhancements
- Extensibility Enhancements
- SQL Enhancements
- GLS Enhancements
- Reliability, Availability, and Supportability Features
- DataBlade API Enhancements
- High-Performance Loader Enhancements
- Backup and Restore Enhancements
- Installation Enhancements
- Changed or New URLs

The *IBM Informix Migration Guide* lists all new environment variables, configuration parameters, system-monitoring interface (SMI) tables in the **sysmaster** database, system catalog tables, and reserved SQL keywords in Version 9.4.

Security Enhancement

Version 9.4 of Dynamic Server supports encrypting data transmissions over the network using the encryption communication support module (ENCCSM).

This option provides complete data encryption with the openSSL library, with many configurable options. A message authentication code (MAC) will be transmitted as part of the encrypted data transmission to ensure data integrity. A MAC is an encrypted message digest.

The encryption algorithms use openSSL 0.9.6 as the code base.

Distributed queries can also be encrypted.

For more information on encryption, see the *IBM Informix Dynamic Server Administrator's Guide*.

Enterprise Replication implements encryption with configuration parameters instead of the ENCCSM. For more information, see "Enterprise Replication Security" on page 2-22.

Database Server Usability Enhancements

Version 9.4 of Dynamic Server supports the following usability enhancements.

Increase Size of Chunks, Chunk Offsets, and Number of Allowable Chunks

Chunks and chunk offsets now have a limit of 4 TB (2**42 bytes) in size. The previous limit was 2 GB (2**31 bytes). The number of chunks per database server is now 32,766. The previous limit was 2,047. These features are enabled by setting large chunk mode with the **onmode** utility.

For information on these new limits, see the *IBM Informix Dynamic Server Administrator's Guide*. For information on how to enable large chunk mode, see the *IBM Informix Administrator's Reference*.

Configurable Event Alarms

You can now configure event alarms with a modifiable shell script, **alarmprogram.sh**.

Set the ALARMPROGRAM configuration parameter to alarmprogram.sh, and edit the file to specify the email address of the database administrator, the email address of the pager service, the mail utility, and whether to automatically backup logical logs.

For more information on event alarm parameters, see the *IBM Informix Dynamic Server Administrator's Guide*. For more information on setting event alarms, see the *IBM Informix Dynamic Server Administrator's Reference*.

Increased Database Server Aliases

You can now specify up to 32 database server aliases with the DBSERVERALIASES configuration parameter.

For more information, see the *IBM Informix Dynamic Server Administrator's Reference*.

Increased File Size Limit

The new file size limit is 4 TB. This limit applies to all database server utilities, including the following:

- The UNLOAD and LOAD statements of SQL (see "LOAD TO and UNLOAD FROM with Large Files" on page 2-26)
- The **onspaces** utility
- The **ontape** utility (see "Full Use of Storage Media and Increased File Size Limit" on page 2-31)
- The shared-memory dump file
- The **dbimport** and **dbexport** utilities
- DataBlade API stream support functions

The previous file size limit was 2 GB. (Logical log files, however, must not exceed 1 GB in size for Version 9.4.)

For more information on **dbimport** and **dbexport**, see the *IBM Informix Migration Guide*.

Full Use of Storage Media

Utilities that use storage media backup and restore or loading and unloading data can use the full size of the storage media. This feature is support by the following utilities:

- The **ontape** utility (see "Full Use of Storage Media and Increased File Size Limit" on page 2-31)
- The **onload** and **onunload** utilities
- The **dbimport** and **dbexport** utilities
- High-Performance Loader utilities: **ipload**, **onpload**, and **onpladm**

Except for the High-Performance Loader utilities, you use this option by setting the tape size to `0`. For information about using this feature with HPL utilities, see "High-Performance Loader Enhancements" on page 2-30.

In previous releases, the user was required to specify a non-zero tapesize value when using these utilities, and risked wasting storage space. The previous limit was 2 GB per storage device.

For more information on the **onload**, **onunload**, **dbimport**, and **dbexport** utilities, see the *IBM Informix Migration Guide*.

Increased Default Values for Tape Block Size Configuration Parameters

The default tape block size for the TAPEBLK and LTAPEBLK configuration parameters in the **onconfig.std** file has been increased to 32 kilobytes in Version 9.4. Here TAPEBLK specifies the block size for tapes used in storage-space backups, and LTAPEBLK specifies the block size for tapes used in logical-log backups.

The default value for TAPEBLK and for LTAPEBLK in earlier releases was 16 kilobytes.

For more information, see the *IBM Informix Dynamic Server Administrator's Reference*.

Chunk Reserve Pages in Non-Root Chunks

Chunk reserve pages are stored in the root chunk. In previous releases of Dynamic Server, you could not add chunks if the root chunk was full. However, for Version 9.4, if you add chunks when the root chunk is full, the new chunk metadata is stored in extended chunk reserve pages allocated from non-root chunks in the root dbspace.

For more information, see the *IBM Informix Dynamic Server Administrator's Guide*.

Restartable Fast Recovery

Restartable fast recovery allows physical logging during the roll forward phase to prevent fast recovery failure. If the physical log overflows during fast recovery, the physical log is extended to a disk file, named **plog_extend.*server_number***. The

location of this file is set by the new PLOG_OVERFLOW_PATH configuration parameter. This file is removed after the first checkpoint during fast recovery.

For more information on fast recovery, see the *IBM Informix Dynamic Server Administrator's Guide*. For more information on the PLOG_OVERFLOW_PATH configuration parameter, see the *IBM Informix Dynamic Server Administrator's Reference*.

Microsoft Transaction Server/XA Support

Transaction managers coordinate distributed queries between Informix and non-Informix databases. Informix supports XA transactions in a tightly-coupled mode, which allows you to use Microsoft Transaction Server (MTS/XA) as a transaction manager. You can use MTS/XA with IBM Informix ODBC Driver.

For information on how to monitor transactions with **onstat -x**, see the *IBM Informix Administrator's Guide* and the *IBM Informix Dynamic Server Performance Guide*. For information on MTS/XA, see the MTS/XA documentation.

Performance Enhancements

The following new features are designed to improve the performance of Dynamic Server.

PDQ is Enabled for Hold Cursors

Cursors created with the WITH HOLD keywords can now be processed in parallel.

For more information on how this feature can affect performance, see the *IBM Informix Dynamic Server Performance Guide*. For more information on the syntax of this feature, see the section on DECLARE in the *IBM Informix Guide to SQL: Syntax*.

Improved Transaction Processing with the B-tree Scanner

The new B-tree scanner improves transaction processing for logged databases when rows are deleted from a table with indexes. The B-tree scanner threads remove deleted index entries and rebalance the index nodes. The B-tree scanner automatically determines which index items are to be deleted, based on a priority list.

For more information on how this feature can affect performance, see the *IBM Informix Dynamic Server Performance Guide*. For information on how to configure the B-tree scanner with the **onstat -C** command, see the *IBM Informix Dynamic Server Administrator's Reference*.

Improved Priority Management for the Buffer Manager

Buffers are now divided into two classes: HIGH priority for frequently accessed buffers, and LOW priority for infrequently accessed buffers. Priority classification is dynamic, based on observed access frequencies of the buffers. The CPU usage of the buffer manager is reduced, thus improving performance.

For more information, see the *IBM Informix Dynamic Server Performance Guide*.

Spatial Query Costing

You can provide cost and selectivity functions for R-tree indexes to allow the optimizer to accurately choose the appropriate index to use for a particular query.

For more information, see the *IBM Informix R-Tree Index User's Guide*.

More Precise LRU Maximum and Minimum Settings

The LRU_MAX_DIRTY and LRU_MIN_DIRTY configuration parameters can take a FLOAT type value, thereby increasing the precision of buffer clean to two positions to the right of the decimal point.

For more information on how these configuration parameters affect performance, see the *IBM Informix Dynamic Server Performance Guide*. For more information on setting these configuration parameters, see the *IBM Informix Dynamic Server Administrator's Reference*.

Enterprise Replication Enhancements

The following new features enhance the extensibility, usability, or performance of the Enterprise Replication facility of Dynamic Server.

All Enterprise Replication features are documented in the *IBM Informix Dynamic Server Enterprise Replication Guide*.

Enterprise Replication Security

Enterprise Replication supports the same levels of network encryption available for client/server communications. Encryption is implemented in Enterprise Replication with the encryption configuration parameters listed in "New and Altered Configuration Parameters" on page 2-23.

Support for ROW and Collection Data types

Enterprise Replication can now replicate the following data types:

- Named and unnamed ROW data types
- Collection data types: LIST, MULTISET, and SET

Faster Queue Recovery

The addition of a table with replicate information to the transaction records and row data tables reduces transaction processing time.

Replication During Queue Recovery

Users can connect to a database server during queue recovery; transactions are added to the queue. However, if the volume of transactions during queue recovery is so large that the logical log is in danger of being overwritten, replication is blocked.

Large Transactions Support

Enterprise Replication automatically spools large transactions to disk instead of holding them in memory. Rows from spooled transactions are paged in and out of memory as needed. Enterprise Replication can replicate transactions up to 4 TB in size.

Improved Availability with HDR

You can use High-Availability Data Replication (HDR) on critical database servers in an Enterprise Replication system to provide identical backup database servers. (Dynamic Server releases earlier than Version 9.4 could support either Enterprise Replication or HDR, but both could not run concurrently.)

Dynamic Log File

Enterprise Replication can request the database server to add a new dynamic log file if replication enters DDRBLOCK mode.

The new CDR_MAX_DYNAMIC_LOGS configuration parameter specifies the number of dynamic log file requests that Enterprise Replication can make in one server session.

New Commands

The new **brief** option for the **cdr list replicate** command displays a summary of participants for all replicates

The new **cdr remove** command removes Enterprise Replication from an HDR server.

New and Altered Configuration Parameters

Enterprise Replication has the following new configuration parameters:

- CDR_DBSPACE specifies the dbspace of the **syscdr** table.
- CDR_ENV sets Enterprise Replication environment variables.
- CDR_MAX_DYNAMIC_LOGS specifies the number of dynamic log file requests that Enterprise Replication can make in one server session.
- ENCRYPT_CDR to enable and set the level of network encryption.
- ENCRYPT_CIPHERS to specify the ciphers to use for encryption.
- ENCRYPT_MAC to specify the level of message authentication coding to use.
- ENCRYPT_MACFILE to specify MAC key files.
- ENCRYPT_SWITCH to define the frequency at which ciphers and secret keys are re-negotiated.

The CDR_QDATA_SBSPACE configuration parameter now allows you to specify up to 32 sbspaces for Enterprise Replication to use for storing spooled row data.

The CDR_QDATA_SBFLAGS configuration parameter is deprecated.

New Environment Variables

The **CDR_LOGDELTA** environment variable determines when the send and receive queues are spooled to disk.

The **CDR_PERFLOG** environment variable enables queue tracing.

The **CDR_ROUTER** environment variable disables intermediate acknowledgements of transactions in hierarchical topologies.

The **CDR_RMSCALEFACT** environment variable sets the number of DataSync threads started for each CPU VP.

Extensibility Enhancements

The following new features are designed to improve the extensibility of Dynamic Server.

Enhanced HDR Support for Extensibility Features

High-Availability Data Replication (HDR) now supports replication of the following extended objects:

- All built-in and extended data types.
- User-defined routines.
- R-tree and functional indexes
- TimeSeries DataBlade module

User-defined data types (UDTs) must be logged and must reside in a single database server. Data types with out-of-row data are replicated if the data are stored in an sbspace or in a different table on the same database server.

HDR does not replicate data stored in operating system files nor in persistent (that is, non-temporary) external files. HDR also does not replicate memory objects that are associated with user-defined routines.

To use user-defined data types, user-defined routines, or DataBlade modules with HDR, you must install the user-defined data types, user-defined routines, or DataBlade modules on both the HDR primary and secondary database servers. Register the user-defined data types, user-defined routines, or DataBlade modules only on the HDR primary database server.

For more information see the *IBM Informix Dynamic Server Administrator's Guide*.

Using an Iterator Function in the FROM Clause of a SELECT Statement

An iterator function can now be specified in the FROM clause of the SELECT statement. (An iterator function is a user-defined function that returns to its calling context more than once, each time returning a value.)

You can query the returned result set of an iterator UDR using a virtual table-interface. You can then manipulate the iterator result set in a number of ways, such as by using the WHERE clause to filter the result set; by joining the UDR result set with other table scans; by running GROUP BY, aggregation, and ORDER BY operations; and so on.

For information on writing iterators, see the *IBM Informix User-Defined Routines and Data Types Developer's Guide*. For information on using iterators in the FROM clause of a SELECT statement syntax, see the *IBM Informix Guide to SQL: Syntax*.

Enhanced CREATE FUNCTION and CREATE PROCEDURE Syntax

Several new features improve the functionality of user-defined functions.

Multiple SLVs in the WHERE Clause of SELECT, UPDATE, and INSERT Statements: Because a user-defined function now can return more than one OUT parameter, DML (data manipulation language) statements that use the returned values from function calls as statement-local variables (SLVs) within queries or subqueries can now support multiple SLVs.

For more information on OUT parameters, see "Multiple OUT Parameters" on page 2-27.

For more information on SLVs, see the *IBM Informix Guide to SQL: Syntax*.

Declaring Names for Returned Values of an SPL UDR: Releases of Dynamic Server earlier than Version 9.4 support user-defined functions written in the SPL language that return one or more values of specified data types. In this release, the RETURNS (or RETURNING clause) of an SPL function can also declare a name for each returned value. This feature can make it easier for SPL functions to pass column headings to SELECT statements.

For more information, see the *IBM Informix User-Defined Routines and Data Types Developer's Guide*.

SQL Enhancements

Besides the enhancements that are described in section "Extensibility Enhancements" on page 2-23, the following additional changes to the IBM Informix dialect of the Structured Query Language (SQL) have been implemented in Version 9.4 of Dynamic Server.

INSTEAD OF Triggers on Views

The CREATE TRIGGER statement has been enhanced to support INSTEAD OF triggers on views. You can define an INSERT, UPDATE, or DELETE event on a specified view that activates the trigger. Rather than directly performing the triggering DML event, the database server executes the Action clause of the INSTEAD OF trigger. This feature provides a mechanism for updating the underlying tables of views that include columns from more than one table; such views were not updatable in earlier releases of Dynamic Server.

For more information, see the *IBM Informix Guide to SQL: Syntax*.

Enhanced SELECT Statement Syntax

The syntax rules for the SELECT statement have been enhanced.

For more information on these features, see the *IBM Informix Guide to SQL: Syntax*.

Ordering by Columns or Expressions Not in Projection List: The ORDER BY clause now can include column names or expressions that do not appear in the select list of the projection clause. The following query, for example, is now valid:

```
SELECT stock_num, manu_code FROM stock ORDER BY unit_price
```

Earlier releases had required that **unit_price** also appear in the Projection clause.

Iterator UDRs in the FROM Clause: As noted in section "Using an Iterator Function in the FROM Clause of a SELECT Statement" on page 2-24, iterator functions are now valid in the FROM clause of the SELECT statement.

Functional Indexes on More Than 16 Columns

Functional indexes are UDRs that accept column names as arguments, and whose return values are specified as index keys in the CREATE INDEX statement. In previous Dynamic Server releases, the number of columns was restricted to no more than 16.

In Version 9.4, however, the number of columns that can be arguments to a functional index is language-dependent. For UDRs written in the C language, a functional index can have up to 102 key parts. A functional index defined in the SPL or Java languages can have up to 341 key parts.

For more information, see the *IBM Informix Guide to SQL: Reference*.

Enhanced Dynamic Query Support

The DESCRIBE statement now recognizes the OUTPUT keyword. The new dynamic SQL statement, DESCRIBE INPUT, can provide information about the retrieved columns and dynamic parameters of prepared DML statements.

For more information on these features, see the *IBM Informix Guide to SQL: Syntax*.

The DESCRIBE INPUT Statement: The DESCRIBE statement in previous releases of Dynamic Server could not provide information about input parameters of the

WHERE clause of prepared INSERT or SELECT statements. It could provide limited support for UPDATE parameters if the **IFX_UPDDESC** environment variable were set. In this release, you can specify the INPUT keyword in the DESCRIBE statement to return information about each input parameter of a prepared DML statement, including the data type, identifier, and length (in bytes).

The DESCRIBE OUTPUT Statement: The client system that executed a dynamic **SQL** application can use the DESCRIBE OUTPUT statement (or simply DESCRIBE, because the OUTPUT keyword is optional) to obtain information about the output parameters of a prepared DML statement. (This is a CSDK feature, but it requires information that the database server did not make available to the client application in releases earlier than Version 9.4.)

Session-Level Non-Default Collation

In previous Dynamic Server releases, the database server sorted NCHAR and NVARCHAR values according to the localized collating sequence of the locale that the **DB_LOCALE** environment variable specified, if that locale defined a COLLATION; otherwise, all sorting operations followed the code set order.

In this release, the new SET COLLATION statement can specify the localized collation of another locale. For the rest of the session (or until the next SET COLLATION statement in the same session), sorting of NCHAR and NVARCHAR values ignores the **DB_LOCALE** setting. You can restore the default collating order by issuing the SET NO COLLATION statement. This feature enables the database server to use different localized collating orders on NCHAR and NVARCHAR data sets within a single database, if both collating orders can operate on the same character set.

Database objects (such as indexes, check constraints, and triggers) that perform collation use the collating order that was in effect when the object was created, rather than the order that is in effect at runtime, if these two collating orders are not the same.

For more information on the SET COLLATION statement, see the *IBM Informix Guide to SQL: Syntax*. For more information on the **DB_LOCALE** environment variable, see the *IBM Informix GLS User's Guide*. For more information on the NCHAR and NVARCHAR data types, see the *IBM Informix Guide to SQL: Reference*.

LOAD TO and UNLOAD FROM with Large Files

The LOAD and UNLOAD statements were previously restricted on most platforms to files no larger than 2 GB for LOAD and UNLOAD flat-file I/O operations. This restriction has been relaxed to 4 TB in Version 9.4.

For more information, see the *IBM Informix Guide to SQL: Syntax*.

SET Residency Statements No Longer Needed

In Dynamic Server releases earlier than Version 9.4, the SET TABLE and SET INDEX statements could specify whether one or more fragments of a table or of an index remain in a shared memory buffer, rather than be written to disk. These statements are no longer supported, because this functionality is now provided automatically by the database server. No error is issued, however, when applications include a SET Residency statement; the SET TABLE or SET INDEX statement is simply ignored.

For more information, see the *IBM Informix Guide to SQL: Syntax*.

Multiple OUT Parameters

In a user-defined routine (UDR), an OUT parameter corresponds to a value returned through a pointer. Earlier releases of Dynamic Server supported no more than one OUT parameter in UDRs, and any OUT parameter was required to appear as the last item in the parameter list. Version 9.4 drops these restrictions, supporting multiple OUT parameters anywhere in the parameter list of the UDR. This feature provides greater flexibility in defining UDRs, and removes the need to return collection variables in contexts where multiple returned values are required. JDBC client applications can use this feature to create multiple statement-local variables (SVLs) in the WHERE clause of a DML statement that invokes the UDR.

For more information on how to use OUT parameters in UDRs, see the *IBM Informix User-Defined Routines and Data Types Developer's Guide*. For more information on OUT parameter syntax, see the *IBM Informix Guide to SQL: Syntax*.

Sequence Objects

This release introduces new DML statements (CREATE SEQUENCE, ALTER SEQUENCE, RENAME SEQUENCE, DROP SEQUENCE) for sequence generators, database objects that multiple users can access concurrently to generate unique integers in the INT8 range.

The GRANT and REVOKE statements have been enhanced to support access privileges on sequence objects, and the CREATE SYNONYM and DROP SYNONYM statements can now reference synonyms for sequence objects in the local database. Two new operators, CURRVAL and NEXTVAL, can read or increment the value of an existing synonym. The system catalog includes a new **syssequences** table for information about sequence objects. Sequences are an efficient way to generate primary key values.

For more information on sequence object syntax, see the *IBM Informix Guide to SQL: Syntax*.

ANSI Join Syntax

The syntax of the SELECT statement has been enhanced to support the ANSI/ISO syntax for cross joins, right outer joins, and full outer joins. The keywords CROSS, RIGHT, and FULL are now supported in the context of queries that join two or more tables. This feature provides greater compliance with the ANSI standard for SQL.

For more information, see the *IBM Informix Guide to SQL: Syntax*.

Unions in Subqueries of SELECT Statements

The UNION operator is allowed in subqueries of SELECT statements. The elements of a union are SELECT statements that can recursively contain other unions.

For more information, see the *IBM Informix Guide to SQL: Syntax*.

LVARCHAR Data Types Greater Than 2048 Bytes

In previous releases, database columns of the LVARCHAR built-in opaque data type had an upper limit of 2048 bytes. Version 9.4 supports a *size* parameter in the declarations of LVARCHAR columns (or LVARCHAR variables of SPL), where *size* can be up to 32,739 bytes.

For backward compatibility, LVARCHAR objects declared with no *size* parameter can store up to 2048 bytes. This feature increases the storage capacity of the varying-length data types of Dynamic Server.

For more information, see the *IBM Informix Guide to SQL: Reference*.

New SQL Reserved Words

IBM Informix Dynamic Server, Version 9.4, recognizes new SQL keywords that might affect migration of your applications. Although you can use almost any word as an SQL identifier, syntactic ambiguities can occur in contexts where the keyword is also valid. An ambiguous statement might not produce the desired results.

For information about workarounds for such ambiguities, see the *IBM Informix Guide to SQL: Syntax*.

The following SQL keywords are new in Dynamic Server, Version 9.4:

COLLATION	FULL	RESTART
CROSS	INSTEAD	RIGHT

If you are migrating from a Dynamic Server release earlier than Version 9.30, see the release notes to Version 9.30 for words that have been added to the list of SQL keywords since Version 9.21.

For a complete list of SQL keywords, see Appendix A of the *IBM Informix Guide to SQL: Syntax*, Version 9.4.

New Environment Variables

The new **USETABLENAME** environment variable can invalidate the use of synonyms in ALTER TABLE and DROP TABLE statements of SQL.

For more information about the **USETABLENAME** environment variable, see the *IBM Informix Guide to SQL: Reference*.

The section "Enterprise Replication Enhancements" on page 2-22 describes additional new environment variables that can affect Enterprise Replication.

GLS Enhancements

Dynamic Server Version 9.4 uses Version 4.0 of the GLS library, which supports important new features for databases that do not use the default locale.

For information on the new collation order feature, see "Session-Level Non-Default Collation" on page 2-26.

All GLS features are documented in the *IBM Informix GLS User's Guide*.

Support for Unicode

The GLS library now supports the International Components for Unicode (ICU) code points for multilingual data, based on the ICU open source implementation of Unicode. By internally mapping the code set from ICU, rather than loading it from external locale files, this feature enables you to store, retrieve, and display strings from multiple languages within the same database.

Support for Unicode Collation

The GLS library now supports the Unicode Collation Algorithm that was developed by the Unicode consortium for comparing two Unicode strings. This de facto standard for multinational applications incorporates ICU technology.

Full Support for Chinese GB18030-2000 Locale

The previous release of the GLS library (Version 3.13.xC4) supported the code points within the Basic Multilingual Plane (BMP) of Unicode (code points 0x00 through 0xFFFF). The new version now supports all GB18030-2000 code points, using ICU.

Reliability, Availability, and Supportability Features

The following additional new features are designed to improve the reliability, availability, and supportability of Dynamic Server.

For more information on these features, see the *IBM Informix Dynamic Server Administrator's Reference*.

Dynamically Monitor Queries

Ability to monitor queries dynamically using the **onmode -Y** command.

Print the Session Control Block Address

Print the session control block address with the **onstat -g ses** command.

Display Environment Variable Settings

Display the current setting and values of environment variables with the **onstat -g env** command.

Print Online Chunk Pages

Ability to specify the number of pages to print, whether to print just the page headers, and to print pages from chunks that are online with the **oncheck** utility.

Display Stored Procedure Information

Display the types and values of host variables in SQL statements, show the stored procedure stack, and show the current SQL statement in a stored procedure using the **onstat -g sql** command.

DataBlade API Enhancements

The following enhancements have been made to functions that are valid within DataBlade API modules.

New mi_get_db_locale() Function

Use the **mi_get_db_locale()** function to return the value of the current database server locale.

Task	Manual
Return the value of the current database server locale.	*IBM Informix DataBlade API Programmer's Guide*
Use the **mi_get_db_locale()** function.	*IBM Informix DataBlade API Function Reference*

New mi_get_transaction_id() Function

Use the **mi_get_transaction_id()** function to return the ID of the current transaction.

Task	Manual
Return the ID of the current transaction.	*IBM Informix DataBlade API Programmer's Guide*
Use the **mi_get_transaction_id()** function.	*IBM Informix DataBlade API Function Reference*

New mi_realloc() function

Use the **mi_realloc()** function to change the size of an existing memory block.

Task	Manual
Change the size of an existing memory block.	*IBM Informix DataBlade API Programmer's Guide*
Use the **mi_realloc()** function.	*IBM Informix DataBlade API Function Reference*

New mi_stack_limit() Function

Use the **mi_stack_limit()** function to determine whether the current user stack has the specified amount of free space.

Task	Manual
Determine whether the current user stack has the specified amount of free space.	*IBM Informix DataBlade API Programmer's Guide*
Use the **mi_stack_limit()** function.	*IBM Informix DataBlade API Function Reference*

New mi_system() Function

Use the **mi_system()** function to execute operating system commands in a separate thread.

Task	Manual
Use the **mi_system()** function.	*IBM Informix DataBlade API Function Reference*

Enhanced Stream Support

Version 9.4 provides stream support for greater than 2 GB files.

High-Performance Loader Enhancements

The following enhancements have been made to the High-Performance Loader (HPL).

All new features for HPL are documented in the *IBM Informix High-Performance Loader User's Guide*.

Full Use of Storage Media

HPL utilities can use the full size of the storage media:

- For **ipload**, check the **Write/read to/from tape until end of device** checkbox on the **Load Select Job** or **Unload Select Job** windows.
- For **onpload** or **onpladm**, specify the **-Z** option with the **onpload** or **onpladm run job** commands.

New Location for the Custom-Code Shared Library File

Previously, the custom-code shared library file was installed in the **/usr/lib** directory. Now it is installed in the **$INFORMIXDIR/lib** directory. You can set the location of this file with the new HPL_DYNAMIC_LIB_PATH configuration parameter.

Custom-Code Function Input and Output Length

You can now use a different length for data in the input and output arguments of custom-code functions by setting the HPLAPIVERSION configuration parameter.

Backup and Restore Enhancements

The following enhancements have been made to the ON-Bar and **ontape** utilities for Dynamic Server, Version 9.4.

All new backup and restore features are documented in the *IBM Informix Backup and Restore Guide*.

Renaming Chunks During a Cold Restore

You can rename chunks by specifying new chunks paths and offsets during a cold restore with ON-Bar and **ontape**. This option is useful if you need to restore storage spaces to a different disk from the one on which the backup was made.

Full Use of Storage Media and Increased File Size Limit

The ontape utility can now use the full size of the storage media if the specified tape size is 0. The **ontape** utility can now backup and restore file sizes up to 4 Terabytes.

Installation Enhancements

The following enhancements have been made to the files used in installation and the installation process.

No Files Installed in the /usr/lib Directory

Files that were previously installed in the **/usr/lib** directory on UNIX are now installed in **$INFORMIXDIR/lib**. Specifically, the HPL custom-code shared library file and the optical shared library file are no longer installed in **/usr/lib** (see *IBM Informix High-Performance Loader User's Guide* and the *IBM Informix Optical Subsystem Guide*). In addition, SmartDisk is no longer supported.

For more information, see the *IBM Informix Dynamic Server Installation Guide for UNIX and Linux*.

More Recent Client and GLS Files Are Not Overwritten

The installation program on UNIX prompts the user to avoid overwriting existing client or GLS files that are more recent than those included with the database server.

For more information, see the *IBM Informix Dynamic Server Installation Guide for UNIX and Linux*.

Serial Number and Key are No Longer Needed

The installation program no longer prompts for a serial number and key.

This change is reflected in the *IBM Informix Dynamic Server Installation Guide for UNIX and Linux* and the *IBM Informix Dynamic Server Installation Guide for Microsoft Windows*.

Changed or New URLs

The URLs for IBM Informix products have changed. The URLs that are listed in new IBM Informix Dynamic Server manuals have been updated for Version 9.4. You will be able to find the same or updated information as before at the following Web sites:

The home page for the IBM Informix product family:

http://www.ibm.com/software/data/informix

The IBM Informix Online Documentation site (the former Answers Online):

http://www.ibm.com/software/data/informix/pubs/library/

The former Informix Developer Network Web site, now the IBM Informix Developer Zone:

http://www.ibm.com/software/data/developer/informix

The former Informix TechInfo Center, now IBM Software Online Support:

http://www.ibm.com/software/data/informix/support

The IBM Informix Developer Zone Systems Management Corner:

http://www7b.software.ibm.com/dmdd/zones/informix/corner_sm.html

Porting and Certification for IBM Informix database servers, including product compatibility:

http://www-306.ibm.com/software/data/informix/pubs/roadmaps.html

For information about the IBM Informix DataBlade modules:

http://www-3.ibm.com/software/data/informix/blades/

In addition, to report problems or to provide comments about IBM Informix user documentation, you can contact IBM Informix Information Development group by electronic mail at:

docinf@us.ibm.com

New Features in Version 9.3

The new features for Dynamic Server, Version 9.3, fall into the following major areas:

- Database server usability enhancements
- DataBlade API enhancements
- Enterprise Replication enhancements
- Extensibility enhancements
- Java enhancements
- Performance enhancements
- SQL enhancements

UNIX Bundle Installer

Use the IBM Informix UNIX Bundle Installer to install your IBM Informix products on UNIX or Linux and configure a demo database server that you can customize. The installer filename is **ids_install**.

For the installation instructions, see the *IBM Informix Dynamic Server Installation Guide for UNIX and Linux*.

Database Server Usability Enhancements

This release includes new features that make the database server easier to install, use, and manage.

Ability to Display the Maximum Number of Connections

When the database server starts, it checks the number of connections that the license allows and writes a message to the message log.

For information on how to display the maximum number of connections, see the chapter on initializing the database server in the *IBM Informix Dynamic Server Administrator's Guide*. For the messages, see the *IBM Informix Administrator's Reference*.

Changes to onconfig.std File

Use the VPCLASS parameter options for VP type, number, affinity, and noage, to configure virtual processor classes. The VPCLASS configuration parameter has replaced the following ONCONFIG configuration parameters:

- AFF_NPROCS
- AFF_SPROC
- NOAGE
- NUMAIOVPS
- NUMCPUVPS

The following configuration parameters are obsolete:

- LBU_PRESERVE
- LOGSMAX

For more information on configuration parameters and **onconfig.std**, see the *IBM Informix Administrator's Reference*.

Database Server Administration Utilities (Windows)

The following IBM Informix utilities simplify administration of the database server on Windows:

- The **ixpasswd.exe** utility changes the logon password for all services that log on as user **informix**.
- The **ixsu.exe** utility launches a command-line window that runs as the specified user.
- The **ntchname.exe** utility changes the registry entries for Dynamic Server from the old hostname to the new hostname.

For more information on these utilities, see the *IBM Informix Administrator's Guide*.

High-Availability Data Replication Failover Scripts

Use the **hdrmkpri.sh** and **hdrmksec.sh** scripts to switch the roles of database servers in a High-Availability Data Replication (HDR) pair. For more information, see the *IBM Informix Administrator's Guide*.

DataBlade API Enhancements

This release includes the following improvements in the DataBlade API.

New PER_STMT_EXEC and PER_STMT_PREP Memory Durations

When a user-defined routine (UDR) calls a memory-allocation function, the memory exists until the duration assigned to that memory expires. The PER_STMT_PREP memory duration lasts for the life of a prepared statement. The PER_STMT_EXEC memory duration lasts for the duration of the SQL statement.

Use the PER_STMT_EXEC and PER_STMT_PREP memory durations instead of the PER_STATEMENT memory duration.

Task	Manual
Use the new memory durations.	*IBM Informix DataBlade API Programmer's Guide*
Use memory durations in functions that have a duration argument.	*IBM Informix DataBlade API Function Reference*
Use **mi_dalloc()** to specify a memory duration.	*IBM Informix Virtual-Table Interface Programmer's Guide*
Allocate user-data memory with PER_STMT_EXEC memory duration.	*IBM Informix Virtual-Table Interface Programmer's Guide*
Display information on the **PRP.*sessionid.threadid*** and **EXE.*sessionid.threadid*** pools.	*IBM Informix Administrator's Reference* (see **onstat -g mem**)

NULL Connections for mi_lo() Functions

The DataBlade API provides a set of **mi_lo*()** functions for handling smart large objects. This feature permits a NULL connection using the same error-handling behavior as with a valid connection. To use **mi_lo*()** functions without a connection, specify the NULL argument.

Task	Manual
Pass a NULL connection to an **mi_lo*()** function.	*IBM Informix DataBlade API Programmer's Guide*
Specify a NULL connection in a call to an **mi_lo*()** routine that takes a connection descriptor.	*IBM Informix DataBlade API Function Reference*

New mi_collection_card() Function to Obtain Cardinality for Collections

Use the **mi_collection_card()** function in a UDR to return the cardinality of a collection (the number of items in a collection such as LIST, SET, and MULTISET).

Task	Manual
Determine the cardinality of a collection.	*IBM Informix DataBlade API Programmer's Guide*

Task	Manual
Use the **mi_collection_card()** function.	*IBM Informix DataBlade API Function Reference*
Use LIST, MULTISET, and SET data types.	*IBM Informix Guide to SQL: Reference*

Access to Files on a Client Computer One Buffer at a Time

The DataBlade API provides a set of **mi_file*()** functions for performing I/O operations on files. Previously, the **mi_file*()** functions transferred the entire file to the client computer but now these functions can transfer the file one buffer at a time.

Task	Manual
Access client files one buffer at a time.	*IBM Informix DataBlade API Programmer's Guide*
Open a file on the client by passing the MI_O_CLIENT_FILE flag to **mi_file_open()**.	*IBM Informix DataBlade API Function Reference*

New Callbacks to Handle Transactions

The database server invokes three new callbacks for transactions:

- The database server invokes a save point callback (MI_EVENT_SAVEPOINT) before committing or rolling back a save point in a transaction.
- The database server calls MI_EVENT_COMMIT_ABORT before committing or rolling back a transaction
- The database server calls MI_EVENT_POST_XACT after committing or rolling back a transaction.

For details, see the *IBM Informix DataBlade API Programmer's Guide*.

New Function for Determining the Transaction State for DataBlades

The **mi_transaction_state()** function returns the current transaction state for a DataBlade module to the caller. The transaction states are `none`, `implicit`, or `explicit`.

Task	Manual
Determine the state of a transaction.	*IBM Informix DataBlade API Programmer's Guide*
Use the **mi_transaction_state()** function.	*IBM Informix DataBlade API Function Reference*

Enterprise Replication Enhancements

Enterprise Replication conversion and reversion is now manual instead of automatic. For instructions, see the *IBM Informix Migration Guide*. For the error messages, see the *IBM Informix Administrator's Reference*.

Dynamic Server, Version 9.3, includes extensibility enhancements, performance improvements, functionality enhancements, and command-line changes for Enterprise Replication.

Replication of Extensible Data Types

Enterprise Replication provides support for replicating the following extensible data types:

- Data stored as smart large objects in sbspaces (CLOB and BLOB data types), CLOB and BLOB columns (explicitly specified in the table schema), and updates to CLOB and BLOB columns (with some restrictions)
- Opaque user-defined types (UDTs)
- Multirepresentational data types, if the required stream support functions exist.

  For information on writing the required functions, refer to the *IBM Informix DataBlade API Function Reference*.
- IBM Informix Spatial DataBlade module

Version 9.3 does not include support for replication of the following user-defined types:

- Row types
- Collections
- Lists
- Sets and Multisets

Enterprise Replication allows the following (with some restrictions):

- UDT column references and UDRs in replicate WHERE clauses
- UDTs for primary key columns

For more information, see *IBM Informix Dynamic Server Enterprise Replication Guide*:

- Replicating simple and smart large objects
- Considerations for replicating opaque data types
- UDT support functions

Support Functions for Replication of User-Defined Types

To replicate UDTs, Enterprise Replication requires that the UDT designer provide two support functions: **streamwrite()** and **streamread()**. The **streamwrite()** function converts the UDT column data from the in-server representation to a representation that can be shipped over the network. On the target server, Enterprise Replication calls the **streamread()** function for each UDT column that it transmitted using the **streamwrite()** function.

For more information, see the section on writing opaque-type support functions in the *IBM Informix DataBlade API Programmer's Guide*.

Performance Enhancements to Enterprise Replication

Enterprise Replication includes the following performance improvements to parallel processing:

- Enterprise Replication now applies all replicates (in replicate sets and individually) in parallel by default.
- Enterprise Replication threads now apply transactions from the same source in parallel unless they contain updates to the same row.
- Enterprise Replication threads normally commit on the target in the same order as on the source.
- Enterprise Replication threads can commit out of order on the target if there are no conflicts.
- Enterprise Replication now uses buffered logging to apply transactions.

Improved parallel processing is built in and requires no user configuration or interaction. However, this feature is automatically disabled if you are using page-level locking.

SERIAL Column Primary Keys

The CDR_SERIAL configuration parameter enables control over generating values for serial and SERIAL8 columns in tables defined for replication. This feature is useful for generating serial column primary keys in an Enterprise Replication environment.

For more information, see CDR_SERIAL in the section on configuration parameters in the *IBM Informix Dynamic Server Enterprise Replication Guide*.

Replicate Sets and Exclusive Replicate Sets

You can manage replicates individually and as part of a *replicate set*. Put tables in *exclusive replicate sets* to guarantee referential integrity between tables when you use any form of time-based replication.

Warning: Replicate groups are not supported in Version 9.3. Before you migrate to Version 9.3, you must remove any replicate groups.

For more information, see creating and managing replicate sets in the *IBM Informix Dynamic Server Enterprise Replication Guide* and migrating Enterprise Replication data in the *IBM Informix Migration Guide*.

Replicating Only Changed Columns

Enterprise Replication provides the ability to replicate only the changed columns, rather than the entire row.

If only changed columns are replicated, the data for all replicated columns might not be available for spooling to the ATS (Aborted Transaction Spooling) and RIS (Row Information Spooling) files. Therefore, the format for these files has changed.

For more information, see the *IBM Informix Dynamic Server Enterprise Replication Guide* regarding:

- Replicating only changed columns
- Aborted Transaction Spooling files
- Row Information Spooling files

Spooling of Replicate Data to Nonlogging Smart Large Objects

Enterprise Replication spools row data in the send and receive queues to an sbspace that you specify in the CDR_QDATA_SBSPACE configuration parameter. You can control logging of these sbspaces.

Enterprise Replication spools transaction records from the send and receive queues to a dbspace that you specify in the cdr_qhdr_dbspace parameter.

For more information, see the *IBM Informix Dynamic Server Enterprise Replication Guide* regarding:

- Setting up send and receive queue spool areas
- Defining replication servers
- Specifying CDR_QDATA_SBSPACE and CDR_QHDR_DBSPACE configuration parameters

In-Place Alters to Add or Drop Shadow Columns (CRCOLS)

Enterprise Replication uses shadow columns for conflict resolution. The database server now processes the following ALTER statements for adding and dropping shadow columns as in-place alters in most cases:

```
ALTER TABLE ... ADD CRCOLS
ALTER TABLE ... DROP CRCOLS
```

In-place alters are quick because the database server updates each row in place instead of copying the entire table. The in-place processing of these ALTER statements requires no user action.

Task	Manual
Prepare tables for conflict resolution.	*IBM Informix Dynamic Server Enterprise Replication Guide*
Add or drop shadow columns: • ALTER TABLE . . . ADD CRCOLS • ALTER TABLE . . . DROP CRCOLS	*IBM Informix Guide to SQL: Syntax*
Understand the performance advantages of in-place alters and when they occur.	*IBM Informix Performance Guide*

New onstat Options for Enterprise Replication

Use the following **onstat** options to obtain information about replication of user-defined routines (UDRs):

- **onstat -g dss UDR**
- **onstat -g dss UDRx**
- **onstat -g grp UDR**
- **onstat -g grp UDRx**

For details, see the appendix on **onstat** commands in the *IBM Informix Dynamic Server Enterprise Replication Guide*.

The cdr finderr Utility

This release includes updates to the command-line interface to support new features, including a new **cdr finderr** utility which looks up a specific Enterprise Replication error number and displays the corresponding error text.

For more information, see the command-line utility reference in the *IBM Informix Dynamic Server Enterprise Replication Guide*.

Extensibility Enhancements

This release includes the following improvements in the area of extensibility.

DeepCopy Function for Multirepresentational Data Types

Use the **DeepCopy** function for user-defined types with multiple representations, such as images. The **DeepCopy** function copies the user-defined type so that the user can safely allocate both the in-row value and out-of-row data with the default memory duration.

After you register the **DeepCopy** function for the multirepresentational types, the database server automatically invokes **DeepCopy**.

Task	Manual
Use multirepresentational data types.	*IBM Informix User-Defined Routines and Data Types Developer's Guide*
Use the **DeepCopy** function in a UDR.	*IBM Informix DataBlade API Programmer's Guide* *IBM Informix User-Defined Routines and Data Types Developer's Guide*

Nearest Neighbor Queries in R-Trees

R-tree indexes support nearest-neighbor queries. A *nearest-neighbor query* asks for items in a spatial database that are the nearest to a specific location or object. If you do a nearest-neighbor query on a map of the San Jose area, Santa Clara would be the nearest neighbor, but not San Francisco. Version 9.3 supports *composite R-tree indexes*.

For more information, see the *IBM Informix R-Tree Index User's Guide*.

Temporary Sbspaces and Smart Large Objects

Smart-large-object performance is significantly faster for certain operations. Writes to temporary smart large objects are faster than for standard smart large objects.

Use *temporary smart large objects* to store text, image, or user-defined data that you need temporarily for a user session. You can store temporary smart large objects in a standard sbspace or *temporary sbspace*. If temporary smart large objects are stored in a temporary sbspace, the metadata and user data are not logged.

To specify the default temporary sbspace, use the SBSPACETEMP configuration parameter.

Task	Manual
Use temporary sbspaces and smart large objects.	*IBM Informix Administrator's Guide*
Use **onspaces** to create temporary sbspaces.	*IBM Informix Administrator's Reference*
Improve temporary space utilization.	*IBM Informix Performance Guide*

Improved Space Allocation of User Data and Metadata in Sbspaces

The database server reserves 40 percent of the user-data space in the sbspace chunk. When the chunk runs out of metadata or user-data space, the database server moves some of the reserved space to the corresponding area. This feature enables the database server to use the space in the sbspace more efficiently.

Task	Manual
Monitor the metadata and user-data areas.	*IBM Informix Administrator's Guide*
Read about sbspace structure.	*IBM Informix Administrator's Reference*
Estimate the size of the metadata area and improve space utilization.	*IBM Informix Performance Guide*

J/Foundation Enhancements

You can create and execute UDRs and applications written in Java. The following improvement for this release is that J/Foundation performance is now faster.

For more information, see *J/Foundation Developer's Guide*, the *IBM Informix JDBC Driver Programmer's Guide*, and "Java Features in 9.21" on page 2-44.

JVM 1.3 Support in J/Foundation

Dynamic Server supports Java 2 and includes the Java Runtime Environment (JRE). The database server supports Version 1.3 of the Java Virtual Machine (JVM) and embeds the hotspot server VM.

Performance Enhancements

This release includes many features that help you monitor and improve performance.

Configurable Default Lock Modes

You can set the default lock mode to page or row for new tables in the following ways:

- LOCK MODE clause in the ALTER TABLE or CREATE TABLE statement
- **IFX_DEF_TABLE_LOCKMODE** environment variable
- DEF_TABLE_LOCKMODE configuration parameter

Task	Manual
Configure lock mode.	*IBM Informix Performance Guide*
Use the DEF_TABLE_LOCKMODE configuration parameter.	*IBM Informix Administrator's Reference*
Use the LOCK MODE clause in the ALTER TABLE or CREATE TABLE statement.	*IBM Informix Guide to SQL: Syntax*

The onstat -g stm Option

Use the **onstat -g stm** option to display the memory that prepared SQL statements use:

```
onstat -g stm session_id
```

For more information on **onstat -g stm**, see the *IBM Informix Performance Guide* and the *IBM Informix Administrator's Reference*.

The Ability to Display the Query Plan Without Executing the Query

To display the query plan without executing the query, use the SET EXPLAIN ON AVOID_EXECUTE statement or the AVOID_EXECUTE optimizer directive. This option allows you to evaluate the query plan that the optimizer has written to the **sqexplain.out** file.

To use this feature as a directive for a single statement:

```
SELECT --+EXPLAIN AVOID_DIRECTIVE
* FROM tablename;
```

To use this feature as a SET EXPLAIN keyword for a block of statements:

```
SET EXPLAIN ON AVOID_EXECUTE;
```

Task	Manual
Improve performance of queries and use optimizer directives.	*IBM Informix Performance Guide*
Use SET EXPLAIN and optimizer directives.	*IBM Informix Guide to SQL: Syntax*

Dynamic Addition of Logical Logs

The database server automatically adds a logical-log file after the current log file when the next log file contains an open transaction. Dynamic log allocation prevents logs from filling and hanging the system during long-transaction rollbacks. You also can choose whether to add a log file manually after the current log file or at the end of the log file list.

The DYNAMIC_LOGS configuration parameter determines whether the database server allocates new logical-log files dynamically. The LTXHWM and LTXEHWM configuration parameters set high-watermarks for long transactions. If DYNAMIC_LOGS is set to 1 or 2, the default LTXHWM value is 80 percent and LTXEHWM is 90 percent.

The **onstat -l** output also displays information about temporary logical logs.

Task	Manual
Use dynamically allocated logical logs.	*IBM Informix Administrator's Guide*
Use the **onparams** and **onstat -l** commands, and the DYNAMIC_LOGS, LTXHWM, and LTXEHWM parameters.	*IBM Informix Administrator's Reference*

SQL Enhancements

This release includes several new SQL statements that ease migration from non-Informix databases to Dynamic Server, Version 9.3.

Optional FROM in the DELETE Statement

The DELETE statement no longer requires the FROM keyword. You can use this syntax:

```
DELETE customer WHERE customer_num = 105;
```

For more information, see the *IBM Informix Guide to SQL: Syntax*.

REVOKE AS User

The REVOKE statement allows the owner of a database object to revoke the privileges of other users. REVOKE ... AS and REVOKE FRAGMENT .. AS allow *user2* to revoke the privileges for *user1*. For example, you can revoke privileges from user names such as **informix** that are authorization identifiers but not users that the operating system recognizes. You can use this syntax:

```
REVOKE privilege FROM user1 AS user2;
```

For more information on the REVOKE statement, see the *IBM Informix Guide to SQL: Syntax*.

Features from Dynamic Server, Version 9.21

These features were introduced in IBM Informix Dynamic Server, Version 9.21.

ANSI Join Syntax

You begin an ANSI join with the [LEFT] [OUTER] JOIN keywords, use the ON clause to specify the join filter, and use the WHERE clause to specify a post-join filter.

Task	Manual
Use the ANSI join syntax.	*IBM Informix Guide to SQL: Syntax*
Use join filters and post-join filters, and interpret SET EXPLAIN output for ANSI joins.	*IBM Informix Performance Guide*

Rename Index Statement

Use the RENAME INDEX statement to change the name of an index.

For more information, see the *IBM Informix Guide to SQL: Syntax*.

Nonlogging (RAW) Tables

You can create nonlogging tables in a logging database on Dynamic Server. These tables are also called raw tables. Raw tables cannot have indexes or referential constraints but can be updated. You can create either a standard or raw table and change tables from one type to another.

Task	Manual
Use nonlogging tables.	*IBM Informix Administrator's Guide*
Load and unload nonlogging tables. Lock nonlogging tables.	*IBM Informix Performance Guide*
Specify the logging type in the ALTER TABLE and CREATE TABLE statements.	*IBM Informix Guide to SQL: Syntax*

onpladm Utility

High-Performance Loader (HPL) includes the command-line utility **onpladm**. You can use the **onpladm** utility to create, modify, describe, list, run, configure, and delete jobs for unloading and loading tables or an entire database. For more information, see *IBM Informix High-Performance Loader User's Guide*.

The onbar -b -l Command

Use the **onbar -b -l** command instead of **onbar -l** to back up the logical logs.

For more information, see the *IBM Informix Backup and Restore Guide*.

9.x DB-Access to 7.x Synonyms

In earlier versions, you could use DB–Access to access synonym names only if the remote database server was Version 9.x. Now you can access synonym names on remote Version 7.x database servers.

SQL Statement Cache Improvements

The database server uses the SQL statement cache (SSC) to store SQL statements that a user executes. When users execute a statement stored in the SQL statement cache, the database server does not parse and optimize the statement again, which improves performance.

In Version 9.21, the SQL statement cache was enhanced to support the following capabilities:

- Insert the statement in the SQL statement cache as a *key-only* entry to track the number of times it has been referenced. After the statement has been referenced a specific number of times, it is inserted fully into the cache.
- Control whether statements enter the SQL statement cache after it exceeds its size limit.
- Define multiple pools for the SQL statement cache.

Task	Manual
Learn about the SQL statement cache	*IBM Informix Performance Guide*
Use qualifying and identical statements; also learn about the memory limits and key-only cache entries.	*IBM Informix Guide to SQL: Syntax*
Configure the SQL statement cache: • STMT_CACHE_HITS • STMT_CACHE_NOLIMIT • STMT_CACHE_NUMPOOL • STMT_CACHE_SIZE	*IBM Informix Administrator's Reference*
Display SQL statement cache statistics: • **onstat -g ssc** • **onstat -g ssc all** • **onstat -g ssc pool**	*IBM Informix Administrator's Reference*
Configure the SQL statement cache on the fly: • **onmode -W STMT_CACHE_HITS** • **onmode -W STMT_CACHE_NOLIMIT** • **onmode -W STMT_CACHE_SIZE**	*IBM Informix Administrator's Reference*
Understand the performance advantages of the SQL statement cache: • Use the SQL statement cache • Enable the SQL statement cache • Configure the SQL statement cache • Monitor the SQL statement cache	*IBM Informix Performance Guide*

DataBlade API Features

The following DataBlade API features were introduced in Version 9.21.

Functions for Controlling the Virtual-Processor Environment

The DataBlade API now provides functions that allow you to control the virtual-processor (VP) environment from within a UDR. These new functions provide the ability to:

- Obtain information about a VP.
- Obtain information about a VP class.
- Lock the UDR.
- Change the VP environment.

For more information, see the *IBM Informix DataBlade API Programmer's Guide*.

Functions for Getting Information About a UDR

The DataBlade API now provides functions that obtain additional information about a UDR, including the:

- Name of the UDR (as defined in the **sysprocedures** system catalog table)
- Routine identifier
- Address of the **MI_FPARAM** structure for the UDR

For more information, see the *IBM Informix DataBlade API Programmer's Guide*.

Java Features in 9.21

The following Java features were introduced in Version 9.21.

JVM 1.2 Support in J/Foundation

Dynamic Server supports Version 1.2 of the Java Virtual Machine (JVM).

Default Values of Java Configuration Parameters

The default values of the JDKVERSION, JVPJAVAHOME, JVPJAVALIB, and JVPJAVAVM parameters in the ONCONFIG file have changed for Dynamic Server.

JDBC 2.0 Support

IBM Informix JDBC driver is bundled with Embedded SQLJ 1.10.1.JC1, a product for embedding SQL statements in Java. Dynamic Server supports the following JDBC 2.0 features:

- Complex data types
- Collections
- Scrollable cursors
- Batch updates
- Interval data types
- Extensions to prepared statement
- Callable statements

GLS Support for J/Foundation

Dynamic Server supports the following GLS features:

- **CLIENT_LOCALE**, **DB_LOCALE**, **GL_DATE**, **GL_DATETIME**, and **DBTIME**, environment variables
- New connection properties (**NEWLOCALE** and **NEWCODESET**) for mapping a locale or code set in the JDBC driver

update_jars.sql Script

Use the **update_jars.sql** script to update the names of jar files in a database after you rename the database.

Java Runtime Environment Variables

Dynamic Server supports the **JVM_MAX_HEAP_SIZE**, **JAR_TEMP_PATH**, **JAVA_COMPILER**, and **AFDEBUG** environment variables.

Partial Support for Variable-Length Opaque-Types

You can now write UDRs and DataBlade modules in Java.

Dynamic Server supports the following items:

- Variable-length opaque data types
- Data I/O conversion routines:
  - input/output
  - send/receive
  - import/export

– importbin/exportbin

References to J/Foundation Features

For more information on J/Foundation features, see these manuals.

Task	Manual
Use JVM 1.2. Use JCBC 2.0 features. Write UDRs and DataBlade modules in Java.	*J/Foundation Developer's Guide*
Specify Java environment variables.	*J/Foundation Developer's Guide* *IBM Informix Guide to SQL: Reference*
Specify Java configuration parameters.	*J/Foundation Developer's Guide* *IBM Informix Dynamic Server Administrator's Reference*
Set GLS environment variables. Use the connection properties.	*J/Foundation Developer's Guide* *IBM Informix GLS User's Guide*
Use the **update_jars.sql** script.	*IBM Informix Guide to SQL: Syntax*

MaxConnect Support

IBM Informix MaxConnect enables IBM Informix Dynamic Server to support greatly increased numbers of client connections. MaxConnect is a new software tier, introduced between the database server and clients, that transparently funnels multiple client connections onto a smaller number of server connections. The database server is freed from managing thousands of client connections, which results in improved response time and decreased CPU cost for the database server.

Important: MaxConnect and the *IBM Informix MaxConnect User's Guide* ship separately from IBM Informix Dynamic Server, Version 9.3.

The following features were introduced in Version 9.21 to support the IBM Informix MaxConnect product, which is separately orderable:

- New network protocols

  The database server supports MaxConnect with two new network protocols: **ontliimc** and **onsocimc**.
- New utility options to monitor MaxConnect
  - **onstat -g imc**
  - **imcadmin**
  - **ISA options**
- New environment variables for MaxConnect
  - **IMCADMIN**
  - **IMCCONFIG**
  - **IMCSERVER**

For more information about installing, configuring, monitoring, and tuning MaxConnect, see the *IBM Informix MaxConnect User's Guide*.

Chapter 3. Using Existing Dynamic Server Features

In This Chapter . . . 3-2
Dynamic Scalable Architecture . . . 3-2
The Shared-Memory Component . . . 3-2
The Disk Component . . . 3-3
The Virtual Processor Component . . . 3-4
Client/Server Connections . . . 3-4
High Performance . . . 3-5
Memory Management . . . 3-5
Dynamically Sharing Memory . . . 3-5
Buffering Transactions . . . 3-5
Using NFS-Mounted Directories . . . 3-5
Fragmentation . . . 3-6
Parallelization . . . 3-6
Query Optimizer . . . 3-6
Fault Tolerance and High Availability . . . 3-6
Backup and Restore . . . 3-6
ontape Utility . . . 3-7
ON-Bar Utility . . . 3-7
IBM Informix Storage Manager . . . 3-7
The archecker Utility . . . 3-8
Fast Recovery . . . 3-8
Mirroring . . . 3-8
Data Replication . . . 3-8
High-Availability Data Replication . . . 3-8
Enterprise Replication . . . 3-8
Types of Data That You Can Replicate . . . 3-9
Database Server Security . . . 3-9
Auditing Database Events . . . 3-10
Informix RDBMS Features . . . 3-10
Structured Query Language (SQL) . . . 3-10
Stored Procedure Language (SPL) . . . 3-10
System Catalog Tables . . . 3-10
Data Types . . . 3-11
Application Types . . . 3-13
OLTP Applications . . . 3-13
DSS Applications . . . 3-13
Database Support . . . 3-14
Relational Databases . . . 3-14
ANSI-Compliant Databases . . . 3-14
Object-Relational Databases . . . 3-14
Simple and Smart Large Objects . . . 3-15
User-Defined Data Types . . . 3-16
Complex Data Types . . . 3-16
User-Defined Routines . . . 3-16
Operator Functions . . . 3-17
User-Defined Casts . . . 3-17
Inheritance . . . 3-17
User-Defined Aggregates . . . 3-17
User-Defined Virtual Processors . . . 3-17
DataBlade Modules . . . 3-18
Dimensional Databases . . . 3-18
Distributed Databases and Queries . . . 3-19
Access Methods . . . 3-19
Primary Access Methods . . . 3-19
Secondary Access Methods . . . 3-19

© Copyright IBM Corp. 1996, 2005

Generic B-Tree Indexes 3-20
R-Tree Indexes 3-20
User-Defined Primary Access Methods 3-20
User-Defined Secondary Access Methods 3-20

In This Chapter

This chapter provides an overview of Dynamic Server architecture and significant features. Dynamic Server delivers database scalability, manageability, and performance.

Dynamic Scalable Architecture

Dynamic Server is a multithreaded *object-relational database* server that uses symmetric multiprocessor (SMP) and uniprocessor architectures. In an SMP system, multiple CPUs or processors all run a single copy of the operating system, sharing memory and communications.

Dynamic scalable architecture (DSA) allows you to scale resources to varying application loads (from small to huge) and improves performance. Key elements of DSA are the virtual processors that manage central processing, disk I/O, networking, and optical functions in parallel.

Scalability has two aspects: speedup and scaleup. *Speedup* means the ability to add computing hardware and achieve faster performance for a decision support (DSS) query or online transaction processing (OLTP). *Scaleup* means the ability to process a larger workload with a correspondingly larger amount of computing resources in the same time. For more information on DSS and OLTP operations, see "Application Types" on page 3-13.

Informix database server architecture consists of the following main components:

- Shared memory
- Disk
- Virtual processor
- Client/server connections

For more information on database server architecture, see the *IBM Informix Performance Guide*. For information on how to use Dynamic Server, see the *IBM Informix Administrator's Guide* and the *IBM Informix Administrator's Reference*. For a glossary of terms that are used in IBM Informix manuals, see the *IBM Informix Guide to SQL: Reference*.

The Shared-Memory Component

Shared memory is an operating-system feature that lets the database server processes and threads 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
- To enable virtual processors and utilities to share data

The database server creates the following portions of shared memory:

- Resident

  Caches data from the disk for faster access

- Virtual

  Maintains and controls the resources required by the virtual processors
- Interprocess (IPC) communications

  Provide a fast communications channel for local client applications that use IPC communication on UNIX
- Virtual extension

  Enables DataBlade modules and user-defined routines (UDRs) to run in user-defined virtual processors

The Disk Component

The database server uses the physical units of storage to allocate disk space. You define the logical units that the database server uses to store data. All the databases and all the system information that is necessary to maintain the database server reside within the disk component.

On UNIX, the database server stores data in two 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* (also called *unbuffered* disk space) is unformatted space where the database server manages the physical organization of the data.
- *Cooked disk space* (also called *buffered* disk space) refers to regular operating-system files.

On Windows, the database server stores data in two types of disk space:

- New Technology File System (NTFS)
- Logical partition or physical drive

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

Physical Unit	Description
Chunk	The largest unit of database server data storage
Page	The physical unit of disk storage to read from and write to databases
Blobpage	The physical unit of disk storage to store simple large objects in a blobspace
Sbpage	The physical unit of disk storage to store smart large objects in an sbspace
Extent	A fixed amount of space to contain the data stored in a table

The database server uses the following logical units to manage disk space. Dbspaces, blobspaces, and sbspaces are composed of one or more chunks.

Logical Storage Unit	Description
Dbspace	Stores databases, tables, logical-log files, the physical log, and internal data
Blobspace	Stores simple large objects (TEXT and BYTE data)
Sbspace	Stores smart large objects (CLOB and BLOB data)
Extspace	References the location of external data
Database	Contains tables and indexes
Table	Consists of a row of column headings with zero or more rows of data values

Logical Storage Unit	Description
Tblspace	Contains the disk space allocated to a given table or fragment

The database server maintains the following storage structures to ensure physical and logical data consistency.

Data Consistency	Description
Logical log	A circular file that stores log records of transactions and database server changes
Physical log	A set of disk pages where the database server stores an unmodified copy of the page (called a *before image*)

For information about storage spaces and logical and physical logs, see the *IBM Informix Administrator's Guide*. Logical-log record formats are discussed in the *IBM Informix Administrator's Reference*.

The Virtual Processor Component

Database server processes are called *virtual processors* because they function like a CPU in a computer. Just as a CPU runs multiple operating-system processes to service multiple users, a virtual processor runs multiple *threads*, or pieces of work, to service multiple SQL client applications. Virtual processors improve database server performance.

Client/Server Connections

You can put a client on one computer and the database server on another or the same computer. A *client* is an application that a user runs to request or modify information from a database by issuing SQL statements. The following IBM Informix tools are client programs:

- DB–Access
- Enterprise Replication
- High Performance Loader (HPL)
- ESQL/C
- IBM Informix JDBC Driver
- ODBC
- DataBlade API

The database administrator specifies the types of connections that the database server supports in the **sqlhosts** file on UNIX or the PROTOCOL field in the SQLHOSTS registry key on Windows.

Use a *network protocol* to connect to and transfer data between database servers, or between a client and a database server. You must establish a *connection* between the client and database server before data transfer can take place and you must maintain it for the duration of the data transfer.

A *multiplexed connection* uses a single network connection between the database server and a client to handle multiple database connections from the client. If you need to manage several hundred or thousands of client connections, consider ordering IBM Informix MaxConnect. For details, see "IBM Informix MaxConnect (UNIX)" on page 1-6.

The database server supports the following types of connections to communicate between client applications and a database server.

Connection Type	Windows	UNIX	Local	Network
Sockets	X	X	X	X
TLI (TCP/IP)		X	X	X
TLI (IPX/SPX)		X	X	X
Shared memory		X	X	
Stream pipe		X	X	
Named pipe	X		X	

For information about client/server configurations that the database server supports, see the *IBM Informix Administrator's Guide*. For instructions on how to use client applications, see the appropriate programmer's manual, as listed in Chapter 6, "Using the Documentation," on page 6-1.

High Performance

Dynamic Server achieves high performance through the following mechanisms:

- Memory management
- Fragmentation
- Parallelization
- Query optimization

Memory Management

Dynamic Server provides several options to help you manage memory to optimize performance.

Dynamically Sharing Memory

All applications that use the same database server share data in the memory space of the database server. The database server adds memory dynamically as it needs it. The database server administrator can control the amount of shared memory that is available to the database server.

Buffering Transactions

You can determine how the database server logs transactions. A *transaction* is a collection of SQL statements that is treated as a single unit of work. 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 how to manage the various aspects of memory to increase performance, see your *IBM Informix Performance Guide* and "SQL Enhancements" on page 2-41. For information on transaction logging, see the *IBM Informix Administrator's Guide*.

Using NFS-Mounted Directories

An IBM Informix storage space can reside on an NFS-mounted directory only if the vendor of that NFS device is certified by IBM Informix. For information about NFS products that you can use to mount a storage space for an IBM Informix database server, see the IBM Informix product family Web site at http://www.ibm.com/software/data/informix.

Fragmentation

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

Dynamic Server supports the following fragmentation schemes:

- *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.

For information on the fragmentation strategies, see the *IBM Informix Database Design and Implementation Guide*. For information about how to create a fragmentation strategy to enhance database performance, see your *IBM Informix Performance Guide*.

Parallelization

The database server can allocate multiple threads to work in parallel on a single query. This feature is known as parallel database query (PDQ).

PDQ can improve performance dramatically when the database server processes queries that DSS applications initiate. PDQ lets the database server distribute the work for one aspect of a query among several processors.

For information on how to implement PDQ and how parallelization can enhance performance, see your *IBM Informix Performance Guide*. For information on the SET PDQPRIORITY environment variable, see the *IBM Informix Guide to SQL: Reference*.

Query Optimizer

The *query optimizer* formulates a query plan 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 must determine whether indexes should be used. If the query includes a join, the optimizer must determine the join plan (*hash* or *nested loop*) and the order in which tables are evaluated or joined.

For more information on the optimizer, see your *IBM Informix Performance Guide*.

Fault Tolerance and High Availability

Dynamic Server uses the following logging and recovery mechanisms to protect data integrity and consistency if an operating-system or media failure occurs:

- Backup and restore
- Fast recovery
- Mirroring
- High-Availability Data Replication (HDR)
- Enterprise Replication

Backup and Restore

Use the ON–Bar or **ontape** utility to back up your database server data and logical logs as insurance against lost or corrupted data. A program error or disk failure

can cause data loss or corruption. If a dbspace, an entire disk, or the database server goes down, use ON–Bar or **ontape** to restore the data from the backup copy. You must use the same utility for both the backup and restore.

The following are basic backup and restore terms:

- A *backup* is a copy of one or more *storage spaces* and the logical logs.
- 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 backups.
- A *restore* re-creates data from a backup.
- A *point-in-time restore* allows you to restore the data in a database to a specific time.

  A point-in-time restore can undo mistakes, such as dropping a table, that might not be fixable otherwise.

ontape Utility

The **ontape** utility does not require a storage manager. Use **ontape** to perform the following tasks:

- Back up and restore storage spaces and logical logs.
- Change database-logging status.
- Start continuous logical-log backups.
- Use data replication.
- Rename chunks to different pathnames and offsets.

ON-Bar Utility

The ON–Bar utility requires a storage manager such as IBM Informix Storage Manager (ISM). Use ON–Bar to perform the following tasks:

- Back up and restore storage spaces and logical logs.
- Perform point-in-time restores.
- Start continuous logical-log backups.
- Verify a backup with the **archecker** utility.
- Perform external backups and restores.

  An *external backup and restore* allows you to copy and physically restore data without using ON–Bar. Then you use ON–Bar for the logical restore.
- Rename chunks to different pathnames and offsets.

For information about backing up data with ON–Bar or **ontape** and the **archecker** utility, see the *IBM Informix Backup and Restore Guide*.

IBM Informix Storage Manager

IBM Informix Storage Manager (ISM) manages data storage for the Informix database server. ISM resides on the same computer as ON–Bar and the database server.

ISM receives backup and restore requests from ON–Bar and directs data to and from storage volumes that are mounted on storage devices. ISM

ui32D 4453 961297399 TTf 9.978

- A *user-defined procedure* is a routine that optionally accepts a set of arguments but does not return any values. A procedure cannot be used in SQL expressions because it does not return a value.

The database server supports UDRs written in the following languages:

- *Stored Procedure Language (SPL),* a language that is internal to the database server
- *External languages*, such as the C or Java languages

For information on implementing user-defined routines, see *IBM Informix User-Defined Routines and Data Types Developer's Guide*.

Operator Functions

An *operator function* is an SQL-invoked function that has a corresponding operator symbol (such as '=' or '+'). These operator symbols are used within expressions in an SQL statement.

The database server provides operator functions for most built-in data types. You can extend an existing operator to operate on a user-defined data type.

For information on extended operations, see *IBM Informix User-Defined Routines and Data Types Developer's Guide*.

User-Defined Casts

A *cast* performs a conversion between two data types. The database server provides casts between the built-in data types. For example, when you add an integer value to a decimal value, the database server performs a cast to change the integer into a decimal so that it can perform the addition.

You can write user-defined cast functions to convert between an existing data type and an extended data type that you create.

For information on implementing user-defined casts, see *IBM Informix User-Defined Routines and Data Types Developer's Guide*.

Inheritance

Inheritance lets you define objects (types and tables) that acquire the properties of other objects and add new properties that are specific to the object that you define.

User-Defined Aggregates

Use a *user-defined aggregate* (UDA) to perform any kind of aggregate computation on a column, such as the average or the count. You can either create a user-defined aggregate or extend an existing aggregate for extended data types.

For the SQL syntax to create and drop UDAs, see the *IBM Informix Guide to SQL: Syntax*. For information on using UDAs, see *IBM Informix User-Defined Routines and Data Types Developer's Guide*.

User-Defined Virtual Processors

You can designate a user-defined virtual processor to run DataBlade modules or UDRs written in the C language. Designate a Java virtual processor to run UDRs written in the Java language.

For information on virtual processors, see the *IBM Informix Administrator's Guide* and *IBM Informix User-Defined Routines and Data Types Developer's Guide*.

DataBlade Modules

IBM and other vendors package some data types and their access methods into DataBlade modules (shared class libraries) that you can add to the database server to store and access nontraditional data types such as two-dimensional spatial objects (lines, polygons, ellipses, and circles), 3D images, sound, video, electronic documents, HTML pages, and time-series data. A DataBlade module might also provide new types of access to large text documents, including phrase matching, fuzzy searches, and synonym matching.

You can do the following:

- Add an IBM Informix or third-party DataBlade module, which is a pre-packaged custom data type.
- Create your own DataBlade module with the DataBlade Developer's Kit.

For information on how to work with and create your own DataBlade modules, see the *IBM Informix DataBlade API Programmer's Guide*, the *IBM Informix DataBlade API Function Reference*, and the *IBM Informix DataBlade Developer's Kit User's Guide*.

Dimensional Databases

Dynamic Server supports *data warehouses* and *data marts*. This typically involves a *dimensional* database that contains large stores of historical data. The databases that track your grocery purchases and voting trends in your state are examples of data warehouses.

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 time. Existing records in a dimensional database are updated infrequently. This type of informational processing is known as online analytical processing (OLAP) or decision-support processing.

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

- Data warehouse

  A database that is optimized for data retrieval

  Data is not stored at the transaction level; some level of data is summarized.
- Data mart

  A subset of a data warehouse that is stored in a smaller database and that is oriented toward a specific purpose or subject rather than enterprise-wide strategic planning

  A data mart can contain operational data, summarized data, spatial data, or metadata.
- Operational data store

  A subject-oriented system that is optimized for looking up one or two records at a time for decision making

  An operational data store is a hybrid form of data warehouse that contains timely, current, integrated information. This data can serve as the common source of data for data warehouses.
- Repository

  A repository combines multiple data sources into one normalized database

  The records in a repository are updated frequently. Data stored in a repository is operational, not historical.

For details on how to plan, build, and implement a dimensional database, see the *IBM Informix Database Design and Implementation Guide*.

Distributed Databases and Queries

Dynamic Server supports distributed queries across multiple databases and multiple database servers for transactions that involve only built-in data types. To issue a distributed query, a client application connects to a single database server, called the *local database server*, and specifies a database, called the *local database*. By default, all the database objects that you reference come from the local database.

All other databases are *external databases*. All other database servers are *remote database servers.* A database on a remote database server is an *external remote database*.

When the external database is on the same database server as the local database, you must qualify the object name with the external database name (for example, **salesdb:contacts**). When the external database is on a remote database server, you must qualify the object name with the remote database server name and the external remote database name (**salesdb@distantserver:contacts**).

The database server supports two multiphase protocols, *two-phase commit* and *heterogeneous commit*, to process transactions that span multiple database servers.

For information about using distributed queries, see *IBM Informix User-Defined Routines and Data Types Developer's Guide*. For information about two-phase commit and heterogeneous commit protocols, see the *IBM Informix Administrator's Guide*. For information about a specific IBM Informix Enterprise Gateway product, see the appropriate *IBM Informix Enterprise Gateway User Manual*.

Access Methods

An *access method* is a set of database server functions that a database server uses to access and manipulate a table or an index. Dynamic Server supports *primary* and *secondary access methods*. You can write routines that provide R-tree indexing and custom primary and secondary access methods.

Primary Access Methods

The *primary access method* handles storage and retrieval of a particular data type in a table. If the primary access method does not handle a particular data type, the database server cannot access values of that type. Dynamic Server provides all of the necessary routines for accessing the built-in data types.

For information on how to use primary access methods, see the *IBM Informix Guide to SQL: Syntax* and *IBM Informix User-Defined Routines and Data Types Developer's Guide*.

Secondary Access Methods

The *secondary access method* handles all indexing operations for a particular data type. If the *operator class* of a secondary access method does not handle a particular data type, you cannot build an index on that data type.

Dynamic Server provides two built-in secondary access methods:

- Generic B-trees
- R-trees

Generic B-Tree Indexes

A B-tree index organizes index information. A B-tree index is arranged as a hierarchy of pages. Dynamic Server uses a B-tree index for the following values:

- Columns that contain built-in data types (known as a *traditional B-tree index*)

  Built-in data types include CHARACTER, DATETIME, INTEGER, FLOAT, and so forth.
- One-dimensional user-defined data types (known as a *generic B-tree index*)
- Values that a user-defined function returns (known as a *functional index*)

For more information on B-trees and functional indexes, see your *IBM Informix Performance Guide*.

R-Tree Indexes

The *R-tree* indexing structure supports spatial data. An R-tree index uses a *bounding box,* which is a set of coordinates that contains one or more objects and supports spatial data (two-dimensional, three-dimensional, and so on). An object can theoretically belong to more than one bounding box. An R-tree index is useful for searches on multidimensional data.

For information about R-trees, see the *IBM Informix R-Tree Index User's Guide*.

User-Defined Primary Access Methods

Dynamic Server supports *external spaces* (*extspaces),* which are storage spaces that the database server does not manage directly. Use **onspaces -c -x** to specify an external space as the storage space for a table for which you create a primary access method.

You can access the following types of data with a primary access method:

- Database tables from other vendors
- Data stored in sequential files
- Remote data stored across a network

For information on creating extspaces, see the *IBM Informix Administrator's Guide*, and *IBM Informix Administrator's Reference*. For information on how to create primary access methods, see the *IBM Informix Virtual-Table Interface Programmer's Guide*.

User-Defined Secondary Access Methods

In many cases, index data is stored outside the Informix dbspace. However, you can build an access method for data stored as a large object in an sbspace. The database server can use a virtual index transparently to access data in an Informix table. Use this method to create an alternative indexing strategy for specialized data types.

For information on how to create secondary access methods, see the *IBM Informix Virtual-Index Interface Programmer's Guide*.

Chapter 4. Installing, Administering, and Tuning the Database Server

In This Chapter 4-1
Database Server Users 4-1
Planning, Installing, and Configuring the Database Server 4-1
Administering the Database Server 4-3
Monitoring Performance 4-6
Troubleshooting the Database Server 4-7

In This Chapter

This chapter describes the tasks that a database server administrator is likely to perform and where to find information on those tasks throughout the documentation set. The task matrixes in this book have the following columns:

- **If You Want To**. This column describes a task you might want to perform.
- **Manual**. This column lists the book that contains information to help you perform the task.

Database Server Users

Table 4-1 shows the major groups of database server users.

Table 4-1. Database Server Users

User	Duties
Database administrator (DBA)	A DBA is primarily responsible for creating, managing, and controlling access for databases.
Database server administrator	The database server administrator is responsible for the installation, configuration, maintenance, administration, and operation of the database server that might manage many individual databases.
Performance specialist	The performance specialist optimizes and tunes the database server and query performance.
Programmers and application developers	Programmers and application developers develop applications, DataBlade modules, and user-defined routines in C, C++, or Java.
Operator	The operator is responsible for backing up and restoring databases and for carrying out routine database server administration tasks.
Database user	Database users access, insert, update, and manage information in databases with SQL, which is often embedded in a client application.

Planning, Installing, and Configuring the Database Server

When you begin working with a new database server, you need to perform the following tasks:

- Set up and configure system hardware and software.
- Install the database server and client applications.
- Migrate data from an earlier version of the database server (if needed).

© Copyright IBM Corp. 1996, 2005

- Configure the environment.
  - Set required environment variables.
  - Prepare connectivity files.
  - Prepare the configuration file.
  - Allocate and initialize disk space.
- Choose a database type.
- Create the demonstration database (optional).

Table 4-2 describes the planning, installation, and configuration tasks.

Table 4-2. Planning, Installation, and Configuration Tasks

If You Want To	Manual
Learn about the new features in Dynamic Server.	Chapter 2, "Using New Features in Dynamic Server," on page 2-1
Acquaint yourself with terms used in IBM Informix manuals.	*IBM Informix Guide to SQL: Reference*
Interpret error messages.	*IBM Informix Error Messages* or **finderr** utility
Plan a database server installation.	*IBM Informix Administrator's Guide*
Plan and configure: • Operating system • Hardware and system software upgrades • Network capacity • Integration with other vendor products and applications • Disk and storage media	System documentation
Determine optimal memory configuration. Determine optimal disk layout and striping.	*IBM Informix Performance Guide* System documentation
Install Dynamic Server on UNIX or Linux: • Standard installation • Silent installation	*IBM Informix Dynamic Server Installation Guide for UNIX and Linux* or the *Quick Beginnings for IBM Informix Dynamic Server Express Edition*
Install Dynamic Server on Windows: • Standard installation • Silent installation • Multiple residency • Cluster installation	*IBM Informix Dynamic Server Installation Guide for Microsoft Windows* or the *Quick Beginnings for IBM Informix Dynamic Server Express Edition*
Initialize the database server. Manage database server operating modes.	*IBM Informix Administrator's Guide*
Test the database server connection with DB–Access.	*IBM Informix DB–Access User's Guide*
Install and configure client applications.	*IBM Informix Client Products Installation Guide*
Install and configure DataBlade modules (optional).	*IBM Informix DataBlade Module Installation and Registration Guide*

Table 4-2. Planning, Installation, and Configuration Tasks (continued)

If You Want To	Manual
Install and configure MaxConnect (optional).	*IBM Informix MaxConnect User's Guide*
Configure the database server manually: • Set environment variables. • Set ONCONFIG parameters. • Configure J/Foundation (optional). • Configure client/server connectivity. • Set up multiple instances of the database server. • Test the database server configuration.	*IBM Informix Administrator's Guide*
Use **Server Setup** in IBM Informix Server Administrator to configure the database server.	ISA online help
Create dbspaces, blobspaces, and sbspaces. Resolve incorrect chunk permissions and ownership. Design and implement logical and physical logs. Implement mirroring.	*IBM Informix Administrator's Guide*
Configure the ON–Bar or **ontape** backup and restore system.	*IBM Informix Backup and Restore Guide*
Set up IBM Informix Storage Manager. Set up the storage volumes and storage devices.	*IBM Informix Storage Manager Administrator's Guide*
Set up a third-party storage manager (optional).	Your storage-manager documentation
Design and set up the Enterprise Replication system.	*IBM Informix Dynamic Server Enterprise Replication Guide*
Design and set up a High-Availability Data Replication (HDR) system.	*IBM Informix Administrator's Guide*
Prepare the old database server version for migration. Migrate to Dynamic Server from an earlier database server version. Move data among different physical equipment (computer and storage devices) and different operating systems. Move data among database servers that have different language support. Work with these utilities: **dbexport**, **dbimport**, **dbload**, **dbschema**, **onload**, **onunload**, **onmode -b**. Revert from Dynamic Server to an earlier database server version.	*IBM Informix Migration Guide*

Administering the Database Server

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

- Prepare the operating system to automatically start and stop the database server when the system is shut down or rebooted.

- Back up and restore storage spaces (dbspaces, blobspaces, and sbspaces) and logical logs.

  When you plan your backup schedule, consider the availability of backup devices and operators to perform backups.
- Check that users have set the correct environment variables.
- Review the database server configuration parameters.
- Transfer data that was created on other Informix database servers.

Table 4-3 lists the administration tasks and where to find information on those tasks.

Table 4-3. Administration Tasks

If You Want To	Manual
Monitor an Informix database server. Configure client/server connections. Manage virtual processors, shared memory, and storage spaces. Manage temporary space usage and table extents. Manage database-logging status, logical-log files, and the physical log. Monitor and manage sbspaces. Resolve long transaction problems. Perform fast recovery and checkpoints. Perform mirroring operations. Verify database consistency using **oncheck** commands. Use High-Availability Data Replication. Understand two-phase and heterogeneous commit protocols. Recover manually from a failed two-phase commit.	*IBM Informix Administrator's Guide*
Use ISA to administer and monitor the database server.	ISA online help
Use the following utilities to perform administrative tasks: • **oncheck** • **ondblog** • **oninit** • **onlog** • **onmode** • ON–Monitor • **onparams** • **onspaces** • **onstat**	*IBM Informix Administrator's Reference*
Locate information on the configuration parameters. Use the SMI tables of the **sysmaster** database to monitor the database server.	*IBM Informix Administrator's Reference*

Table 4-3. Administration Tasks (continued)

If You Want To	Manual
Interpret logical-log records and message-log messages. Understand database server disk structures and storage. See a list of the files that the database server uses. Work with event alarms.	*IBM Informix Administrator's Reference*
Use the ON–Bar or **ontape** utility. Back up and restore storage spaces and logical logs. Use the **archecker** utility to verify backed-up data. Perform an external backup and restore. Set up HDR with external backup and restore.	*IBM Informix Backup and Restore Guide*
Use ISA to administer and monitor the database server.	ISA online help
Connect your database server to storage devices for ON–Bar backup and restore operations. Issue ISM commands. Manage backup media and storage devices. Track the location of all backup data. Move backup data through a managed life cycle. Provide disaster recovery for a database server instance. Perform an imported restore to a database server on another computer.	*IBM Informix Storage Manager Administrator's Guide*
Use the **ipload**, **onpladm**, and **onpload** utilities to load or unload large quantities of data to or from an Informix database. Use the High-Performance Loader (HPL) GUI. Move data to a different computer or configuration. Alter the schema of a table.	*IBM Informix High-Performance Loader User's Guide*
Detect unusual user actions and unwanted activities and identify the perpetrators. Detect unauthorized access attempts. Assess potential security compromises. Use the secure-auditing utilities (**onaudit**, **onshowaudit**) to set up, administer, and interpret audit trails.	*IBM Informix Trusted Facility Guide*
Use the Optical Subsystem interface for an optical storage subsystem to store TEXT and BYTE data (simple large objects) on optical platters (WORM optical media). Use SQL statements to store and retrieve data to and from the optical storage subsystem.	*IBM Informix Optical Subsystem Guide*
Use the IBM Informix SNMP subagent to extract information from an Informix database server and pass that information to a network manager.	*IBM Informix SNMP Subagent Guide*

Table 4-3. Administration Tasks (continued)

If You Want To	Manual
Design, define, monitor, and control your Enterprise Replication system.	*IBM Informix Dynamic Server Enterprise Replication Guide*
Set up locales for different languages, cultural conventions, and code sets.	*IBM Informix GLS User's Guide*

Monitoring Performance

After the database server is up and running, the database server administrator or performance specialist is responsible for maintaining the optimum performance of the database server and database applications, as follows:

- Monitor system resources that are critical to performance.
- Identify database activities that affect these critical resources.
- Identify and monitor queries that are critical to performance.
- Use the database server utilities for performance monitoring and tuning.
- Optimize query execution.
- Eliminate performance bottlenecks in the following ways:
  - Balance the load on system resources.
  - Adjust the configuration of your database server.
  - Adjust the arrangement of your data.
  - Allocate resources for decision-support queries.
  - Create indexes to speed up retrieval of your data.

Table 4-4 lists performance-related tasks and where to find information on those tasks.

Table 4-4. Performance Tuning Tasks

If You Want To	Manual
Use different types of tables (STANDARD, RAW, TEMP).	*IBM Informix Administrator's Guide*
Use the **onstat -g** utilities to monitor database server performance.	*IBM Informix Administrator's Reference*
Improve backup and restore performance.	*IBM Informix Backup and Restore Guide*
Query the system catalog tables.	*IBM Informix Guide to SQL: Reference*

Table 4-4. Performance Tuning Tasks (continued)

If You Want To	Manual
Adjust the database server configuration.	*IBM Informix Performance Guide*
Allocate resources for DSS or OLTP systems.	
Balance the load on system resources.	
Collect performance statistics.	
Control the placement and size of tables and table extents.	
Create and manage indexes to speed up retrieval of data.	
Design and use parallel database queries (PDQ).	
Eliminate database server performance bottlenecks.	
Fragment tables for improved performance.	
Identify and monitor queries that are critical to performance.	
Improve checkpoint performance and manage LRU queues.	
Improve the performance of a query.	
Manage data distributions.	
Monitor critical system resources (CPU, memory, disk, virtual processors).	
Monitor and track locking and isolation levels.	
Optimize the disk layout.	
Tune buffer cache.	
Use case studies to tune performance.	
Use optimizer directives and SET EXPLAIN to optimize query plans.	
Use secondary access methods such as B-trees.	
Use the **onperf** utility for performance monitoring and tuning.	
Use the SQL statement cache.	
Use UPDATE STATISTICS.	
Write complex SQL statements, including outer joins and subqueries.	

Troubleshooting the Database Server

Usually, the database server runs smoothly, but when something goes wrong or a puzzling error message displays, an array of diagnostic tools is available to help you solve the problem. Technical Support also can assist you in troubleshooting and fixing problems with Dynamic Server.

Table 4-5 on page 4-8 describes the diagnostic tools available for troubleshooting database operations and the database server.

Table 4-5. Troubleshooting Tasks

If You Want To	Manual
Use the **onstat -g** utilities to diagnose database server problems. Use the **onmode -I** option to collect diagnostic information. Use event alarms to automatically trigger administrative actions. Find corrective actions to unnumbered error messages. Collect diagnostic dumps using the DUMP* configuration parameters.	*IBM Informix Administrator's Reference*
Use the **archecker** utility to verify and diagnose problems with backups. Find corrective actions to ON–Bar return codes.	*IBM Informix Backup and Restore Guide*
Find corrective actions to numbered error messages and ON–Bar messages.	*IBM Informix Error Messages*
Fix data replication problems.	*IBM Informix Dynamic Server Enterprise Replication Guide*

Chapter 5. Designing, Maintaining, and Extending the Database

In This Chapter 5-1
Designing, Developing, and Extending the Database 5-1
Developing Application Programs that Access the Database 5-3

In This Chapter

This chapter describes the tasks that database administrators (DBA) and application developers are likely to perform and where to find information on those tasks.

Designing, Developing, and Extending the Database

Table 5-1 lists the tasks for designing, developing, and extending the database.

Table 5-1. Database Tasks

If You Want To	Manual
Work with the tables in the **sysmaster** database.	*IBM Informix Administrator's Reference*
Place tables on disk.	*IBM Informix Performance Guide*
Estimate table size and manage table extents.	
Modify tables (truncate, alter, modify columns, load, attach fragments).	
Denormalize data to improve performance.	
Create and manage B-tree indexes.	
Work with specialized indexes (R-tree and secondary access methods).	
Set appropriate lock modes and monitor locks.	
Design a fragmentation strategy (round-robin or expression-based).	
Fragment indexes and temporary tables.	
Use the WHERE clause and joins to filter queries.	
Find corrective actions to error messages.	*IBM Informix Error Messages*

© Copyright IBM Corp. 1996, 2005

Table 5-1. Database Tasks (continued)

If You Want To	Manual
Design a database (choose whether to implement the relational, object-relational, or dimensional database model).	*IBM Informix Database Design and Implementation Guide*
Build a relational or object-relational database: • Define the data objects. • Create an entity-relationship diagram. • Normalize the data. • Create and populate the database.	
Build and implement a dimensional database for data warehousing.	
Choose data types for the database.	
Set up check constraints and referential constraints.	
Determine the primary keys and foreign keys in tables.	
Extend a database with user-defined casts.	
Understand type and table inheritance.	
Grant and limit access to a database.	
Use views and privileges.	
Define a fragmentation strategy or distribution schema.	
Invoke the DB–Access utility.	*IBM Informix DB–Access User's Guide*
Connect to or create one or more databases and transfer data between a database and external text files.	
Display information about databases and verify database server status.	
Perform ad hoc queries that you execute once or infrequently.	
Execute and debug SQL statements and SPL routines. Display system catalog tables and the Information Schema.	
Access, modify, and retrieve information from the database server.	
Use menus, screens, SQL statements, and SPL routines to view, access, retrieve, store, and modify data in a database.	
Work with relational (**stores_demo**) and object-relational (**superstores_demo**) demonstration databases.	
Learn how GLS affects database server migration.	*IBM Informix Migration Guide*
Load and unload data.	
Display the database schema with **dbschema**.	
Use the system catalog tables to track objects.	*IBM Informix Guide to SQL: Reference*
Set environment variables.	
Find a description of the tables in the **stores_demo** or **superstores_demo** database.	
Look up definitions in the glossary.	

Table 5-1. Database Tasks (continued)

If You Want To	Manual
Create databases and manage access. Compose correct SQL statements. Learn the categories of SQL statements. Use segments such as arguments, expressions, and identifiers. Write procedures with SPL and store them in a database. Look up reserved words.	*IBM Informix Guide to SQL: Syntax*
Learn database concepts. Compose basic and advanced SELECT statements. Use functions and SPL routines in SQL statements. Modify data in a database. Set locks. Work with casts on extended data types. Create and use triggers. Use embedded SQL in programs.	*IBM Informix Guide to SQL: Tutorial*
Assign data types to columns.	*IBM Informix Database Design and Implementation Guide* *IBM Informix Guide to SQL: Reference* *IBM Informix Guide to SQL: Syntax* *IBM Informix Guide to SQL: Tutorial*
Use the Optical Subsystem interface for an optical storage subsystem to store TEXT and BYTE data (simple large objects) on optical platters (WORM optical media). Use SQL statements to store and retrieve data to and from the Optical Subsystem.	*IBM Informix Optical Subsystem Guide*

Developing Application Programs that Access the Database

Table 5-2 lists the tasks for developing, compiling, and running client application programs and DataBlade modules that access database server data.

Table 5-2. Application Development Tasks

If You Want To	Manual
Test database applications that you intend to store for use in a production environment.	*IBM Informix DB–Access User's Guide*
Write procedures with SPL and store them in a database. Use a primary access method.	*IBM Informix Guide to SQL: Syntax*
Use embedded SQL in programs. Program in a multiuser environment. Create and use routines with SPL. Work with user-defined and system-defined casts on extended data types.	*IBM Informix Guide to SQL: Tutorial*

Table 5-2. Application Development Tasks (continued)

If You Want To	Manual
Use IBM Informix ODBC Driver to access relational databases with SQL. Create custom applications with IBM Informix ODBC API functions.	*IBM Informix ODBC Driver Programmer's Manual*
Embed SQL statements directly into C programs.	*IBM Informix ESQL/C Programmer's Manual*
Create new data types and user-defined routines using Java.	*J/Foundation Developer's Guide*
Use the GLS features that let Informix SQL APIs and database servers handle different languages, cultural conventions, and code sets.	*IBM Informix GLS User's Guide*
Work with the TP/XA library in an X/Open distributed transaction-processing (DTP) environment. Develop applications for a third-party transaction manager and an Informix database server.	*IBM Informix TP/XA Programmer's Manual*
Create new data types and user-defined routines using C. Define new data types or extend the functionality of existing data types. Extend operations on data types, create new casts, extend operator classes for secondary access methods, and create opaque data types for your database or DataBlade programs. Create application-specific SPL or external routines for application end-users. Create and register a user-defined routine (UDR) to invoke within an SQL statement or another routine.	*IBM Informix User-Defined Routines and Data Types Developer's Guide*
Use DataBlade API functions to develop server and client applications that access data stored in a Dynamic Server database. Write server routines and client LIBMI applications that use smart large objects and complex and extended data types.	*IBM Informix DataBlade API Programmer's Guide*
Use DataBlade API functions. Use ESQL/C functions with the DataBlade API.	*IBM Informix DataBlade API Function Reference*
Use Java to create client applications or applets that run against Dynamic Server. Install and load the IBM Informix JDBC Driver. Use standard JDBC to connect to a database or database server. Use standard JDBC to send queries, retrieve results, get database and column metadata, and handle errors. Learn how standard Java data types map to Informix data types. Store and retrieve XML documents. Use the IBM Informix HTTP proxy servlet. Debug JDBC API programs. Improve query performance in your JDBC applications.	*IBM Informix JDBC Driver Programmer's Guide*
Use the object-oriented C++ programming language to create database client applications for Informix database servers. Use Object Interface for C++to create value objects that let C++ client applications support DataBlade module data types.	*IBM Informix Object Interface for C++ Programmer's Guide*
Work with the R-tree secondary access method.	*IBM Informix R-Tree Index User's Guide*

Table 5-2. Application Development Tasks (continued)

If You Want To	Manual
Develop a secondary access method with the Virtual-Index Interface (VII) to create new types of indexes. Use functions in the VII library.	*IBM Informix Virtual-Index Interface Programmer's Guide*
Develop a primary access method with the Virtual-Table Interface (VTI) so that users can access external data. Use functions in the VTI library.	*IBM Informix Virtual-Table Interface Programmer's Guide*
Develop applications using DataBlade modules.	"DataBlade Manuals" on page 6-4

Chapter 6. Using the Documentation

In This Chapter 6-1
The IBM Informix Documentation Set 6-1
IBM Informix Dynamic Server Manuals 6-1
Client SDK and Connectivity Manuals 6-3
DataBlade Manuals 6-4

In This Chapter

This chapter contains an alphabetical list of the IBM Informix manuals provided.

The IBM Informix Documentation Set

This set describes all the manuals available with the database server, client products, tools, and DataBlade modules.

IBM Informix Dynamic Server Manuals

Table 6-1 summarizes the documentation that is available with Dynamic Server.

Table 6-1. Database Server Manuals

Book Title	Description
IBM Informix Backup and Restore Guide	This manual explains the concepts and methods required to back up and restore data with the ON–Bar and **ontape** utilities. It includes information on the **archecker** utility.
IBM Informix Database Design and Implementation Guide	This guide documents how to design, implement, and manage Informix databases. It includes data models that illustrate different approaches to database design and shows you how to use SQL to implement and manage databases.
IBM Informix DataBlade API Function Reference	This manual describes the API functions.
IBM Informix DataBlade API Programmer's Guide	This manual describes the API, the C-language application programming interface that is provided with Dynamic Server. Use the API to develop client and server applications that access data stored in a Dynamic Server database.
IBM Informix DB–Access User's Guide	This guide describes how to use the DB–Access utility to access, modify, and retrieve information from Informix database servers.
IBM Informix Dynamic Server Administrator's Guide	This user guide for system and database server administrators discusses the concepts and procedures for managing Dynamic Server. It is intended to help you understand, configure, and use the database server. The short manual name is *IBM Informix Administrator's Guide*.
IBM Informix Administrator's Reference	This reference manual provides the syntax of database server utilities such as **onmode** and **onstat**, and comprehensive descriptions of configuration parameters, SMI tables in the **sysmaster** database, logical-log records, disk structures, files that the database server uses, trapping errors, event alarms, and message-log messages.
IBM Informix Dynamic Server Getting Started Guide	This guide provides an overview of IBM Informix products, summarizes the new features in this release, and provides a roadmap to user tasks in the documentation set for the database server.

© Copyright IBM Corp. 1996, 2005

Table 6-1. Database Server Manuals (continued)

Book Title	Description
IBM Informix Dynamic Server Enterprise Replication Guide	This guide contains information to help you understand the concepts of data replication, design your own Enterprise Replication system, install Enterprise Replication, and administer and manage data replication throughout your enterprise.
IBM Informix Dynamic Server Installation Guide for UNIX and Linux	This guide contains instructions for installing Dynamic Server on UNIX and Linux. It also describes how to solve common installation problems.
IBM Informix Dynamic Server Installation Guide for Microsoft Windows	This guide contains instructions for installing Dynamic Server on Windows.
IBM Informix Dynamic Server Performance Guide	This guide explains how to configure and operate Dynamic Server to achieve optimum performance and optimize SQL queries. The short manual name is *IBM Informix Performance Guide*.
IBM Informix Error Messages	This HTML file in the IBM Informix Online Documentation site includes causes and solutions for numbered error messages that you might receive from IBM Informix products. Use the UNIX **finderr** utility or the Windows **Informix Error Messages** utility to locate the latest information on error messages.
IBM Informix GLS User's Guide	This manual describes the Global Language Support (GLS), which allows IBM Informix client products and database servers to handle different languages, cultural conventions, and code sets.
IBM Informix Guide to SQL: Reference	This manual describes the Informix system catalog tables, data types, environment variables, the **stores_demo** and **superstores_demo** databases. It also contains a glossary.
IBM Informix Guide to SQL: Syntax	This manual contains the complete syntax descriptions of all Informix (SQL) and Stored Procedure Language (SPL) statements, and functions.
IBM Informix Guide to SQL: Tutorial	This tutorial provides instructions for using SQL to query and modify data in a relational database. It discusses how to embed SQL in programs, create and use stored-procedure language (SPL) routines, create and use triggers, and use casts for extended data types.
IBM Informix High-Performance Loader User's Guide	This guide describes how to use the High-Performance Loader (HPL) to efficiently load and unload large quantities of data to or from an Informix database.
IBM Informix Migration Guide	This manual describes the tasks that you perform when you move data from one location to another and when you migrate existing databases to various Informix database servers. It discusses such database server utilities as **dbexport**, **dbimport**, **dbload**, **dbschema**, **onload**, **onunload**, and **onmode -b**.
IBM Informix Optical Subsystem Guide	This guide describes how to use the Optical Subsystem, a utility that supports the storage of TEXT and BYTE data on optical disks.
IBM Informix R-Tree Index User's Guide	This guide describes the R-tree secondary access method and how to create an R-tree index on user-defined types.
IBM Informix SNMP Subagent Guide	This manual describes the subagent that allows a Simple Network Management Protocol (SNMP) network manager to monitor the status of Informix database servers. It includes a glossary of terms used in the guide.
IBM Informix Storage Manager Administrator's Guide	This guide describes IBM Informix Storage Manager (ISM). ISM receives backup and restore requests from ON–Bar and directs the data to and from storage volumes that are mounted on storage devices.

Table 6-1. Database Server Manuals (continued)

Book Title	Description
IBM Informix Trusted Facility Guide	This guide describes the secure auditing facility and includes information on how to set up and administer audit trails, and extract and interpret audit records.
IBM Informix User-Defined Routines and Data Types Developer's Guide	This guide explains how to define new data types and create UDRs on Dynamic Server. It describes the tasks that you must perform to extend operations on data types, create new casts, extend operator classes for secondary access methods, write opaque data types, and create and register routines.
IBM Informix Virtual-Index Interface Programmer's Guide	This manual explains how to use the Virtual-Index Interface (VII), typically in a DataBlade module, to create a secondary access method. A *virtual* index accesses data from a source outside the database server or specific data inside large objects. The manual describes the syntax, API function calls, and data structures.
IBM Informix Virtual-Table Interface Programmer's Guide	This manual explains how to create a primary access method with the Virtual-Table Interface (VTI). A *virtual* table is dynamically created from a source outside the database server or specific data inside of large objects. The manual describes the syntax, API function calls, and data structures.
J/Foundation Developer's Guide	This guide explains how to use J/Foundation to write user-defined routines (UDRs) in the Java language. It describes the library of classes and interfaces that allow programmers to create and execute Java UDRs that access Dynamic Server.

Client SDK and Connectivity Manuals

Table 6-2 lists the IBM Informix Client SDK and connectivity manuals that you can use when you work with Dynamic Server.

Table 6-2. Client Manuals for Dynamic Server

Book Title	Description
IBM Informix Client Products Installation Guide	This guide contains instructions for installing the IBM Informix Client Software Developer's Kit and IBM Informix Connect on Linux, UNIX, and Windows.
IBM Informix Embedded SQLJ User's Guide	This manual describes how to use IBM Informix Embedded SQLJ to embed SQL statements in Java programs. When you run a SQLJ program, it uses IBM Informix JDBC Driver to connect to the database.
IBM Informix ESQL/C Programmer's Manual	This manual explains how to use ESQL/C, the Informix implementation of embedded SQL for C, to create client applications with database-management capabilities.
IBM Informix JDBC Driver Programmer's Guide	This guide describes how to install, load, and use IBM Informix JDBC Driver to connect to an Informix database from a Java application or applet. You can use IBM Informix JDBC Driver to write user-defined routines.
IBM Informix MaxConnect User's Guide	This manual describes how to install, administer, and tune performance for MaxConnect. MaxConnect enables the database server to support nearly unlimited numbers of client connections and reduces response times and CPU usage.
IBM Informix Object Interface for C++ Programmer's Guide	This guide describes how C++ and DataBlade developers can develop IBM Informix client applications with the C++ programming language.

Table 6-2. Client Manuals for Dynamic Server (continued)

Book Title	Description
IBM Informix Object Translator	This online help describes how to use the IBM Informix Object Translator, a client-side tool, to integrate object-based applications and XML documents into the database.
IBM Informix ODBC Driver Programmer's Manual	This manual explains how to use the IBM Informix ODBC Driver to access Informix databases and the database server. The IBM Informix ODBC Driver is the IBM Informix implementation of the Microsoft Open Database Connectivity (ODBC) interface.
IBM Informix OLE DB Provider Programmer's Guide	This manual explains how to use IBM Informix OLE DB Provider to enable Active Data Objects applications and web pages, for example, to access the database server.
IBM Office Connect	This online help describes how to connect your Microsoft Excel worksheets to databases for online retrieval, modification, and updating of database data.
IBM Informix TP/XA Programmer's Manual	This guide describes how to use the TP/XA library, which facilitates communication between a third-party transaction manager and your database server. TP/XA is supplied with IBM Informix ESQL/C.

DataBlade Manuals

Table 6-3 lists the manuals that you can use when you develop or use DataBlade modules and web-based applications with Dynamic Server.

Table 6-3. DataBlade and Tools Manuals

Book Title	Description
IBM Informix Built-In DataBlade Modules User's Guide	This guide explains how to use the following DataBlade modules that are included with Dynamic Server: • MQ DataBlade module, to allow IBM Informix database applications to communicate with other MQ applications. • Large Object Locator, a foundation DataBlade module that can be used by other modules that create or store large-object data.
IBM Informix DataBlade Developer's Kit User's Guide	This guide describes how to develop and package DataBlade modules using BladeSmith and BladePack.
DataBlade Module Development Overview	This manual provides an overview of DataBlade module development.
IBM Informix DataBlade Module Installation and Registration Guide	This guide explains how to install DataBlade modules and use the BladeManager application to manage DataBlade modules in Informix databases. BladeManager runs on client computers.
Excalibur Text Search DataBlade Module User's Guide	This manual explains how to perform text searches and retrieval using the Excalibur Text Search DataBlade module.
IBM Informix Data Director for Web Tutorial	This tutorial teaches you how to create a small web site using Data Director for Web. You can choose the exercises that teach what you need to learn or copy the entire web site into a database and use it as an example.
IBM Informix Data Director for Web User's Guide	This manual describes how to use the Data Director for Web with the IBM Informix Web DataBlade module to develop and manage web sites. Also, see the tutorial.
IBM Informix Geodetic DataBlade Module User's Guide	This manual explains how to use the IBM Informix Geodetic DataBlade module to store and use spatio-temporal data such as maps.

Table 6-3. DataBlade and Tools Manuals (continued)

Book Title	Description
IBM Informix Image Foundation DataBlade Module User's Guide	This manual explains how the Image Foundation DataBlade module provides a base on which new or specialized image types and image processing technologies can be quickly added or changed.
IBM Informix Spatial DataBlade Module User's Guide	This guide explains how to use the IBM Informix Spatial DataBlade module to store, manipulate, index, and analyze multidimensional spatial data.
IBM Informix TimeSeries DataBlade Module User's Guide	This manual explains how to use the IBM Informix TimeSeries DataBlade module to store and manage time-stamped data such as stock reports.
IBM Informix TimeSeries Real-Time Loader DataBlade Module User's Guide	This manual describes how to use the IBM Informix TimeSeries Real-Time Loader DataBlade module to load time-stamped data and make the data available to queries in real-time.
IBM Informix Video Foundation DataBlade Module User's Guide	This manual describes how to use the IBM Informix Video Foundation DataBlade module to store video technology in a media management system.
IBM Informix Web DataBlade Module Administrator's Guide	This manual describes how to administer web applications that use the IBM Informix Web DataBlade module to dynamically retrieve data from Informix databases.
IBM Informix Web DataBlade Module Application Developer's Guide	This manual explains how to use the IBM Informix Web DataBlade module to develop web applications that dynamically retrieve data from Informix databases.

Appendix A. Database Server Utilities

Dynamic Server includes the following utilities that let you perform administrative tasks and capture information about configuration and performance. These utilities are described in detail in the relevant manual of the Dynamic Server documentation set, as shown in the final column of the table.

Table A-1.

Utility	Description	Where Described
archecker	Verify backups and perform table-level restores	*IBM Informix Backup and Restore Guide*
cdr	Control Enterprise Replication operations.	*IBM Informix Dynamic Server Enterprise Replication Guide*
dbexport	Unload a database into text files for later import into another database and create a schema file.	*IBM Informix Migration Guide*
dbimport	Create and populate a database from text files. Use the schema file with dbimport to re-create the database schema.	*IBM Informix Migration Guide*
dbload	Load data into databases or tables.	*IBM Informix Migration Guide*
dbschema	Create a file that contains the SQL statements needed to replicate a specified table, view, or database, or view the information schema.	*IBM Informix Migration Guide*
imcadmin	Start or stop MaxConnect, or gather statistics on MaxConnect.	*IBM Informix MaxConnect User's Guide*
ISA	Perform various administrative tasks using the IBM Informix Server Administrator (ISA).	ISA online help
ism	Manage IBM Informix Storage Manager, storage devices, and media volumes.	*IBM Informix Storage Manager Administrator's Guide*
onaudit	Manage audit masks and auditing configurations.	*IBM Informix Trusted Facility Guide*
onbar	Back up and restore storage spaces and logical logs.	*IBM Informix Backup and Restore Guide*
oncheck	Check specified disk structures for inconsistencies, repair inconsistent index structures, and display information about disk structures.	*IBM Informix Administrator's Reference*
ondblog	Change the logging mode.	*IBM Informix Administrator's Reference*
oninit	Bring the database server online.	*IBM Informix Administrator's Reference*
onload	Load data that was created with **onunload** into the database server.	*IBM Informix Migration Guide*
onlog	Display the contents of logical-log files.	*IBM Informix Administrator's Reference*
onmode	Change the database server operating mode and perform various other operations on shared memory, sessions, transactions, parameters, and segments.	*IBM Informix Administrator's Reference*
ON–Monitor	Perform administrative tasks using the ON–Monitor menus.	*IBM Informix Administrator's Reference*

© Copyright IBM Corp. 1996, 2005

Table A-1. (continued)

Utility	Description	Where Described
onparams	Modify the configuration of logical logs or physical logs.	*IBM Informix Administrator's Reference*
onperf	Monitor database server performance (create graphs, query trees, show status and metrics).	*IBM Informix Performance Guide*
onpladm	Write scripts and create files that automate data load and unload jobs.	*IBM Informix High-Performance Loader User's Guide*
onshowaudit	Extract information from an audit trail.	*IBM Informix Trusted Facility Guide*
onspaces	Modify dbspaces, blobspaces, sbspaces, or extspaces.	*IBM Informix Administrator's Reference*
onstat	Monitor the operation of the database server.	*IBM Informix Administrator's Reference*
onstat -g	Monitor and debug the database server.	*IBM Informix Administrator's Reference* *IBM Informix Performance Guide*
ontape	Log, back up, and restore data.	*IBM Informix Backup and Restore Guide*
onunload	Unload data from the database server.	*IBM Informix Migration Guide*
Server Setup	Configure the database server, storage spaces, network connectivity, and J/Foundation.	ISA online help
Server Studio Java Edition by AGS	Explore database servers, execute SQL statements and stored procedures (SPLs), and display the results.	Server Studio online help

Appendix B. Accessibility

The syntax diagrams in the HTML version of this manual are available in dotted decimal syntax format, which is an accessible format that is available only if you are using a screen reader.

Dotted Decimal Syntax Diagrams

In dotted decimal format, each syntax element is written on a separate line. If two or more syntax elements are always present together (or always absent together), the elements can appear on the same line, because they can be considered as a single compound syntax element.

Each line starts with a dotted decimal number; for example, 3 or 3.1 or 3.1.1. To hear these numbers correctly, make sure that your screen reader is set to read punctuation. All syntax elements that have the same dotted decimal number (for example, all syntax elements that have the number 3.1) are mutually exclusive alternatives. If you hear the lines 3.1 USERID and 3.1 SYSTEMID, your syntax can include either USERID or SYSTEMID, but not both.

The dotted decimal numbering level denotes the level of nesting. For example, if a syntax element with dotted decimal number 3 is followed by a series of syntax elements with dotted decimal number 3.1, all the syntax elements numbered 3.1 are subordinate to the syntax element numbered 3.

Certain words and symbols are used next to the dotted decimal numbers to add information about the syntax elements. Occasionally, these words and symbols might occur at the beginning of the element itself. For ease of identification, if the word or symbol is a part of the syntax element, the word or symbol is preceded by the backslash (\) character. The * symbol can be used next to a dotted decimal number to indicate that the syntax element repeats. For example, syntax element *FILE with dotted decimal number 3 is read as 3 * FILE. Format 3* FILE indicates that syntax element FILE repeats. Format 3* * FILE indicates that syntax element * FILE repeats.

Characters such as commas, which are used to separate a string of syntax elements, are shown in the syntax just before the items they separate. These characters can appear on the same line as each item, or on a separate line with the same dotted decimal number as the relevant items. The line can also show another symbol that provides information about the syntax elements. For example, the lines 5.1*, 5.1 LASTRUN, and 5.1 DELETE mean that if you use more than one of the LASTRUN and DELETE syntax elements, the elements must be separated by a comma. If no separator is given, assume that you use a blank to separate each syntax element.

If a syntax element is preceded by the % symbol, this identifies a reference that is defined elsewhere. The string following the % symbol is the name of a syntax fragment rather than a literal. For example, the line 2.1 %OP1 means that you should refer to a separate syntax fragment OP1.

The following words and symbols are used next to the dotted decimal numbers:

? Specifies an optional syntax element. A dotted decimal number followed

© Copyright IBM Corp. 1996, 2005

by the ? symbol indicates that all the syntax elements with a corresponding dotted decimal number, and any subordinate syntax elements, are optional. If there is only one syntax element with a dotted decimal number, the ? symbol is displayed on the same line as the syntax element (for example, 5? NOTIFY). If there is more than one syntax element with a dotted decimal number, the ? symbol is displayed on a line by itself, followed by the syntax elements that are optional. For example, if you hear the lines 5 ?, 5 NOTIFY, and 5 UPDATE, you know that syntax elements NOTIFY and UPDATE are optional; that is, you can choose one or none of them. The ? symbol is equivalent to a bypass line in a railroad diagram.

! Specifies a default syntax element. A dotted decimal number followed by the ! symbol and a syntax element indicates that the syntax element is the default option for all syntax elements that share the same dotted decimal number. Only one of the syntax elements that share the same dotted decimal number can specify a ! symbol. For example, if you hear the lines 2? FILE, 2.1! (KEEP), and 2.1 (DELETE), you know that (KEEP) is the default option for the FILE keyword. In this example, if you include the FILE keyword but do not specify an option, default option KEEP is applied. A default option also applies to the next higher dotted decimal number. In this example, if the FILE keyword is omitted, default FILE(KEEP) is used. However, if you hear the lines 2? FILE, 2.1, 2.1.1! (KEEP), and 2.1.1 (DELETE), the default option KEEP only applies to the next higher dotted decimal number, 2.1 (which does not have an associated keyword), and does not apply to 2? FILE. Nothing is used if the keyword FILE is omitted.

***** Specifies a syntax element that can be repeated zero or more times. A dotted decimal number followed by the * symbol indicates that this syntax element can be used zero or more times; that is, it is optional and can be repeated. For example, if you hear the line 5.1* data-area, you know that you can include more than one data area or you can include none. If you hear the lines 3*, 3 HOST, and 3 STATE, you know that you can include HOST, STATE, both together, or nothing.

Notes:

1. If a dotted decimal number has an asterisk (*) next to it and there is only one item with that dotted decimal number, you can repeat that same item more than once.
2. If a dotted decimal number has an asterisk next to it and several items have that dotted decimal number, you can use more than one item from the list, but you cannot use the items more than once each. In the previous example, you could write HOST STATE, but you could not write HOST HOST.
3. The * symbol is equivalent to a loop-back line in a railroad syntax diagram.

+ Specifies a syntax element that must be included one or more times. A dotted decimal number followed by the + symbol indicates that this syntax element must be included one or more times. For example, if you hear the line 6.1+ data-area, you must include at least one data area. If you hear the lines 2+, 2 HOST, and 2 STATE, you know that you must include HOST, STATE, or both. As for the * symbol, you can only repeat a particular item if it is the only item with that dotted decimal number. The + symbol, like the * symbol, is equivalent to a loop-back line in a railroad syntax diagram.

Notices

IBM may not offer the products, services, or features discussed in this document in all countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106-0032, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:
INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created

© Copyright IBM Corp. 1996, 2005

programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
J46A/G4
555 Bailey Avenue
San Jose, CA 95141-1003
U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

All IBM prices shown are IBM's suggested retail prices, are current and are subject to change without notice. Dealer prices may vary.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. You may copy, modify, and distribute these sample programs in any form without payment to IBM for the purposes of developing, using, marketing, or distributing application programs conforming to IBM's application programming interfaces.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

> © (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp. (enter the year or years). All rights reserved.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Trademarks

AIX; DB2; DB2 Universal Database; Distributed Relational Database Architecture; NUMA-Q; OS/2, OS/390, and OS/400; IBM Informix®; C-ISAM®; Foundation.2000™; IBM Informix ® 4GL; IBM Informix®DataBlade®Module; Client SDK™; Cloudscape™; Cloudsync™; IBM Informix®Connect; IBM Informix®Driver for JDBC; Dynamic Connect™; IBM Informix®Dynamic Scalable Architecture™(DSA); IBM Informix®Dynamic Server™; IBM Informix®Enterprise Gateway Manager (Enterprise Gateway Manager); IBM Informix®Extended Parallel Server™; i.Financial Services™; J/Foundation™; MaxConnect™; Object Translator™; Red Brick™; IBM Informix® SE; IBM Informix® SQL; InformiXML™; RedBack®; SystemBuilder™; U2™; UniData®; UniVerse®; wintegrate®are trademarks or registered trademarks of International Business Machines Corporation.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries.

Windows, Windows NT, and Excel are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

UNIX is a registered trademark in the United States and other countries licensed exclusively through X/Open Company Limited.

Other company, product, and service names used in this publication may be trademarks or service marks of others.

Index

A

Access method
  defined 3-19
  primary 3-20, 5-3, 5-5
  secondary 3-19, 5-5
Accessibility xviii
  dotted decimal format of syntax diagrams B-1
  syntax diagrams, reading in a screen reader B-1
Active data objects 1-6
Ad hoc query 5-2
Add-In, DBDK Visual C++ 1-3
Administrative tasks 4-3, 4-6
Administrator
  database server 4-1
  IBM Informix Database 1-7
AFDEBUG environment variable 2-44
AFF_NPROCS configuration parameter 2-33
AFF_SPROC configuration parameter 2-33
Aggregate
  user-defined 3-17
alarmprogram.sh script 2-19
Alarms
  diagnosing problems 4-8
  using 4-5
Allocation, dynamic log 2-41
ALTER TABLE statement
  in-place 2-38
  lock mode 2-40
  logging mode 2-42
  shadow columns 2-38
Altering
  tables 5-1
ANSI joins 2-27, 2-41
ANSI-compliant
  database 3-14
Applets
  Java 5-4
Application developer 4-1
archecker utility 4-8, 6-1, A-1
Architecture
  fault tolerance and high availability 3-6
  high performance 3-5
  memory management 3-5
  parallelization 3-6
Arguments 5-3
Attach fragments 5-1
Auditing
  defined 3-10, 6-3
  onaudit utility A-1
AVOID_EXECUTE directive 2-40

B

B-tree index 3-20, 4-7, 5-1
B-tree scanner 2-21
Backup and restore, external 3-7, 4-5
Backups
  IBM Informix Storage Manager 3-7
  improving performance 4-6
  logical logs 2-42, 4-5
Backups *(continued)*
  ON-Bar utility 3-7, A-1
  ontape utility A-2
  verifying 4-8
Before-image 3-4
BladeManager 1-2, 6-4
BladePack 1-3
BladeSmith 1-3
BLOB data type 2-36, 3-12, 3-15
Blobpage 3-3
Blobspaces
  creating 4-3
  defined 3-3
  onspaces utility A-2
Boldface type x
BOOLEAN data type 3-9, 3-12
Bottleneck, performance 4-7
Bounding box 3-20
Buffer cache 4-7
Buffer manager 2-21
Buffered
  disk space 3-3
  transactions 3-5
Built-in data types
  list 3-12
  replication 3-9
Bundle Installer 2-33
BYTE data type 1-4, 3-15

C

C programs 5-4
C++
  Object Interface 5-4
Cache
  buffer 4-7
  enabling SQL statement 2-43
  entry fully-inserted 2-43
  SQL statements 2-43, 4-7
Callbacks 2-35
Cardinality function 2-34
Case studies 4-7
Casts
  user-defined 5-2
  using 5-3
cdr finderr utility 2-38
cdr utilities A-1
CDR_DBSPACE configuration parameter 2-23
CDR_ENV configuration parameter 2-23
CDR_LOGDELTA environment variable 2-23
CDR_MAX_DYNAMIC_LOGS configuration parameter 2-23
CDR_PERFLOG environment variable 2-23
CDR_QDATA_SBSPACE configuration parameter 2-37
CDR_QHDR_DBSPACE configuration parameter 2-37
CDR_RMSCALEFACT environment variable 2-23
CDR_ROUTER environment variable 2-23
CDR_SERIAL configuration parameter 2-37
CHAR data type 3-12
CHARACTER VARYING data type 3-12
Check constraints 5-2

© Copyright IBM Corp. 1996, 2005

Checkpoint
improving performance 4-7
using 4-4
Chinese GB18030-2000 locale 2-29
Chunks
defined 3-3
number per database 2-19
offset, size limit 2-19
permissions and ownership 4-3
renaming 2-31
reserve pages
location 2-20
size limit 2-19
Client application
defined 3-4
installing 4-2
types 3-13
Client SDK products
defined 1-4
ESQL/J 1-5
IBM Informix GLS 1-5
IBM Informix-ESQL/C 1-4
JDBC Driver 1-5
OLE DB Provider 1-6
TP/XA 1-6
CLIENT_LOCALE 2-44
Client/server
architecture, defined 3-4
setting up connectivity 4-3, 4-4
CLOB data type 2-36, 3-12
Cluster
installation 4-2
Code, sample, conventions for xiv
Collation
changing locale 2-26
Unicode algorithm 2-29
Collection data type 2-36, 3-12, 3-16
cardinality 2-34
replicating 2-22
Columns
assigning data types 5-3
built-in data type 3-20
data types, overview 3-11
retrieving from a database 3-10
storing large objects 3-15
Command-line conventions
how to read xii
sample diagram xii
Commit
and abort callback 2-35
heterogeneous 3-19, 4-4
protocols 4-4
Communications.
See Connectivity.
Complex data type 3-12, 3-16
Compliance
ANSI 3-14
with industry standards xviii
Component, disk 3-3
Composite R-tree index 2-39
Configurable lock mode 2-40
Configuration parameters
AFF_NPROCS 2-33
AFF_SPROC 2-33
CDR_DBSPACE 2-23
CDR_ENV 2-23
CDR_QDATA_SBSPACE 2-37
Configuration parameters *(continued)*
CDR_SERIAL 2-37
DBSERVERALIASES 2-19
DEF_TABLE_LOCKMODE 2-40
DYNAMIC_LOGS 2-41
ENCRYPT_CDR 2-23
ENCRYPT_CIPHERS 2-23
ENCRYPT_MAC 2-23
ENCRYPT_MACFILE 2-23
ENCRYPT_SWITCH 2-23
HPL_DYNAMIC_LIB_PATH 2-31
HPLAPIVERSION 2-31
JDKVERSION 2-44
JVPJAVAHOME 2-44
JVPJAVALIB 2-44
JVPJAVAVM 2-44
LBU_PRESERVE 2-33
list of 2-18, 4-4
LOGSMAX 2-33
LRU_MAX_DIRTY 2-22
LRU_MIN_DIRTY 2-22
LTAPEBLK 2-20
LTXEHWM 2-41
LTXHWM 2-41
NOAGE 2-33
NUMAIOVPS 2-33
NUMCPUVPS 2-33
PLOG_OVERFLOW_PATH 2-20
SBSPACETEMP 2-39
STMT_CACHE_HITS 2-43
STMT_CACHE_NOLIMIT 2-43
STMT_CACHE_NUMPOOL 2-43
STMT_CACHE_SIZE 2-43
TAPEBLK 2-20
VPCLASS 2-33
Configuring
backup and restore 4-3
client/server connections 4-4
database server 4-3
Enterprise Replication 4-3
HDR 4-3
ISM 4-3
locales 4-6
memory 4-2
physical and logical logs 4-3
SQL statement cache 2-43
tasks 4-1, 4-3
Connections
configuring 4-3, 4-4
database server 4-2
database versus network 3-4
defined 3-4
display max number 2-33
mi_lo functions 2-34
multiplexed 3-4
properties 2-44
Connectivity
client/server 3-4
configuring A-2
ODBC standard 1-6
Consistency
using oncheck 4-4, A-1
Constraints 5-2
Contact information xviii
Continuous log backup 3-7
Conventions
command-line xii

Conventions *(continued)*
documentation x
sample-code xiv
syntax diagrams xi
syntax notation xi
typographical x
Cooked disk space 3-3
Coordinating database server 3-19
Coordinating server 3-19
Cost-based optimizer 3-6
CPU
monitoring 4-7
virtual processor 3-4
CRCOLS
adding 2-38
dropping 2-38
CREATE DISTINCT TYPE statement 3-16
CREATE OPAQUE TYPE statement 3-12
CREATE PROCEDURE statement 3-9
CREATE ROLE statement 3-9
CREATE TABLE statement
lock mode 2-40
logging mode 2-42
CREATE VIEW statement 3-9
Cursor
hold with PDQ 2-21

D

Data
distribution 4-7
I/O conversion routines 2-44
models 5-2
storage 3-3
Data Director For Web 6-4
Data mart 1-1, 3-18
Data recovery
data replication 3-8
mirroring 3-8
Data replication
Enterprise Replication 2-22, 2-23, 2-35, 2-38, 3-8, 4-6
fixing problems 4-8
High-Availability Data Replication 3-8, 4-4
pair 3-8
Data types
assigning to columns 5-3
BLOB 2-36, 3-15
built-in 3-12
BYTE 3-15
choosing 5-2
CLOB 2-36, 3-15
collection 2-36
complex 2-36
defining 5-4
distinct 3-16
documentation 6-2
geodetic 2-36
HTML 2-36
LIST 2-34, 2-36
multirepresentational 2-36, 2-38
MULTISET 2-36
opaque 2-36, 2-44, 5-4
row type 3-16
SET 2-34
TEXT 3-15
user defined 3-16
Data warehouse
defined 1-1, 3-18
designing 5-2
Database logging status 3-7, 4-4
Database server aliases, limit 2-19
Database servers
administrator 4-1
allocating logs dynamically 2-41
auditing users 4-5
available data types 3-11
client/server architecture 3-4
configuring 4-3, A-2
distributed databases 3-19
extending 3-15
fault-tolerance 3-6
files used 4-5
high performance of 3-5
initializing 4-2
installing 4-2
local 3-19
migrating 4-3
monitoring 4-4
multiple instances 4-3
operating modes 4-2
operators 4-1
parallel database query 3-6
performance specialist 4-1
remote 3-19
security 3-9
system catalogs 3-10
transaction manager 5-4
users 4-1
verifying consistency 4-4
Windows utilities 2-33
Databases
administrator 4-1
ANSI compliant 3-14
controlling access 5-2, 5-3
data warehousing 3-18
DB-Access, using 1-3
defined 3-3
denormalized 5-1
designing 5-2, 6-1
dimensional 3-18
displaying schema 5-2
distributed 3-19
external 3-19
external data, using 5-2
external remote 3-19
implementing 5-2
loading 5-2
local 3-19
management system
object-relational 3-14
relational 3-14
modifying data 5-3
normalized 5-2
Object Explorer 1-7
object-relational 3-14
supported types 3-14
tasks 5-1, 5-3
users 4-1
DataBlade
data replication 3-9
defining data types 3-13, 5-4
developing applications 5-5
documentation 6-4

IBM Office Connect User's Guide 6-4
IDBA.
See IBM Informix Database Administrator.
Identical statements 2-43
Identifiers 5-3
IFX_DEF_TABLE_LOCKMODE environment variable 2-40
Image Foundation DataBlade module 6-5
IMCADMIN environment variable 2-45
imcadmin utility 2-45, A-1
IMCCONFIG environment variable 2-45
IMCSERVER environment variable 2-45
import function 2-44
importbin function 2-45
Imported restore 4-5
Improving performance 5-1
In-place alters
ADD and DROP CRCOLS 2-38
Index
access methods 3-19
B-tree 3-20
column limit 2-25
fragmenting 3-6, 5-1
functional 3-20
OLTP applications 3-13
optimizer-determined 3-6
R-Tree 3-20
repair A-1
using access methods 4-7
index.htm 1-5
Industry standards, compliance with xviii
Information Schema A-1
INFORMIXDIR environment variable 1-5
INFORMIXDIR/bin directory ix
Inheritance 3-17, 5-2
Initializing database server 4-2
Input function 2-44
Installation Guides xv
Installing
client applications 4-2
client files 2-31
database server 4-2
DataBlades 4-2, 6-4
GLS files 2-31
Informix products 1-2, 2-33
MaxConnect 4-3
no files in /usr/lib directory 2-31
planning for 4-2
serial and key not needed 2-31
tasks 4-1, 4-3
INTEGER data type 3-12
Integration, vendor products 4-2
International Components for Unicode 2-28
Interprocess communications 3-3
INTERVAL data type 3-12
IPC.
See Interprocess communications.
ipload utility 4-5
IPX/SPX
See Network.
ism utility A-1
ISO 8859-1 code set viii
Isolation level 4-7
Iterator, in a FROM clause 2-24
ixpasswd.exe utility 2-33
ixsu.exe utility 2-33

J

J/Foundation
9.21 features 2-44
accessing opaque types 2-44
configuring 4-3, A-2
connection properties 2-44
documentation 6-3
embedding SQL statements 2-44
runtime environment variables 2-44
send and receive functions 2-44
updating names of jar files 2-44
user-defined routines 5-4
using applications 1-5, 5-4
J/Foundation Developer Guide 6-3
JAR files
renaming 2-44
JAR_TEMP_PATH environment variable 2-44
Java Runtime Environment 2-40, 2-44
Java Virtual Machine 2-40, 2-44
JAVA_COMPILER environment variable 2-44
Java.
See J/Foundation.
JavaSoft specifications 1-5
JDBC
defined 1-5
Version 2.0 support 2-44
JDKVERSION parameter 2-44
Join
ANSI 2-27, 2-41
methods 3-6
JRE.
See Java Runtime Environment.
JVM_MAX_HEAP_SIZE environment variable 2-44
JVM.
See Java Virtual Machine.
JVPJAVAHOME configuration parameter 2-44
JVPJAVALIB configuration parameter 2-44
JVPJAVAVM configuration parameter 2-44

K

Key-only cache entry 2-43
Keys
primary and foreign 5-2
Keywords
in syntax diagrams xiii

L

Language, types of 3-17
Large Object Locator DataBlade 6-4
LBU_PRESERVE configuration parameter 2-33
LIBMI applications 5-4
Library
ESQL/C 1-5
GLS feature 1-5, 1-9
ODBC 1-6
Limiting database access 5-2
Linux
installing database server 1-2, 6-2
LIST data type
defined 3-12
obtaining cardinality 2-34
replication not supported 2-36
Load balancing 4-7
LOAD TO statement 2-26

Loading tables
migration 5-2
modifying 5-1
Local database 3-19
Local database server 3-19
Locales
Chinese GB18030-2000 2-29
collation, changing 2-26
data formats 1-5
GLS feature viii, 1-9
setting up 4-6
Lock mode, configurable 2-40
Locking
new tables 2-40
setting lock mode 5-1, 5-3
UDRs 2-44
Log files
list 2-41
Logging mode A-1
Logging.
See Logical log.
Logical log
backup 2-42, 3-7, 4-5
configuring 4-3
defined 3-4
dynamic allocation 2-41
managing 4-4
onlog utility A-1
onparams utility A-2
records 4-5
Logical partition 3-3
Logical units of storage, list of 3-3
LOGSMAX configuration parameter 2-33
Long transaction 2-41, 4-4
Lotus
Domino 1-9
Notes 1-9
LRU queues 4-7
LRU_MAX_DIRTY configuration parameter 2-22
LRU_MIN_DIRTY configuration parameter 2-22
LTAPEBLK configuration parameter 2-20
LTXEHWM configuration parameter 2-41
LTXHWM configuration parameter 2-41
LVARCHAR data type 2-27, 3-9, 3-12

M

Machine notes xvi
Manual
listed 6-1
MaxConnect
defined 1-6, 2-45
documentation 6-3
imcadmin utility 2-45, A-1
installing 4-3
Maximum number of connections 2-33
Media, optical 1-4
Memory
dynamically sharing memory 3-5
managing
buffered transactions 3-5
shared memory 4-4
monitoring 4-7
optimal configuration 4-2
Memory duration 2-34, 2-38
Messages.
See Error messages.
Metadata
partitioning 2-39
Methods
access 3-19
mi_collection_card() function 2-34
mi_dalloc() function 2-34
MI_EVENT_COMMIT_ABORT callback 2-35
MI_EVENT_POST_XACT callback 2-35
MI_EVENT_SAVEPOINT callback 2-35
mi_fparam structure 2-44
mi_get_db_locale() function 2-29
mi_get_transaction_id() function 2-29
mi_lo functions 2-34
mi_realloc() function 2-30
mi_stack_limit() function 2-30
mi_system() function 2-30
mi_transaction_state() function 2-35
Microsoft Open Database Connectivity 1-6
Migration
database server 1-2, 4-3
documentation 6-2
Enterprise Replication 2-35, 2-37
GLS feature 5-2
utilities A-1
Mirroring
defined 3-8
implementing 4-3
performing 4-4
Mode
database server 4-2
MONEY data type 3-12
Monitoring
database server 4-4
locks 5-1
MaxConnect 2-45
SQL statement cache 2-43
system and queries 4-7
transactions 2-21
Moving data 4-3, 4-5
MTS/XA 2-21, 2-45
Multibyte character string 1-5
Multiple
OUT parameter 2-27
residency 4-2
Multiplexing connections 1-6, 3-4
Multirepresentational data 2-36, 2-38
MULTISET data type
defined 3-12
obtaining cardinality 2-34
replication not supported 2-36
Multiuser environment 5-3

N

Named
pipe connection 3-5
return values 2-24
row data type 3-12, 3-16
NCHAR data type 3-12
Nearest-neighbor
query 2-39
Network
capacity, planning 4-2
protocols 2-45
SNMP 1-4, 4-5
New features 4-2
New Technology File System 3-3

NEWCODESET connection property 2-44
NEWLOCALE connection property 2-44
NFS directory 3-5
NOAGE configuration parameter 2-33
Nonlogging tables 2-42
Normalized database 5-2
ntchname utility 2-33
NTFS 3-3
Null values 2-34
NUMAIOVPS configuration parameter 2-33
NUMCPUVPS configuration parameter 2-33
NUMERIC data type 3-12
NVARCHAR data type 3-12

O

Object Explorer 1-7
Object Interface for C++ 1-5, 5-4
Object Translator 6-4
Object-relational database 3-14, 5-2
Objects, data 5-2
ODBC driver 1-6, 2-21, 5-4
Office Connect 6-4
OLE DB provider 1-6
OLTP applications 3-13, 4-7
ON-Bar utility A-1
  -b -l command 2-42
  configuring 4-3
  defined 3-7
  documentation 6-1
  renaming chunks during restore 2-31
  return codes 4-8
  using full capacity of storage media 2-31
ON-Monitor utility 4-4
onaudit utility 4-5, A-1
oncheck utility
  defined A-1
  printing chunk pages 2-29
  verifying consistency 4-4
ONCONFIG file
  setting parameters 4-3
onconfig.std file 2-33
ondblog utility 4-4, A-1
oninit utility 4-4, A-1
Online help xvii
Online manuals xvii
Online notes xv, xvi
Online transaction processing 1-1, 4-7
onload utility 4-3, A-1
onlog utility 4-4, A-1
onmode utility
  -b option 4-3
  -I option 4-8
  -W options 2-43
  -Y option 2-29
  described A-1
onparams utility 2-41, 4-4, A-2
onperf utility 4-7, A-2
onpladm utility 2-42, 4-5, A-2
onpload utility 4-5
onshowaudit utility 4-5, A-2
onsocimc protocol 1-6
onspaces utility 4-4, A-2
onstat utility
  -g dss UDR option 2-38
  -g dss UDRx option 2-38
  -g env option 2-29
onstat utility *(continued)*
  -g grp UDRx option 2-38
  -g imc option 2-45
  -g mem option 2-34
  -g ses option 2-29
  -g sql option 2-29
  -g ssc option 2-43
  -g stm option 2-40
  -x option 2-21
  defined A-2
  diagnosing problems 4-6
  usage 4-4
ontape utility
  configuring 4-3
  documentation 6-1
  listed A-2
  renaming chunks during restore 2-31
  using full capacity of storage media 2-31
ontliimc protocol 1-6
onunload utility 4-3, A-2
Opaque data types
  creating 5-4
  defined 3-12
  support for replicating 2-36
Operating modes 4-2
Operating system
  configure 4-2
  raw and cooked disk space 3-3
  UNIX files 3-3
Operational data store 3-18
Operator class, extending 5-4
Operator, backup 4-1
Optical platter 1-4
Optical Subsystem
  defined 1-4
  documentation 6-2
  using 4-5, 5-3
Optimizer
  cost-based 3-6
Optimizer directives
  AVOID_EXECUTE 2-40
  using 4-7
Optimizing
  performance 6-2
Outer join 4-7
Output function 2-44
Ownership, resolving 4-3

P

Page 3-3
Parallel database queries 2-21, 3-6, 4-7
Parallel processing
  defined 3-6
  Enterprise Replication 2-36
PDQ
  See Parallel database query.
PER_STATEMENT memory duration 2-34
PER_STMT_EXEC memory duration 2-34
PER_STMT_PREP memory duration 2-34
Performance
  B-tree scanner 2-21
  backup and restore 4-6
  buffer manager 2-21
  denormalizing data 5-1
  fragmentation 3-6
  High-Performance Loader 1-3

Performance *(continued)*
HPL 1-3
LRU settings 2-22
memory management 3-2, 3-5
monitoring 4-6
new improvements 2-36
onperf utility A-2
optimizing 6-2
parallelization 3-6
PDQ with hold cursors 2-21
queries 4-7, 5-4
spatial query costing 2-21
specialist 4-1
SPL routines 3-10
statistics 4-7
tuning mechanisms 3-5
Permissions
resolving 4-3
Physical drive 3-3
Physical log
configuring 4-3
defined 3-4
fast recovery 3-8
managing 4-4
onparams utility A-2
overflow 2-20
Physical units of storage, list of 3-3
Planning
database design 5-2
database server installation 4-2
tasks 4-1, 4-3
Platter, optical 1-4
PLOG_OVERFLOW_PATH configuration parameter 2-20
Point-in-time restore 3-7
Populating databases 5-2
Post-transaction callback 2-35
Primary access method 3-20, 5-3, 5-5
Primary key
UDT columns 2-36
using in tables 5-2
Printed manuals xvii
Priority
management for buffers 2-21
Private installation 4-2
Privileges 2-41, 5-2
Processes
compared to threads 3-4
Processing ALTER statements 2-38
Programmers 4-1
Programming tasks 5-3, 5-5
Properties
Inspector 1-7
Protocol
See Network.
Protocol for multiplexing connections 1-6
Proxy server 5-4
PRP.sessionid.threadid pool 2-34

Q

Qualifying statements 2-43
Query
ad hoc 5-2
defined 3-10
filtering 5-1
improving performance 4-7, 5-4
language, structured 3-10
Query *(continued)*
monitoring 4-7
nearest neighbor 2-39
optimizer 3-6
parallel database 3-6
system catalog 3-10
Query plans
displaying without executing the query 2-40
optimizing 4-7

R

R-tree index
access method 3-20
documentation 6-2
nearest-neighbor query 2-39
query costing 2-21
using 5-1, 5-4
Rational
ProjectConsole 1-9
Rapid Developer 1-9
RequisitePro 1-9
Suite AnalystStudio 1-9
Unified Process 1-9
XDET 1-9
Raw disk space 3-3
Raw table 2-42, 4-6
RDBMS.
See Relational Database Management System.
Read-only scroll cursors 2-44
REAL data type 3-12
Records, logical log 4-5
Recovery.
See Restore.
Referential constraint 5-2
Registering DataBlades 6-4
Relational database
application types 3-13
components of 3-10
data types 3-11
DB-Access 1-3
designing 5-2
system catalog 3-10
Relational Database Management System 3-14
Release Notes xv
Remote database
access 2-38
server 3-19
RENAME INDEX statement 2-42
Renaming
jar files 2-44
Replicate
groups 2-37
sets 2-37
Replicating
changed columns 2-37
smart large objects 2-36
user-defined types 2-36
Repository, defined 3-18
Reserved area
sbspaces 2-39
Reserved SQL keywords 2-18, 5-3
Residency, multiple 4-2
Restore
archecker utility A-1
defined 3-6, 3-7
external 4-5

Restore *(continued)*
imported 4-5
improving performance 4-6
ON-Bar utility 3-7, A-1
ontape utility A-2
point-in-time 3-7
Return codes
ON-Bar utility 4-8
Reverting database server 4-3
REVOKE statement 2-41, 3-9
Revoking privileges 2-41, 3-9
Roll back
preventing system hangs 2-41
Root dbspace 3-8
Round-robin fragmentation 3-6, 5-1
Routine identifier 2-44
ROW data types
defined 3-12
named 3-16
replicating 2-22
unnamed 3-16

S

Salvaging logical-log files 3-7
Sample-code conventions xiv
Save point callback 2-35
Save points 2-35
Sbpage 3-3
sbspaces
creating 4-3
defined 3-3, 3-15
managing 4-4
onspaces utility A-2
reserved space 2-39
temporary 2-39, 3-15
SBSPACETEMP configuration parameter 2-39
Scaleup 3-2
Schema 4-5, 5-2
Screen reader
reading syntax diagrams B-1
Scroll cursors 2-44
Searching text
Excalibur Text Search 6-4
Secondary-access methods 3-19, 4-7, 5-5
Security
authorized users 4-5
database server 3-9
encrypting transmissions 2-18
stored procedures 3-9
Segment 5-3
SELECT statements 2-24, 2-27, 5-3
send function 2-44
Sequence objects 2-27
SERIAL or SERIAL8 data type 3-12
Server Setup 4-3, A-2
Server Studio Java Edition 2.30 by AGS 1-6, A-2
Server.
See Database server.
SET data type 2-34, 3-12
SET EXPLAIN statement
ANSI joins 2-42
AVOID_EXECUTE keyword 2-40
SET residency statement not needed 2-26
Shadow columns 2-38
Shared memory
connection 3-5
Shared memory *(continued)*
interprocess communication 3-3
managing 3-5, 4-4
performance 3-2
Silent installation 4-2
Simple large objects
defined 3-15
optical storage 1-4
replication 3-9
using Optical Subsystem 4-5, 5-3
Simple Network Management Protocol 1-4, 6-2
Single-byte character string 1-5
SMALLFLOAT data type 3-12
SMALLINT data type 3-12
Smart large objects
APIs 3-15
copying data 2-38
defined 3-15
I/O properties 3-15
optical storage 1-4
replicating 2-35
sbspaces 3-15
spooling replicated data 2-37
temporary 2-39, 3-15
SmartDisk, not supported 2-31
SMI tables 4-4
SMI.
See System-monitoring interface.
SMP.
See Symmetric multiprocessing.
SNMP subagent 4-5, 6-2
Software dependencies viii
Software upgrades 4-2
Spatial database 2-39
Spatial DataBlade 2-36
Spatial DataBlade module 6-5
Spatial query costing 2-21
Speedup 3-2
SPL routines
creating UDRs 5-3
using 5-2, 5-3
Spooling replicated data 2-37
sqexplain.out file 2-40
SQL code xiv
SQL Editor 1-7
SQL reserved words 2-28
SQL statements
ALTER TABLE 2-40, 2-42
cache 2-42, 4-7
client applications 3-4
composing 5-3
CREATE OPAQUE TYPE 3-12
CREATE PROCEDURE 3-9
CREATE ROLE 3-9
CREATE SEQUENCE 2-27
CREATE TABLE 2-40, 2-42
CREATE VIEW 3-9
database server security 3-9
DELETE 2-41
DESCRIBE INPUT 2-25
DESCRIBE OUTPUT 2-25
displaying memory used 2-40
documentation 6-2
embedded SQL 5-3
GRANT 3-9
INSTEAD OF triggers on views 2-25
invoking UDRs 5-4

SQL statements *(continued)*
LOAD TO 2-26
memory durations 2-34
ORDER BY 2-25
RENAME INDEX 2-42
reserved keywords 2-18
REVOKE 2-41, 3-9
SELECT 5-3
SET COLLATION 2-26
SET EXPLAIN ON AVOID_EXECUTE 2-40
SET residency 2-26
UNLOAD TO 2-26
UPDATE STATISTICS 4-7
sqlhosts file or registry 3-4
SQLJ, defined 1-5
Standard
installation 4-2
table 4-6
Statement cache, SQL 2-42
Statement Local Variables, multiple 2-24
Statistics 4-7
STMT_CACHE_HITS configuration parameter 2-43
STMT_CACHE_NOLIMIT configuration parameter 2-43
STMT_CACHE_NUMPOOL configuration parameter 2-43
STMT_CACHE_SIZE configuration parameter 2-43
Storage manager, third-party 4-3
Storage media
configure 4-3
planning 4-2
using full capacity of 2-20
Storage spaces
backing up 3-7
managing 3-3, 4-4
using Server Setup A-2
Storage volumes 4-3
Stored procedure and security 3-9
stores_demo database viii, 5-2
Storing on optical disk 4-5
Stream-pipe connection 3-5
streamread() support function 2-36
streamwrite() support function 2-36
Striping, disks 4-2
Subquery, writing 4-7
superstores_demo database ix, 5-2
Symmetric multiprocessing 3-2
Synonym names in DB-Access 2-42
Syntax diagrams
conventions for xi
keywords in xiii
reading in a screen reader B-1
variables in xiv
Syntax segment xiii
sysmaster database 4-4, 5-1
sysprocedures table 2-44
syssscstat table 2-43
sysstmtcache table 2-43
System catalog tables
defined 3-10
documentation 6-2
querying 4-6
sysprocedures 2-44
tables, list of 2-18
using 5-2
System failure, and ON-Bar 3-7
System requirements
database viii
System-monitoring interface tables
list of 2-18
syssscstat 2-43
sysstmtcache 2-43
using 4-4

T

Table
access methods 3-19
altering schema 4-5
defined 3-3
demonstration databases 5-2
extents 4-4, 4-7
fragmenting 3-6, 5-1
inheritance 5-2
locking 2-40
modifying 5-1
nonlogging 2-42
placing on disk 5-1
RAW 4-6
security level 3-9
STANDARD 4-6
system catalog 3-10
TEMP 4-6
Table editor 1-7
Table fragmentation.
See Fragmentation.
Tape
using the full capacity 2-20
Tape block size 2-20
TAPEBLK configuration parameter 2-20
Task-documentation matrix 4-1
Tasks
administrative 4-3, 4-6
application programming 5-3, 5-5
database-related 5-1, 5-3
performance 4-6
plan, install, and configure 4-1, 4-3
troubleshooting 4-7, 4-8
Tblspace 3-4
TCP/IP 3-5
Temp table 4-6
Temporary
dbspaces 4-4
sbspaces 2-39, 3-15
smart large object 2-39, 3-15
table, fragmenting 5-1
Terms, defined 4-2, 5-2
TEXT data type 1-4, 3-15
Text search
Excalibur Text Search 6-4
Third-party storage managers 4-3
Threads 3-4
TIME data type 3-9
Time-stamped data 6-5
Timeseries DataBlade 6-5
Tivoli
Storage Manager 1-8
Tivoli Monitoring for Databases 1-8
TME 10 NetView 1-8
TLI connection 3-5
TOC Notes xv
TP/XA library 1-6, 5-4
Transaction state, DataBlades 2-35
Transactions
callbacks 2-35

Transactions *(continued)*
committing and rolling back 2-35
defined 3-5
distributed 3-19
long 2-41, 4-4
manager 2-21, 5-4
Triggers 2-25, 5-3
Troubleshooting 4-7, 4-8
Truncating tables 5-1
Two-phase commit 3-19, 4-4
Type of tables 4-6
Typographical conventions x

U

UDT.
See User-defined type.
Unbuffered disk space 3-3
Unicode code points 2-28
Union, in subqueries 2-27
UNIX
Bundle Installer 2-33
files 3-3
installing database server 1-2, 6-2
UNLOAD TO statement 2-26
Unnamed row type 3-12, 3-16
UPDATE STATISTICS statement 4-7
update_jars.sql script 2-44
User-defined
access methods 3-17
aggregate 3-17
cast, using 5-2, 5-3
functions 3-16
procedure 3-17
User-defined routines
aggregates 3-17
defined 3-16
documentation 6-3
Java routines 1-5, 5-4
memory durations 2-34
multiple OUT parameters 2-27
naming returned values 2-24
obtaining information about 2-43
registering 5-4
SPL 5-3
User-defined type
defined 3-16
primary key column 2-36
remote database access 2-38
replication 2-35, 3-9
WHERE clause, column reference 2-36
Users, types of vii, 4-1
USETABLENAME environment variable 2-28
Utilities
archecker A-1
auditing 6-3
cdr A-1
DB-Access 1-3, 5-2, 6-1
dbexport 4-3, A-1
dbimport 4-3, A-1
dbload 4-3, A-1
dbschema 4-3, 5-2, A-1
ipload 4-5
ISA 1-4, A-1
ism A-1
ixpasswd 2-33
ixsu 2-33
Utilities *(continued)*
migration 6-2
ntchname 2-33
ON-Monitor utility 4-4
onaudit 4-5, A-1
onbar A-1
oncheck 4-4, A-1
ondblog utility 4-4, A-1
oninit 4-4, A-1
onload 4-3, A-1
onlog 4-4, A-1
onmode
-b option 4-3
-I option 4-8
defined A-1
onparams 2-41, 4-4, A-2
onperf 4-7, A-2
onpladm 4-5, A-2
onpload 4-5
onshowaudit 4-5, A-2
onspaces 4-4, A-2
onstat -g 4-6
onstat utility 4-4, A-2
ontape utility A-2
onunload 4-3, A-2

V

VARCHAR data type 3-12
Variable-length opaque type 2-44
Variables, in syntax diagrams xiv
Varying character data type 3-9
Verifying backups 4-8
Video Foundation DataBlade 6-5
Views 2-25, 5-2
VII 5-5
Virtual processors
controlling 2-43
defined 3-4
dynamic scalable architecture 3-2
managing 4-4
monitoring 4-7
shared-memory component 3-2
user-defined 3-17
Virtual-index interface 5-5
Virtual-table interface 5-5
Visual disabilities
reading syntax diagrams B-1
Volume, storage 4-3
VPCLASS configuration parameter 2-33
VTI 5-5

W

Web DataBlade module 6-5
Web pages, accessing with
OLE DB 1-6
Web DataBlade 6-5
WebSphere
Application Server 1-7
MQ 1-8
Portal 1-7
Studio tools 1-8
WHERE clause, UDT column reference 2-36
Windows
installing database server 1-2, 6-2

Words, reserved 5-3
WORF 1-8
WORM optical media 1-4, 4-5, 5-3

X

X/Open
environment 5-4
XML documents 5-4, 6-4

Printed in USA

G251-2281-02