Informix Product Family
Informix
Version 11.50

IBM Informix Getting Started Guide

GI13-1803-02

Informix Product Family
Informix
Version 11.50

IBM Informix Getting Started Guide

GI13-1803-02

Note

Before using this information and the product it supports, read the information in “Notices” on page C-1.

This edition replaces GI13-1803-01.

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.

Contents

Introduction

This introduction provides an overview of the information in this publication and describes the conventions it uses.

About This Publication

Use this publication to get started with IBM® Informix®, Version 11.50. This publication describes the products bundled with Informix, an overview of major features in Informix, and the documentation for Informix. It also summarizes the basic tasks for using the database server and provides a quick reference to command-line utilities.

This section discusses the organization of the publication and the intended audience.

Types of Users

This publication is written for all Informix users:
- Database server administrators
- Database administrators
- Performance engineers
- Database users
- Programmers in the following categories
  - Application developers
  - DataBlade® module developers
  - Authors of user-defined routines
- Technical support

This publication is written with the assumption that you have the following background:
- A working knowledge of your computer, your operating system, and the utilities that your operating system provides
- Some experience working with relational databases or exposure to database concepts
- Some experience with computer programming
- Some experience with database server administration, operating- system administration, or network administration

If you have limited experience with relational databases, SQL, or your operating system, refer to Chapter 6, “Using the Documentation,” on page 6-1, for a list of supplementary titles.

Software Dependencies

This publication is written with the assumption that you are using Informix, Version 11.50, as your database server. Check the release notes for specific version compatibility.

Assumptions About Your Locale

IBM Informix products can support many languages, cultures, and code sets. All the information related to character set, collation, and representation of numeric data, currency, date, and time is brought together in a single environment, called a Global Language Support (GLS) locale.

The examples in this publication are written with the assumption that you are using the default locale, **en_us.8859-1**. This locale supports U.S. English format conventions for date, time, and currency. In addition, this locale supports the ISO 8859-1 code set, which includes the ASCII code set plus many 8-bit characters such as such as é, è, and ñ" using symbols for "é," "è," and "ñ".

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

For instructions on how to specify a nondefault locale, additional syntax, and other considerations related to GLS locales, see the *IBM Informix GLS User's Guide*.

Demonstration Database

The DB-Access utility, which is provided with your IBM Informix database server products, includes one or more of the following demonstration databases:

- The **stores_demo** database illustrates a relational schema with information about a fictitious wholesale sporting-goods distributor. Many examples in IBM Informix publications are based on the **stores_demo** database.
- The **superstores_demo** database illustrates an object-relational schema. The **superstores_demo** database contains examples of extended data types, type and table inheritance, and user-defined routines.

For information about how to create and populate the demonstration databases, see the *IBM Informix DB-Access User's Guide*. For descriptions of the databases and their contents, see the *IBM Informix Guide to SQL: Reference*.

The scripts that you use to install the demonstration databases reside in the **$INFORMIXDIR/bin** directory on UNIX and in the **%INFORMIXDIR%\bin** directory on Windows.

What's New for Version 11.50

This publication includes information about new features and changes in existing functionality.

For more information about the Informix product family, go to http://www.ibm.com/software/data/informix/.

For a comprehensive list of new features in IBM Informix, Version 11.50, see Chapter 2, "Using New Features in Informix," on page 2-1.

Example code conventions

Examples of SQL code occur throughout this publication. 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 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. If you are using DB–Access, you must delimit multiple statements with semicolons.

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 documentation for your product.

Additional documentation

Documentation about this release of IBM Informix products is available in various formats.

You can access or install the product documentation from the Quick Start CD that is shipped with Informix products. To get the most current information, see the Informix information centers at ibm.com®. You can access the information centers and other Informix technical information such as technotes, white papers, and IBM Redbooks® publications online at http://www.ibm.com/software/data/sw-library/.

Compliance with industry standards

IBM Informix products are compliant with various standards.

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.

The IBM Informix Geodetic DataBlade Module supports a subset of the data types from the *Spatial Data Transfer Standard (SDTS)—Federal Information Processing Standard 173*, as referenced by the document *Content Standard for Geospatial Metadata*, Federal Geographic Data Committee, June 8, 1994 (FGDC Metadata Standard).

IBM Informix Dynamic Server (IDS) Enterprise Edition, Version 11.50 is certified under the Common Criteria. For more information, see *Common Criteria*

Certification: Requirements for IBM Informix Dynamic Server, which is available at http://www.ibm.com/e-business/linkweb/publications/servlet/pbi.wss?CTY=US&FNC=SRX&PBL=SC23-7690-00.

Chapter 1. Introducing Informix and Client Products

This chapter provides an overview of IBM Informix, Version 11.50, IBM Informix Client Software Development Kit, and related products. For a list of the publications and description of each product, see "The IBM Informix Documentation Set" on page 6-1.

New editions and product names

IBM Informix Dynamic Server editions were withdrawn and new Informix editions are available. Some products were also renamed. The publications in the Informix library pertain to the following products:

- IBM Informix database server, formerly known as IBM Informix Dynamic Server (IDS)
- IBM OpenAdmin Tool (OAT) for Informix, formerly known as OpenAdmin Tool for Informix Dynamic Server (IDS)
- IBM Informix SQL Warehousing Tool, formerly known as Informix Warehouse Feature

For more information about the Informix product family, go to http://www.ibm.com/software/data/informix/.

IBM Informix

A *database server* is a software package that manages access to one or more databases for one or more client applications. Informix is a fast and scalable database server that manages traditional relational, object-relational, and web-based databases. Informix supports alphanumeric and rich data, such as graphics, multimedia, geospatial, HTML, and user-defined types. You can use Informix on UNIX, Linux, Mac OS X, 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.

IBM Informix Edition Comparison

IBM Informix is available in different editions to fit different business needs. For details about the differences between editions, see the following Web site: http://www.ibm.com/software/data/informix/ids/ids-ed-choice/.

Installation and Migration

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

If you migrate to Informix, Version 11.50, 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 Informix:

- BladeManager
- DataBlade API
- IBM Data Studio
- IBM Informix Client Software Development Kit
- IBM Informix Connect
- IBM Informix DataBlade Developers Kit (DBDK)
- IBM Informix JDBC Driver
- IBM Data Server Driver for JDBC and SQLJ for Informix
- IBM Data Server Provider for .NET
- IBM Informix Spatial DataBlade Module
- IBM Informix Web DataBlade Module
- International Language Supplement
- IBM Informix Warehouse
- Rational® Application Developer for WebSphere® Software

BladeManager

Use the BladeManager to register new DataBlade modules in IBM 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 Informix. 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, Informix 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*.

IBM Informix Client SDK Products

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

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

IBM Informix ESQL/C

Informix 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:

- Informix ESQL/C libraries of C functions for accessing the database server
- Informix 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 Informix 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 IBM Informix database.

For more information, see the *IBM Informix Embedded SQLJ User's Guide* and "IBM Informix JDBC Driver" on page 1-4.

IBM Informix GLS

The Global Language Support (GLS) feature lets the database server handle different languages, cultural conventions, and code sets using different locales. GLS allows you to create databases that use the diacritics, collating sequence, and monetary and time conventions of the language that you select. A GLS locale is an environment that has defined conventions for a particular language or culture. See "Assumptions About Your Locale" on page x 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.

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 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 IBM 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. Use TP/XA for distributed transaction processing in a multivendor database setting.

IBM Informix DataBlade Developers Kit (DBDK)

The Informix DataBlade Developers 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 *IBM Informix DataBlade Module Development Overview* and *IBM DataBlade Developers Kit User's Guide*.

IBM Informix JDBC Driver

IBM Informix JDBC Driver lets Java programmers access IBM Informix databases from within Java applications or applets. Programmers can create client applications that use JDBC to connect to Informix, 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 IBM Informix database server data types.

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

IBM Data Server Driver for JDBC and SQLJ

The IBM Data Server Driver for JDBC and SQLJ, provides support for client applications written in Java using JDBC.

The driver, which is based on the Distributed Relational Database Architecture™ (DRDA®) protocol, is also integrated with DB2®. This capability means that the common features of the IBM Data Server Driver for JDBC and SQLJ allow you to write client applications that can use both DB2 and Informix data servers.

IBM Informix supports only the type 4 version of the IBM Data Server Driver for JDBC and SQLJ, which means that sessions connect directly to the database or data server, without a middle layer.

The IBM Data Server Driver for JDBC and SQLJ is compliant with the Sun Microsystems JDBC 3.0 and JDBC 4.0 specifications.

For more information, see *IBM Data Server Driver for JDBC and SQLJ for Informix*.

IBM Data Server Provider for .NET (Windows)

The IBM Data Server Provider for .NET extends data server support for the ADO.NET interface. The provider delivers high-performing, secure access to IBM data servers.

For more information, see the *IBM Data Server Provider for .NET Programmer's Guide, Informix Edition*.

IBM Informix Spatial DataBlade Module

The IBM Informix Spatial DataBlade Module embeds a geographic information system (GIS) into your IBM Informix kernel. The IBM Informix Spatial DataBlade Module implements the Open GIS Consortium, Inc. (OpenGIS, or OGC) SQL3 specification of abstract data types (ADTs). These data types can store spatial data such as the location of a landmark, a street, or a parcel of land. TheIBM Informix Spatial DataBlade Module also conforms to the OpenGIS Simple Features Specification for SQL Revision 1.1.

The Spatial DataBlade module comes with a sample data disk that contains worldwide location-based data that can be visualized and manipulated using the free IBM Informix Spatial DataBlade Module bundle. This bundle can be ordered or downloaded free from http://www.ibm.com/software/data/informix/blades/spatial/.

IBM Informix Web DataBlade Module

Use the IBM Informix Web DataBlade Module to create Web applications that incorporate data retrieved dynamically from the Informix database.

Using the Web DataBlade module, you need not develop a CGI application to dynamically access database data. Instead, you create HTML pages that include Web DataBlade module tags and functions that dynamically execute the SQL statements you specify and format the results. These pages are called Application Pages (AppPages). The types of data you retrieve can include traditional data types, as well as HTML, image, audio, and video data.

For more information, see the *IBM Informix Web DataBlade Module User's Guide* and the *IBM Informix Web DataBlade Module Application Developer's Guide*.

International Language Supplement

IBM Informix products include a core set of GLS locale files, including the default locale and most locales to support English, Western European, Eastern European, Asian, and African territories. If you do not find a locale to support your language and territory, you can get additional locales in the International Language Supplement (ILS) product. The ILS provides all available GLS locales and code-set conversion files. It also includes error messages to support several languages.

IBM Informix Warehouse

IBM Informix Warehouse is a warehouse infrastructure platform with IBM Informix and the IBM Informix Warehouse Feature. The Informix Warehouse Feature offers an integrated tool to help extract, load, and transform operational data into an Informix data warehouse. You can use Informix Warehouse to build a data warehousing solution that includes a highly scalable relational database, data access capabilities, and front-end analysis tools.

For more information, see http://www.ibm.com/informix/warehouse. For information on installing and using the IBM Informix Warehouse Feature, see http://publib.boulder.ibm.com/infocenter/idshelp/v115/topic/com.ibm.dwe.navigate.doc/welcome_ifxwarehouse.html.

Rational Application Developer for WebSphere Software

Rational Application Developer is part of the Rational Software Development Platform series of products. The products are all built on Eclipse, which is an open-source platform for creating application development tools. IBM Rational Application Developer for WebSphere Software extends Eclipse with visual construction development. It helps Java developers rapidly design, develop, assemble, test, profile and deploy Java/J2EE, Portal, Web, Web services and SOA applications. You can learn more about this product on the Web at http://www.ibm.com/software/awdtools/developer/application/index.html.

Related IBM Informix Products

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

IBM Informix Server Administrator (ISA)

IBM Informix Server Administrator (ISA) is not included with Informix. ISA is available for download from http://www.ibm.com/software/data/informix/downloads.html.

OpenAdmin Tool for Informix

A PHP-based Web browser administration tool, the OpenAdmin Tool for Informix, provides the ability to administer multiple database server instances from single location. Some tasks you can perform with OpenAdmin include: gathering and analyzing performance statistics, monitoring system health, defining and managing automated tasks through the SQL administration API, creating and displaying performance histograms for analysis and tuning, and monitoring high availability solutions that include HDR, shared disk secondary servers, and remote standalone secondary servers. You can easily plug in your own extensions to OpenAdmin to create the functionality you need.

OpenAdmin is an open-source program that you can download from http://www.openadmintool.com.

IBM Informix MaxConnect (UNIX)

IBM Informix MaxConnect is a networking product for IBM Informix database servers on UNIX. Two protocols for multiplexing connections, **ontliimc** and **onsocimc**, are available for Informix MaxConnect users. Informix 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. Informix MaxConnect increases system scalability to many thousands of connections and saves system resources, reducing response times and CPU requirements. You can install Informix 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*.

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 IBM Informix Web DataBlade Module

DataBlade Modules

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

- IBM Informix Image Foundation DataBlade Module
- IBM Informix Excalibur Text Search DataBlade
- IBM Informix Geodetic DataBlade Module
- IBM Informix TimeSeries DataBlade Module
- Informix Video Foundation DataBlade Module
- IBM Informix Web DataBlade Module

For a brief description of each of these, see "DataBlade Publications" on page 6-4.

IBM Data Studio

IBM Data Studio is an integrated data-management environment. Data architects, developers, and database administrators can use IBM Data Studio to collaborate throughout the data-driven application development lifecycle. Learn how IBM Data Studio can help throughout the data lifecycle at: http://www.ibm.com/software/data/studio/

The following information center introduces IBM Data Studio and the underlying workbench technology, and explains how to use its integrated development environment to achieve your database development goals: http://publib.boulder.ibm.com/infocenter/dstudio/v1r1m0/index.jsp

Client Products for Informix

For some client APIs, you can choose between two drivers that have different communication protocols:

- IBM Informix proprietary protocol

  Informix Client Software Development Kit and Informix JDBC Driver

  IBM Informix clients can connect only to IBM Informix. They support the full range of Informix functionality and are compatible with all earlier, supported versions of Informix.
- Distributed Relational Database Architecture (DRDA)

  IBM Data Server drivers

  Data Server clients can connect to both Informix and DB2 database servers. Data Server clients have some Informix-specific functionality restrictions, but they typically provide a larger range of functionality than IBM Informix clients. Data Server clients support Informix Version 11.10 and later.

JDBC Drivers

You can use either of the following JDBC drivers with IBM Informix. They are both included with Informix on most platforms:

- IBM Informix JDBC Driver
- IBM Data Server Driver for JDBC and SQLJ

  Also available on the IBM Informix download site: https://www14.software.ibm.com/webapp/iwm/web/preLogin.do?lang=en_US&source=swg-informixfpd

.NET Providers

You can use either of the following .NET Providers with IBM Informix. They are both Included with Informix:

- IBM Informix .NET Provider

  The Informix .NET Provider is provided with Client SDK and supports existing .NET applications.
- IBM Data Server .NET Provider for Informix

For information about the differences between the .NET providers, see the *IBM Informix Client Products Installation Guide*.

Both client programs are also available at the IBM Informix download site: https://www14.software.ibm.com/webapp/iwm/web/preLogin.do?lang=en_US&source=swg-informixfpd

PHP Drivers

You can use either of the following open source PHP drivers with IBM Informix:

- PHP Driver for IBM Informix

Available from the PHP Web site: http://pecl.php.net/package/PDO_INFORMIX. The PHP Driver forIBM Informix requires Client SDK to be installed on the same computer.

- PHP Driver for Data Server clients

  Available from the PHP Web site: http://pecl.php.net/package/PDO_IBM. The PHP Driver for Data Server clients requires the IBM Data Server Driver for ODBC and CLI to be installed on the same computer. The IBM Data Server Driver for ODBC and CLI is available on the IBM Informix download site: https://www14.software.ibm.com/webapp/iwm/web/preLogin.do?lang=en_US&source=swg-informixfpd.

Ruby on Rails Adapters

You can use either of the following open source Ruby on Rails adapters with IBM Informix:

- Ruby on Rails Adapter for IBM Informix

  Available from the Ruby Web site: http://rubyforge.org/projects/ruby-informix as a Ruby gem. The Ruby on Rails adapter for IBM Informix requires Client SDK to be installed on the same computer.

- Ruby on Rails Adapter for Data Server clients

  Available from the Ruby Web site: http://rubyforge.org/projects/rubyibm as a Ruby gem. The Ruby on Rails adapter for Data Server clients requires the IBM Data Server Driver for ODBC and CLI to be installed on the same computer. The IBM Data Server Driver for ODBC and CLI is available on the IBM Informix download site: https://www14.software.ibm.com/webapp/iwm/web/preLogin.do?lang=en_US&source=swg-informixfpd.

Other Informix Drivers

The following drivers are only available with Client SDK:

- IBM Informix ESQL/C
- IBM Informix GLS
- IBM Informix Object Interface for C++
- IBM Informix ODBC Driver
- IBM Informix OLE DB Provider

Other Related IBM Products

You can use many IBM products with IBM Informix. The Informix Interoperability with Other Products Web site shows which versions of other products are certified to work with specific versions of Informix. As new releases are certified, the Web site is updated: http://www.ibm.com/software/data/informix/ids/interop/.

Chapter 2. Using New Features in Informix

This chapter describes the new features in IBM Informix, Version 11.50, 11.10, 10.00, 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 11.50.xC9 of IBM Informix

IBM Informix, Version 11.50.xC9 contains new functionality in the following areas:

- Administration
  - "Reserving memory for critical activities"
  - "Configuring the server response to low memory"
  - "Handle high severity event alarms" on page 2-2
  - "Reserving space for BYTE and TEXT data in round-robin fragments" on page 2-2
  - "Enhancements to the OpenAdmin Tool (OAT) for Informix" on page 2-2
  - "IBM Informix TimeSeries Plug-in for OpenAdmin Tool (OAT)" on page 2-3
- Application Development
  - "Comparing date and interval values " on page 2-2
- Enterprise Replication
  - "Easier setup of faster consistency checking for Enterprise Replication " on page 2-2
- Security
  - "Support for later versions of IBM Global Security Kit (GSKit)" on page 2-2

Reserving memory for critical activities

You can enable the server to reserve a specific amount of memory for use when critical activities (such as rollback activities) are needed and the server has limited free memory. If you enable the new LOW_MEMORY_RESERVE configuration parameter by setting it to a specified value in kilobytes, the critical activities can complete even when you get out-of-memory errors. You can also dynamically adjust the value of the LOW_MEMORY_RESERVE configuration parameter with the **onmode -wm** or **-wf** command.

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

Configuring the server response to low memory

You can configure the actions that the server takes to continue processing when memory is critically low. You can specify the criteria for terminating sessions based on idle time, memory usage, and other factors so that the targeted application can continue and avoid out-of-memory problems. Configuring the low memory response is useful for embedded applications that have memory limitations.

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

Support for later versions of IBM Global Security Kit (GSKit)

Informix 11.50 now supports GSKit version 7 or later, but ships with GSKit version 7. If you have a later version of GSKit software installed on your computer, you can set the GSKIT_VERSION configuration parameter so that the database server uses the later version.

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

Handle high severity event alarms

You can plan responses to severity 4 and 5 event alarms based on their explanations and user actions.

This feature is documented in the *IBM Informix Administrator's Reference*.

Easier setup of faster consistency checking for Enterprise Replication

When you increase the speed of consistency checking by creating an index on the ifx_replcheck shadow column, you no longer need to include the conflict resolution shadow columns in the replicated table. In the CREATE TABLE statement, the WITH REPLCHECK keywords do not require the WITH CRCOLS keywords.

This feature is documented in the *IBM Informix Enterprise Replication Guide*.

Comparing date and interval values

You can compare the values of two DATETIME data types or the values of two INTERVAL data types to determine if the first value is before, after, or the same as the second value.

This feature is documented in the *IBM Informix DataBlade API Function Reference*.

Reserving space for BYTE and TEXT data in round-robin fragments

If you have tables that contain BYTE or TEXT columns and that are fragmented by the round-robin distribution scheme, you can reserve space for inserting BYTE and TEXT data by setting the PN_STAGEBLOB_THRESHOLD configuration parameter.

This feature is documented in the *IBM Informix Administrator's Reference*.

Enhancements to the OpenAdmin Tool (OAT) for Informix

The IBM OpenAdmin Tool (OAT) for Informix supports IBM Informix 11.50 and 11.70. OAT 2.73 has the following enhancements that support Informix 11.50 database servers.

- Create historical system reports. You can create system reports based on the historical data for a database server on the **Performance Analysis** > **System Reports** > **Historical Data** page. The reports provide information about the SQL statements that were run on the database server, including the slowest SQL statements and the SQL statements with the most I/O time and with the most buffer activity.
- View performance history for multiple fields and servers. You can view historical data on profiled fields, including automatic checkpoints, disk and buffer activity, memory, virtual processors, and the B-tree scanner on the

Performance Analysis > **Performance History** page. You can view data on specific fields over time and compare performance on multiple servers. Previously, you could view data for a single field on one server at a time.
- View information about locks. You can analyze information about locks and monitor locks by using the lock reports on the **Performance Analysis** > **System Reports** page. The reports provide information about the locks on the database server, including the owner and waiter, and the number of requests, waits, and timeouts.
- View memory usage. You can view information about server memory usage on the **Server Administration** > **Memory Manager** page.
- Specify the minimum OAT version for a plug-in. When you develop a plug-in for OAT, you can specify a minimum required version of OAT.
- Uninstall plug-ins. You can uninstall plug-ins for OAT on the **Plug-in Manager** page. In previous versions, you could disable plug-ins, but you could not uninstall them.
- Restore OAT menu items. On the **Menu Manager** page, you can restore OAT menu items that are deleted. In previous versions, you could not restore deleted menu items.

See the OAT help for more information.

IBM Informix TimeSeries Plug-in for OpenAdmin Tool (OAT)

The TimeSeries plug-in provides a graphical interface for reviewing and administering the **TimeSeries** data type provided by the IBM Informix TimeSeries DataBlade Module. A time series is a set of data recorded as it varies over time.

With the TimeSeries plug-in, you can monitor the database objects related to your time series:
- View the **TimeSeries** subtypes, containers, and calendars that are used for the time series data in a database.
- View the tables and indexes that contain **TimeSeries** subtypes.
- View the virtual tables that are based on tables that contain **TimeSeries** subtypes.
- Monitor the percentage of the space that is in the dbspaces for the containers.

You can also create and drop containers, calendars, and virtual tables.

See the TimeSeries plug-in help for more information.

New Features in Version 11.50.xC8 of IBM Informix

IBM Informix, Version 11.50.xC8 contains new functionality in the following areas:
- Administration
  - "Setting the file seek position for large files" on page 2-4
  - "SNMP support for Windows 64-bit software" on page 2-4
- High-availability
  - "Locking tables from updatable secondary servers in a cluster" on page 2-4
  - "Configuring log flow control for RS secondary servers" on page 2-4

Setting the file seek position for large files

Use the **mi_file_seek8()** function to set the position for the next read or write operation on an open file that is larger than 2 GB. You can use the **mi_file_tell8()** function to return the current position, relative to the beginning of the file, in an operating-system file that is larger than 2 GB.

This feature is documented in the *IBM Informix DataBlade API Function Reference*.

SNMP support for Windows 64-bit software

You can use the Simple Network Management Protocol (SNMP) for a 64-bit Informix server that is running on a Windows 64-bit operating system. However, you cannot use that protocol if you are running a 32-bit Informix server on a Windows 64-bit operating system.

Locking tables from updatable secondary servers in a cluster

You can set exclusive locks and shared locks from updatable secondary servers in a cluster.

If you set exclusive locks from an updatable secondary server, sessions can read the table but not update it. When one session has an exclusive lock on a given table, no other session can obtain a shared or exclusive lock on that table.

If you set shared locks from an updatable secondary server, sessions can read the table but cannot modify it until the lock is released.

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

Configuring log flow control for RS secondary servers

You can limit log activity on the primary server so that remote standalone (RS) secondary servers in the cluster can catch up. This configuration can improve performance over congested or intermittent networks. You use the RSS_FLOW_CONTROL configuration parameter to set thresholds that start and stop flow control.

This feature is documented in the *IBM Informix Administrator's Reference*.

New Features in Version 11.50.xC6 of IBM Informix

IBM Informix, Version 11.50.xC6 contains new functionality in the following areas:

- Administration
  - "Connection Manager Proxy Support" on page 2-5
  - "External Backups on RS Secondary Servers" on page 2-5
  - "Dynamically Start, Stop, or Restart Listen Threads" on page 2-6
  - "Improved Transaction Information " on page 2-6
  - "SQL Administration API Portal: Arguments by Functional Categories" on page 2-6
  - "View Event Alarms in the Scheduler" on page 2-6
  - "Enhancements to the OpenAdmin Tool for Informix" on page 2-6
  - "Enhancements to the Schema Manager plug-in for the OpenAdmin Tool for Informix" on page 2-6

- Application Development
  - “An ALTER FRAGMENT Operation Can Now Force Out Transactions to Get Exclusive Access to Tables” on page 2-7
  - “RETAINUPDATELOCKS Environment Option” on page 2-7
  - “Enhancements to Merging Information into a Target Table with the MERGE Statement” on page 2-7
  - “New Column Size Field Format for CDC Records” on page 2-7
- Enterprise Replication
  - “Enable or Disable Enterprise Replication Event Alarms” on page 2-8
  - “Enhancements to the Enterprise Replication plug-in for the OpenAdmin Tool for Informix” on page 2-8
- High-availability
  - “XA Transaction Support for Updatable Secondary Servers in a High-availability Cluster” on page 2-9
- Installation
  - “Simplified Deployment of an Informix Instance (Windows)” on page 2-9
  - “Dynamic dbspace Relocation when Deploying Snapshots (Windows)” on page 2-9
  - “Deploying Instances with the Deployment Utility ” on page 2-9
  - “Installing Informix by Using an RPM package (Linux)” on page 2-9
- Migration
  - “Migrating or Upgrading High-availability Clusters” on page 2-10
  - “Upgrading to a New Server or Fix pack ” on page 2-10
  - “Quickly Reverting to Your Source Server After a Failed Upgrade ” on page 2-10
- Performance
  - “Light Scans on Tables ” on page 2-10
  - “Process Multiple Basic Text Search Queries Simultaneously” on page 2-11
- Warehousing
  - “Load and Unload Data with External Tables” on page 2-11

Connection Manager Proxy Support

Connection Manager can now be configured as a proxy server when clients connect to Informix data servers from outside a firewall. You can use proxy names when you configure high-availability cluster connections using the Connection Manager.

This information is documented in the *IBM Informix Administrator's Reference* and the *IBM Informix Administrator's Guide*.

External Backups on RS Secondary Servers

To reduce the processing load on the primary server in a high-availability cluster you can create an external backup on an RS secondary server instead of on the primary server. The RS secondary server must not contain any non-logged objects. You can restore the backup on any server in the cluster.

This feature is documented in the *IBM Informix Administrator's Reference*, the *IBM Informix Backup and Restore Guide*, and the *IBM Informix Administrator's Guide*.

Dynamically Start, Stop, or Restart Listen Threads

You can now dynamically start, stop, or restart (stop and start) an existing listen thread for a SOCTCP or TLITCP network protocol without interrupting existing connections. Use the new **onmode -P** commands or the new SQL administration API **admin()** or **task()** functions.

This information is documented in the *IBM Informix Administrator's Reference* and the *IBM Informix Administrator's Guide*.

Improved Transaction Information

The **onstat -x** utility output now provides an estimated rollback time for transactions that are rolling back.

The **onstat -x** utility is documented in the *IBM Informix Administrator's Reference*.

SQL Administration API Portal: Arguments by Functional Categories

You can view **admin()** and **task()** function arguments, listed by category, and follow the links to information about the arguments.

This information is documented in the *IBM Informix Administrator's Reference*.

View Event Alarms in the Scheduler

You can see event alarms in the **ph_alerts** table in the **sysadmin** database. You can query the **ph_alert** table on a local or remote server to view the recent event alarms for that server.

This feature is documented in the *IBM Informix Administrator's Reference*.

Enhancements to the OpenAdmin Tool for Informix

The OpenAdmin Tool (OAT) Version 2.27 has the following enhancements:

- The OAT user interface is available in multiple languages: Czech, Brazilian Portuguese, English, German, Hungarian, Italian, Japanese, Korean, Polish, Russian, Simplified Chinese, Slovak, Spanish, and Traditional Chinese. Earlier versions were available only in English. Note that OAT supports viewing and modifying data only in databases with the U.S. English locale. For example, if you access a database with an Italian or a Japanese locale, the data might not appear accurately.
- On the High Availability Clusters page (MACH), you can configure the Connection Manager as a proxy server when clients connect to Informix data servers from outside a firewall.
- You can display performance and status information for a database server by using the **onstat** utility in OAT.

See the OpenAdmin Tool help for more information.

Enhancements to the Schema Manager plug-in for the OpenAdmin Tool for Informix

The Schema Manager plug-in Version 1.2 has the following enhancements:

- You can create tables and define the attributes necessary to make the tables fully functional.

- You can drop or truncate tables.
- You can load and unload data from a source that is external to the database server by using external tables. You can create an external table that has the same schema as the database table, or use a previously defined external table.

See the Schema Manager plug-in for OAT help, for more information.

An ALTER FRAGMENT Operation Can Now Force Out Transactions to Get Exclusive Access to Tables

The ALTER FRAGMENT operation requires exclusive access and exclusive locks on all the tables involved in the operation. Now a DBA can force out other transactions that opened or locked the tables involved in an ALTER FRAGMENT ON TABLE operation. When the FORCE_DDL_EXEC environment option is enabled, the ALTER FRAGMENT ON TABLE operation waits to get exclusive access to the table, as specified by the lock mode wait value. If necessary, the server rolls back the transactions that have access or locks on the tables involved in the operation until the timeout value of the FORCE_DDL_EXEC environment option is reached.

The feature is documented in the *IBM Informix Guide to SQL: Syntax*.

RETAINUPDATELOCKS Environment Option

The RETAINUPDATELOCKS environment option can improve concurrency in Dynamic SQL applications that include the SELECT . . . FOR UPDATE statement. When your session uses the Committed Read, Dirty Read, or Cursor Stability isolation levels, you can set this option to instruct the database server to postpone releasing update locks until a transaction ends. Retaining update locks prevents users in other sessions from locking rows that you have not yet updated. By default the setting persists for the current session unless you reset it or you change the isolation level to one that does not support the RETAINUPDATELOCKS environment option.

The feature is documented in the *IBM Informix Guide to SQL: Syntax*.

New Column Size Field Format for CDC Records

The CDC_REC_DELETE, CDC_REC_INSERT, CDC_REC_UPDAFT, and CDC_REC_UPDBEF records have fields listing the size of each variable-length column in the row, if any. The column size fields in these CDC records are now 4 bytes long and in network byte, big-endian, order. Previously, the column size fields were the size of the operating system platform's integer byte length, in native byte order. To use the new column size field format, when you run the **cdc_opensess()** function, specify 1 as the value for both the *major_version* and *minor_version* arguments.

This feature is documented in the *IBM Informix Change Data Capture Programmer's Guide*.

Enhancements to Merging Information into a Target Table with the MERGE Statement

You can remove rows from a target table when you are merging information into it from a source table by using the Delete clause in the MERGE statement. You can combine the Delete clause and the Insert clause to add new rows to the target table if they do not match the join condition, and delete the rows that do match the join condition.

You can use an external table, created by the CREATE EXTERNAL TABLE statement, as a source table in the MERGE statement. You can use a table protected by LBAC as a source or target table in a MERGE statement.

You can include savepoints in a transaction to preserve the effects of the MERGE statement after a partial rollback.

You can specify a more selective join predicate because you no longer need to include an equality condition.

The MERGE statement is documented in the *IBM Informix Guide to SQL: Syntax* and the *IBM Informix Guide to SQL: Tutorial*.

Enable or Disable Enterprise Replication Event Alarms

You can control which Enterprise Replication event alarms are enabled by setting the CDR_ALARMS environment variable with the CDR_ENV configuration parameter. By default, most Enterprise Replication event alarms are enabled. The state change events 49, and 51 through 70, which indicate that a **cdr** command was run, are disabled by default.

This feature is documented in the *IBM Informix Enterprise Replication Guide*.

Enhancements to the Enterprise Replication plug-in for the OpenAdmin Tool for Informix

The Enterprise Replication (ER) plug-in Version 2.4 has the following enhancements:

- You can improve the performance of the consistency check and synchronization tasks for replicates and replicate sets. You can limit the amount of data to check to the rows updated since a specified time. For replicates, you can also specify a subset of a table to check with a WHERE clause. For replicate sets, you can process the replicates in a replicate set in parallel.
- You can delete information about completed check and synchronization tasks from the OAT display.
- You can delete ER objects including servers, replicates, replicate sets, and templates.
- You can instantiate an ER template on multiple servers at the same time. In previous releases, a template was instantiated on one server at a time.
- You can change the master replicate server for a replicate. In previous releases, the master replicate server could not be changed.
- The wizard for checking and synchronizing replicate sets and replicates is updated to improve usability.

See the Enterprise Replication plug-in for OAT help, for more information.

XA Transaction Support for Updatable Secondary Servers in a High-availability Cluster

Client applications that connect to updatable secondaries in a high-availability cluster can use XA transactions. Previously, only client applications connected to primary servers could use XA transactions.

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

Simplified Deployment of an Informix Instance (Windows)

You can set configuration parameters, essential environment variables, and SQLHOST connectivity information in a deployment utility configuration file (**ifxdeploy.conf**). You can reuse the file to deploy instances with the deployment utility. When you plan to embed a snapshot of the same instance in multiple locations, you can use the customized configuration file to centralize the installation setup work and ensure that the deployed instances are consistent.

If you do not specify certain environment variables when you run the **ifxdeploy** command, or set them in an **ifxdeploy.conf** file, the deployment utility sets values based on the process environment.

The feature is documented in the *IBM Informix Installation Guide for Windows*.

Dynamic dbspace Relocation when Deploying Snapshots (Windows)

You can dynamically reconfigure the chunk paths of a snapshot by using the **-relocate** option of the deployment utility. The **-relocate** option initializes the dbspaces required for deploying the template instance without a separate step for disk space initialization.

The feature is documented in the *IBM Informix Installation Guide for Windows*.

Deploying Instances with the Deployment Utility

You can use the deployment utility (**ifxdeploy**) to rapidly deploy a configured Informix instance to multiple computers, avoiding the need to manually configure instances on each computer and wait for each instance to initialize. By setting configuration parameters, essential environment variables, and SQLHOST connectivity information in a deployment utility configuration file (**ifxdeploy.conf**) template, you can reuse the configuration file to deploy the instance multiple times. If you are deploying an instance with existing data, you can use the **-relocate** option to store the data in a different location from where it was originally stored. You can call the utility programmatically or from a script as part of an application installation that embeds Informix. The utility automatically extracts tar files that are compressed with Gzip.

This feature is documented in the *IBM Informix Installation Guide for UNIX, Linux, and Mac OS X*.

Installing Informix by Using an RPM package (Linux)

You can install IBM Informix products by using RPM Package Manager (RPM) on some Linux operating systems (see the Machine Notes for the list of operating systems). RPM simplifies installation by installing the product from a single package, and tracks the packages that are installed on a system. By default, this method installs the database server in the `/opt/IBM/informix` directory.

This information is documented in the *IBM Informix Installation Guide for UNIX, Linux, and Mac OS X*.

Migrating or Upgrading High-availability Clusters

Information was added to the Migration Guide to help you coordinate the migration of all servers in a high-availability cluster. If you use high-availability clusters, refer to this information when you upgrade to a new PID or fix pack, migrate to a new version of Informix, or revert to the previous version of the database server if necessary. High-availability clusters must be rebuilt after migration.

Upgrading to a New Server or Fix pack

To upgrade to a new Informix version or fix pack, follow the procedures that are described in the *IBM Informix Migration Guide*. Likewise, to revert to the prior version of the server after successfully upgrading, follow the reversion procedures that are described in the *IBM Informix Migration Guide*.

Quickly Reverting to Your Source Server After a Failed Upgrade

If an upgrade to a new Informix version or fix pack fails, use the new **onrestorept** utility to restore the server to a consistent, pre-upgrade state. You can undo changes made during the upgrade in minutes (and in some cases, in seconds). Previously, if a fix pack upgrade failed, you had to restore the database by using a level-0 archive.

The new CONVERSION_GUARD and RESTORE_POINT_DIR configuration parameters specify information that the **onrestorept** utility can use if an upgrade fails.

The **onrestorept** utility is documented in the *IBM Informix Migration Guide*. and the new configuration parameters are documented in the *IBM Informix Administrator's Reference*.

Light Scans on Tables

You can now enable Informix to perform light scans on compressed tables, tables with rows that are larger than a page, and tables with any data, including VARCHAR, LVARCHAR, and NVARCHAR data. The server does not scan pieces of a row (such as smart large objects) that are stored outside of the row.

To enable light scans, set the BATCHEDREAD_TABLE configuration parameter to 1. You can also use the IFX_BATCHEDREAD_TABLE environment option of the SET ENVIRONMENT statement to change the value of the BATCHEDREAD_TABLE configuration parameter for a session.

To display scan information, use the new **onstat -g scn** command. This command supersedes the **onstat -g lsc** command.

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

Process Multiple Basic Text Search Queries Simultaneously

If basic text search queries are slow because multiple users are running queries at the same time, you can add more BTS virtual processors so that queries run simultaneously, each in its own virtual processor. Previously, you could only create one BTS virtual processor and queries ran serially.

This feature is documented in the *IBM Informix Database Extensions User's Guide*.

Load and Unload Data with External Tables

Informix supports external tables. You can read and write from a source that is external to the database server. External tables provide an SQL interface to data in text files managed by the operating system or to data from a FIFO device. To create external tables, use the CREATE EXTERNAL TABLE statement. Use the existing DROP TABLE statement to drop an external table.

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

New Features in Version 11.50.xC5 of IBM Informix

IBM Informix, Version 11.50.xC5 contains new functionality in the following areas:

- High Availability
  - "Configuring RS Secondary Server Latency for Disaster Recovery" on page 2-12
- Administration
  - "Deploying Informix with the Deployment Utility (Windows)" on page 2-12
  - "Forcing the Database Server to Shut Down" on page 2-12
  - "Enhanced Support for Multibyte Character Strings" on page 2-12
  - "Upgrading to Informix v11.50.xC5 in Silent Mode (Windows)" on page 2-12
  - "New default for the INFORMIXTERM environment variable for terminal set up on UNIX" on page 2-13
  - "onconfig Portal: Configuration Parameters Listed by Functional Categories" on page 2-13
  - "onstat Portal: onstat Commands Listed by Functional Categories" on page 2-13
  - "Enhancements to the Enterprise Replication plug-in for the OAT for Informix" on page 2-13
- Application Development
  - "Retrieving Data by Using Hierarchical Queries" on page 2-15
- Enterprise Replication
  - "Enterprise Replication Stops if Memory Allocation Fails" on page 2-13
  - "Improving the Performance of Checking or Synchronizing Replicate Sets (Windows)" on page 2-13
  - "Notification of an Incorrect Log Position When Enterprise Replication Restarts" on page 2-14
  - "Improving the Performance of Consistency Checking with an Index" on page 2-14
  - "Specifying the Range of Data Sync Threads to Apply Replicated Transactions" on page 2-14

- Warehousing
  - "Loading Data into a Warehouse with the MERGE Statement" on page 2-14

Configuring RS Secondary Server Latency for Disaster Recovery

To aid in disaster recovery scenarios, you can now configure RS secondary servers to wait for a specified period of time before applying logs. Delaying the application of log files allows you to recover quickly from erroneous database modifications by restoring from the RS secondary server.

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

Deploying Informix with the Deployment Utility (Windows)

This command-line utility facilitates faster deployment of a configured Informix instance. This utility is currently available on Windows. The utility can be called programmatically or from a script as part of an application installation, and therefore supports completely silent Informix deployment.

This feature is documented in the *IBM Informix Installation Guide for Windows*.

Forcing the Database Server to Shut Down

Use the onclean utility to force the database server to shut down when normal shut down with the onmode utility fails to shut down the server or when you cannot restart the server. The onclean utility cleans up shared memory, semaphores, and stops database server virtual processes.

Use the onshutdown script to try to shut down the server normally. If the server does not shut down after a specified time, the server is forced to shut down.

This information is documented in the *IBM Informix Administrator's Reference*.

Enhanced Support for Multibyte Character Strings

In database locales that support multibyte code sets, such as **UTF-8**, a single logical character can occupy up to four bytes of storage. To prevent multibyte character strings from being truncated, use the new **SQL_LOGICAL_CHAR** session environment variable to instruct Informix to interpret the declared size in units of logical characters. By default, any explicit or default size specifications are interpreted in units of bytes.

This feature is documented in the *IBM Informix Guide to SQL: Syntax*.

Upgrading to Informix v11.50.xC5 in Silent Mode (Windows)

You can upgrade from Informix v11.50.xC4 or earlier to Informix v11.50.xC5 in silent mode. You will need to record a response file for Informix v11.50.xC5. You cannot use a response file that was recorded when Informix v11.50.xC4 or earlier versions were installed.

See the *IBM Informix Installation Guide for Windows* for more information.

New default for the INFORMIXTERM environment variable for terminal set up on UNIX

The INFORMIXTERM environment variable now defaults to the **terminfo** directory. If you use DB-Access on UNIX, the terminal definitions will be taken from the **terminfo** directory unless you set the INFORMIXTERM environment variable to the **termcap** file.

This information is documented in the *IBM Informix Guide to SQL: Reference*.

onconfig Portal: Configuration Parameters Listed by Functional Categories

You can view configuration parameters listed in functional categories as they appear in the onconfig.std file and follow the links to individual configuration parameter topics.

This information is documented in the *IBM Informix Administrator's Reference*.

onstat Portal: onstat Commands Listed by Functional Categories

You can view onstat commands listed in functional categories and follow the links to individual onstat command topics.

This information is documented in the *IBM Informix Administrator's Reference*.

Enhancements to the Enterprise Replication plug-in for the OAT for Informix

The ER plug-in Version 2.23 has the following enhancements:

- You can modify an ER server by changing the idle time out or the ATS directory or RIS directory.
- You can add or remove participants in an ER replicate, or modify replicate attributes, including the conflict-resolution rule, scope, and replication frequency.
- You can control replication processing on an ER server and delete ER objects, including replicates and replicate sets.

See the OAT help for more information.

Enterprise Replication Stops if Memory Allocation Fails

In previous releases, when data sync threads could not allocate memory, replication continued, but the replicated transactions were aborted and ATS or RIS files were created. Starting in Informix v11.50.xC5, if data sync threads encounter a memory allocation error while replaying replicated transactions, replication stops to prevent data from becoming inconsistent. Event alarm 47 is raised. To resume replication, solve the memory issue and run the **cdr start** command.

This behavior is documented in the *IBM Informix Enterprise Replication Guide*.

Improving the Performance of Checking or Synchronizing Replicate Sets (Windows)

This feature, which was added in Informix v11.50.xC4 for other operating systems, is now available on Windows platforms. You can increase the speed of a consistency check or a synchronization operation on a replicate set by performing the operation on each replicate in parallel. Specify the number of parallel processes to use for processing a replicate set by using the --process option.

This feature is documented in the *IBM Informix Enterprise Replication Guide*.

Notification of an Incorrect Log Position When Enterprise Replication Restarts

If the replay log position has been overrun or is later than the current log position when replication restarts, event alarm 75 is now raised. To resume replication, run the **cdr cleanstart** command and then the **cdr check --repair** command.

This behavior is documented in the *IBM Informix Enterprise Replication Guide*.

Improving the Performance of Consistency Checking with an Index

You can now increase the speed of a consistency check on a replicate by indexing a new shadow column, ifx_replcheck. You add the ifx_replcheck column to your replicated table using the WITH REPLCHECK clause and create a unique index on the ifx_replcheck column and your primary key columns. You can also alter an existing table to add the ifx_replcheck column. The replicated table must also have the CRCOLS shadow columns. You cannot perform a table-level restore on a table that contains the ifx_replcheck column.

This feature is documented in the *IBM Informix Enterprise Replication Guide* and *IBM Informix Guide to SQL: Syntax*.

Specifying the Range of Data Sync Threads to Apply Replicated Transactions

To specify a minimum and maximum number of data sync threads for applying replicated transactions, set the CDR_APPLY configuration parameter. By default, replication allocates a range of one to four data sync threads for each CPU virtual processor.

This configuration parameter is documented in the *IBM Informix Enterprise Replication Guide*.

Loading Data into a Warehouse with the MERGE Statement

Instead of using separate UPDATE and INSERT statements to load data from an OLTP database into a database warehouse environment, use the new MERGE statement, which can combine UPDATE and INSERT operations into a single SQL statement.

The MERGE statement can merge records from a table, view, or query (the source) with the records in a local table (the target). You can specify a logical condition that MERGE applies to a join of the source and target objects.

The MERGE statement supports Update and Insert triggers on the target table. Any constraints on the target table are enforced in MERGE operations.

This feature is documented in the *IBM Informix Guide to SQL: Syntax*.

Retrieving Data by Using Hierarchical Queries

You can now retrieve data from a table by using hierarchical queries, which maintain the relationship between the data.

The SELECT statement of Informix now supports START WITH .. CONNECT BY syntax for recursively querying a table in which a hierarchy of parent-child relationships exist. The syntax can define recursive queries that reflect the topology of the data hierarchy.

- CONNECT_BY_ISLEAF
- CONNECT_BY_ISCYCLE
- CONNECT_BY_ROOT
- LEVEL
- PRIOR
- ORDER SIBLINGS BY
- SYS_CONNECT_BY_PATH

This implementation of hierarchical queries uses extensions to the ISO standard for SQL.

This feature is documented in the *IBM Informix Guide to SQL: Syntax*.

New Features in Version 11.50.xC4 of IBM Informix

IBM Informix, Version 11.50.xC4 contains new functionality in the following areas:

- High Availability
  - "High Availability Enabled on Windows 64-bit Operating Systems" on page 2-16
- Administration
  - "Save Disk Space by Compressing Data" on page 2-16
  - "Improved Options for Specifying Processor Affinity" on page 2-17
  - "Enhanced Startup Script Customization" on page 2-17
  - "Disable IPv6 Support" on page 2-17
  - "Enhancements to the OpenAdmin Tool (OAT) for Informix" on page 2-17
  - "ONINIT_STDOUT environment variable allows capturing oninit output (Windows)" on page 2-18
  - "Generating a Customized Database Server Configuration File" on page 2-18
  - "Upgrade Informix by Overwriting the Existing Installation (Windows)" on page 2-18
  - "Changing how Informix searches for user names (Windows)" on page 2-18
- Enterprise Replication
  - "Generate ATS and RIS Files in XML Format for Custom Repair Scripts" on page 2-19
  - "Prevent Upserts During Conflict Resolution" on page 2-19
  - "New Event Alarms for Enterprise Replication" on page 2-19
  - "Monitor the Progress of a Consistency Check or a Direct Synchronization" on page 2-19
  - "Specify the Duration of a Consistency Check" on page 2-20
  - "Schedule Synchronization or Consistency Checking Operations" on page 2-20

  - "Improve Consistency Checking Performance" on page 2-20
  - "Improve the Performance of Checking or Synchronizing Replicate Sets (UNIX)" on page 2-20
  - "Role separation for Enterprise Replication (UNIX)" on page 2-20
  - "Enhancements to the Enterprise Replication plug-in for the OpenAdmin Tool (OAT) for Informix" on page 2-20
- Performance
  - "Enable Concurrent I/O to Improve Performance on AIX Operating Systems" on page 2-21
  - "IFX_LARGE_PAGES Environment Variable (AIX, Solaris)" on page 2-21
- Application Development
  - "DataBlade Module Registration through SQL" on page 2-21
  - "Control the Results of a Fuzzy Search with the Basic Text Search DataBlade Module" on page 2-21
  - "Map Characters for Indexing with the Basic Text Search DataBlade Module" on page 2-22
  - "Default Boolean Operator in Basic Text Search Queries" on page 2-22
  - "Storage for Temporary Basic Text Search Files" on page 2-22
  - "Track Basic Text Search Query Trends" on page 2-22
  - "Fragment **bts** Indexes by Expressions" on page 2-22
  - "Monitor Change Data Capture Sessions" on page 2-23
  - "View Change Data Capture Error Message Text" on page 2-23
- Security
  - "Automatic Check for Installation Path Security (UNIX, Linux, Mac OS X)" on page 2-23
  - "Utility to Enhance the Security of the Informix Installation Path (UNIX, Linux, Mac OS X)" on page 2-23

High Availability Enabled on Windows 64-bit Operating Systems

You can use high availability features in this version of IBM Informix for Windows 64-bit systems. See "New Features in Version 11.50.xC1 of IBM Informix" on page 2-32 for more information.

Save Disk Space by Compressing Data

You can now use SQL administration API commands to save disk space by compressing row data in a table or in one or more table fragments. You can also use SQL administration API commands to consolidate free space in a table or fragment, return this free space to the dbspace, and estimate the amount of space that is saved by compressing the data.

For information on licensing and pricing, contact your IBM representative.

You can display the following types of information about compression:

- Active compression dictionaries that describe how the data is compressed, with the new **onstat -g ppd** command
- All compression dictionaries, by querying the new **syscompdicts_full** table and **syscompdicts** view in the **sysmaster** database

- Progress of currently running compression operations, with the new **onstat-g dsk** command
- Uncompressed contents of compressed log records, with a new **onlog** utility option
- Percentage of compressed rows, with the **onstat -pT** option

If you prefer, you can use the OAT to manage compression and view information about compression.

If you are upgrading to Version 11.50.xC4 from Version 11.50.xC1, 11.50.xC2, or 11.50.xC3, follow the procedures that are described in the *IBM Informix Migration Guide* to drop and recreate the sysmaster database and to revert to earlier versions, which do not support compression.

The compression commands and procedures for using them are documented in the *IBM Informix Administrator's Guide* and the *IBM Informix Guide to SQL: Syntax*. Information on the **oncheck -pT** option, the **onstat g -ppd** option, the **onstat g -dsk** option, the **syscompdicts_full** table, and the **syscompdicts** view, are in the *IBM Informix Administrator's Reference*. Migration issues are documented in the *IBM Informix Migration Guide*.

Improved Options for Specifying Processor Affinity

When you specify the processor affinity with the VPCLASS configuration parameter, you can now specify individual processors or ranges of processors. When specifying a range of processors, you can also specify an incremental value with the range that indicates which CPUs in the range should be assigned to the virtual processors.

This information is documented in the *IBM Informix Administrator's Guide* and *IBM Informix Administrator's Reference*.

Enhanced Startup Script Customization

You can now use the **oninit -w** command to customize startup scripts and automate startup on Windows operating systems. The **-w** flag forces the command to wait until the server successfully initializes before returning to the command prompt. In a high-availability environment, you can use the **oninit -w** command only on primary servers; it is not valid on secondary servers.

This information is documented in the *IBM Informix Administrator's Reference*.

Disable IPv6 Support

You can now optionally disable Internet Protocol Version 6 (IPv6) support for better performance on certain platforms. You can disable IPv6 support for one or more database instances or client applications, or for all instances and client applications.

This information is documented in the *IBM Informix Administrator's Guide*.

Enhancements to the OpenAdmin Tool (OAT) for Informix

The OpenAdmin Tool (OAT) for Informix Version 2.24 has the following enhancements:

- You can save disk space by compressing tables and table fragments. You can consolidate free space in tables and fragments (repack) and return free space to the dbspace (shrink). You can also estimate the amount of space that you can save if you compress data.
- You can use Query by Example to perform these standard SQL operations on a table: query, insert, update, and delete.
- You can perform a query drill-down of both live and historical data in SQL Explorer.
- You can create an external directive to modify a query's behavior, by using Query Optimizer Workbench, a new feature of SQL Explorer. You can also manage existing directives.
- You can export OAT connection information to an XML file and import it into another instance of OAT. The connection information includes the OAT group name, user name, and password, Informix server information, and OAT environment variables.
- The PDO_INFORMIX driver supplied with OAT is upgraded to version 1.2.6.

See the OAT help for more information.

ONINIT_STDOUT environment variable allows capturing oninit output (Windows)

You can now capture the output of the **oninit** command on Windows systems by setting the new ONINIT_STDOUT environment variable. Capturing the output of the **oninit** command is useful in certain situations such as when using the **-v** (verbose) option or when you want to see output from an unhandled exception in a process launched within a virtual processor.

This feature is documented in the *IBM Informix Administrator's Guide* and the *IBM Informix Guide to SQL: Reference*.

Generating a Customized Database Server Configuration File

Use the new **genoncfg** utility to generate a customized Informix configuration file that is optimized for both anticipated usage and your host environment.

This information is documented in the *IBM Informix Administrator's Reference*.

Upgrade Informix by Overwriting the Existing Installation (Windows)

When you are upgrading Informix from one fixpack level to another in the same product version, you can install the product in the running database server installation directory. You also can overwrite an existing installation for training or specialized technical needs. Take all necessary backup precautions before upgrading with this method. It is recommended that most users follow the IBM Informix Migration Guide for version-to-version upgrades and migrations.

This information is documented in the *IBM Informix Installation Guide for Windows* and *Quick Beginnings for IBM Informix Dynamic Server Express Edition*.

Changing how Informix searches for user names (Windows)

You can now use the CHECKALLDOMAINSFORUSER configuration parameter to configure how the database server searches for user names logging in to a networked Windows environment. You set the CHECKALLDOMAINSFORUSER to determine whether Informix searches for the user name on the local host only or on all domains.

This information is documented in the *IBM Informix Administrator's Guide* and *IBM Informix Administrator's Reference*.

Generate ATS and RIS Files in XML Format for Custom Repair Scripts

You can now choose to create Aborted Transactions Spooling (ATS) and Row Information Spooling (RIS) files in the new XML format, the traditional text format, or both formats. You can use the XML format of ATS and RIS files to create custom repair scripts based on your business logic. Use the **−−atsrisformat** option of the **cdr define server** or **cdr modify server** commands to specify the file format.

This information is documented in the *IBM Informix Enterprise Replication Guide*.

Prevent Upserts During Conflict Resolution

You can use the new *delete wins* conflict resolution rule to perform conflict resolution that compares the time stamps of conflicting rows but prevents upserts. An upsert is an UPDATE operation that is converted to an INSERT operation because the row to update was not found on the target server. An upsert operation can occur if a row is deleted from a target server before an UPDATE operation is processed on that target server or if an UPDATE operation was processed by the target server before the INSERT operation for that row. Depending on your business logic, you might want to prevent deleted rows from being reinserted and UPDATE operations from occurring before INSERT operations.

This information is documented in the *IBM Informix Enterprise Replication Guide*.

New Event Alarms for Enterprise Replication

Event alarms are now generated in the following situations:

- When a replicated transaction fails and an ATS or RIS file is generated.
- When you run an Enterprise Replication command that changes the state of a server, replicate, or replicate set. State changes can include stopping, starting, modifying, deleting, or changing the connection status.
- When a connection between Enterprise Replication servers is dropped. A connection can be terminated by an administration command, such as **cdr disconnect server**, an idle timeout, or a network issue.

This information is documented in the *IBM Informix Enterprise Replication Guide*.

Monitor the Progress of a Consistency Check or a Direct Synchronization

You can now monitor information on the progress of a consistency check or direct synchronization operation, including the estimated duration of the operation. Use the **cdr stats check** command to monitor consistency checks. Use the **cdr stats sync** command to monitor synchronization operations.

This information is documented in the *IBM Informix Enterprise Replication Guide*.

Specify the Duration of a Consistency Check

By default, inconsistent rows are rechecked for up to five seconds, which might not be enough time for replicated transactions to be applied on the target server. You can now specify the number of seconds to spend on rechecking the consistency of inconsistent rows. Rechecking prevents transactions that are in progress from being listed as inconsistent in the consistency report. You can use the **--inprogress** option of the **cdr check replicate** and **cdr check replicateset** commands to specify the maximum number of seconds to perform rechecking.

This information is documented in the *IBM Informix Enterprise Replication Guide*.

Schedule Synchronization or Consistency Checking Operations

You can now use the Scheduler to automate the running of synchronization and consistency check operations by using the **--background** option with the **cdr check** or **cdr sync** commands.

This information is documented in the *IBM Informix Enterprise Replication Guide*.

Improve Consistency Checking Performance

You can now reduce the duration of a consistency check by performing a check in parallel and by controlling the amount of data that is checked. You can specify a time from which to check updated rows by using the **--since** option. You can specify a subset of a table to check by using the **--where** option. You can prevent the checking of large objects by using the **--skipLOB** option.

This information is documented in the *IBM Informix Enterprise Replication Guide*.

Improve the Performance of Checking or Synchronizing Replicate Sets (UNIX)

You can increase the speed of a consistency check or a synchronization operation on a replicate set by performing the operation on each replicate in parallel. Specify the number of parallel processes to use for processing a replicate set by using the **--process** option.

This information is documented in the *IBM Informix Enterprise Replication Guide*.

Role separation for Enterprise Replication (UNIX)

You can now run Enterprise Replication commands as a member of the DBSA group. Previously, only user **informix** could run Enterprise Replication commands. For some commands, the DBSA user needs to be granted additional permissions on certain data or **syscdr** tables.

This information is documented in the *IBM Informix Enterprise Replication Guide*.

Enhancements to the Enterprise Replication plug-in for the OpenAdmin Tool (OAT) for Informix

The Enterprise Replication plug-in for the OpenAdmin Tool (OAT) for Informix has the following enhancements for administering enterprise replication:

- Define an Enterprise Replication server as a new node in an existing ER domain or the first node in a new ER domain.

- Define replicates and replicate sets.
- Start, stop, suspend, and resume replicate sets and replicates.
- Check replicate sets and replicates for data inconsistencies.
- Repair data inconsistencies in replicate sets and replicates.
- Synchronize data to repair inconsistencies in replicate sets and replicates.
- Repair failed transactions by using Aborted Transaction Spooling (ATS) and Row Information Spooling (RIS) files.

See the Enterprise Replication plug-in help in the OpenAdmin Tool (OAT) for Informix for more information.

Enable Concurrent I/O to Improve Performance on AIX Operating Systems

You can now improve the performance of cooked files used for dbspace chunks by setting concurrent I/O, which allows multiple reads and writes to a file at the same time. Concurrent I/O provides performance improvements over direct I/O because it avoids the serialization of noncompeting reads and writes from normal file-system locking. Similar to direct I/O, concurrent IO avoids file system buffering. Concurrent I/O can be especially beneficial when you have a file striped across multiple disks. To enable concurrent I/O on AIX®, set the DIRECT_IO configuration parameter to 2.

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

IFX_LARGE_PAGES Environment Variable (AIX, Solaris)

The **IFX_LARGE_PAGES** environment variable can enable the use of large pages for non-message shared memory segments that are resident in physical memory. The DBSA must set the RESIDENT configuration parameter accordingly, and must use operating system commands to configure a pool of these large pages. Informix can then use large pages from that pool, if they are available, for shared virtual memory segments.

This information is documented in the *IBM Informix Guide to SQL: Reference*.

DataBlade Module Registration through SQL

You can now use the built-in **SYSBldPrepare( )** function to register one or more DataBlade modules or to unregister a DataBlade module, as an alternative to using the BladeManager utility. This enables you to register DataBlade modules from any client API that supports SQL, such as DB-Access, SPL, C API, ESQL, JDBC, or ODBC, and without requiring that BladeManager be installed. Another new function, **SYSBldRelease( )**, returns the version string of the **SYSBldPrepare( )** function.

This feature is documented in the *IBM Informix Guide to SQL: Syntax*.

Control the Results of a Fuzzy Search with the Basic Text Search DataBlade Module

You can now specify the degree of similarity of search results in fuzzy searches when using the Basic Text Search DataBlade module. Specify a number between 0 and 1, where a higher value results in a higher degree of similarity. To limit results, specify a higher number. To maximize results, specify a lower number. The default degree of similarity is 0.5.

This information is documented in the *IBM Informix Database Extensions User's Guide*.

Map Characters for Indexing with the Basic Text Search DataBlade Module

You can now map characters in your data to other characters during indexing with the Basic Text Search DataBlade module. For example, you can specify that letters with diacritical marks are indexed as the same letters without marks. You can also standardize inconsistent prefixes or delete character strings from indexed text. To use character maps, include the **btsmapchars** parameter when you create your **bts** index.

This information is documented in the *IBM Informix Database Extensions User's Guide*.

Default Boolean Operator in Basic Text Search Queries

You can now change the default Boolean operator between search terms in Basic Text Search queries from OR to AND by using the **query_default_operator** parameter when you create a **bts** index. The default operator is represented by a blank space between terms. Many popular end-user search engines use AND as the default operator between search terms, where end-users expect the search results to contain all their search terms.

This information is documented in the *IBM Informix Database Extensions User's Guide*.

Storage for Temporary Basic Text Search Files

You can now specify that temporary files used by the Basic Text Search DataBlade module are stored in a separate sbspace from the one used to store the **bts** index. Separating temporary files from the **bts** index might improve query performance.

This information is documented in the *IBM Informix Database Extensions User's Guide*.

Track Basic Text Search Query Trends

You can now track what queries are run against your **bts** index by including the **query_log** parameter when you create a **bts** index. You can use query trends information to provide hints to end-users on popular queries or work on optimizing the most popular queries.

This information is documented in the *IBM Informix Database Extensions User's Guide*.

Fragment bts Indexes by Expressions

You can now fragment **bts** indexes by expressions into multiple sbspaces instead of a single sbspace.

This information is documented in the *IBM Informix Database Extensions User's Guide*.

Monitor Change Data Capture Sessions

You can now monitor the sessions involved in change data capture by using the new **onstat -g cdc** command. Use the options to display information about the captured tables, the buffers being used by sessions, the configuration of sessions, or the data capture activity.

This information is documented in the Documentation Notes for the *IBM Informix Change Data Capture Programmer's Guide*.

View Change Data Capture Error Message Text

You can now view the error message text corresponding to an error name by using the new cdc_errortext() function.

This information is documented in the Documentation Notes for the *IBM Informix Change Data Capture Programmer's Guide*.

Automatic Check for Installation Path Security (UNIX, Linux, Mac OS X)

The installation program now verifies that all the directories in the installation path are secure. For most environments, installation will proceed without any user interaction with the security check. However, if any directories are not secure, you will be presented with options to address the problem, including one option to automatically reset the directory owner, group, and permissions settings.

Utility to Enhance the Security of the Informix Installation Path (UNIX, Linux, Mac OS X)

The new **onsecurity** utility checks the security of the runtime path specified by the INFORMIXDIR environment variable to prevent an attacker from using privileged programs that run with the database server, such as the **oninit**, **onmode**, and **onstat** utilities. When the **onsecurity** utility detects a potential security vulnerability, it prevents the database server from being initialized and disables other essential operations so that it is not possible for an unauthorized user to tamper with your installation.

This information is documented in the *IBM Informix Security Guide*.

New Features in Version 11.50.xC3 of IBM Informix

IBM Informix, Version 11.50.xC3 contains new functionality in the following areas:

- Administration
  - Performing Unattended Installations on Mac OS X
  - Setting up Data Source Names on Mac OS X
  - Using SQL administration API to Dynamically Update Configuration Parameters
  - Dynamically Updating the LTXEHWM, LTXHWM, and DYNAMIC_LOGS Configuration Parameters
  - Dynamically Updating Enterprise Replication Configuration Parameters in the ONCONFIG File
  - Improved Consistency Reporting after Enterprise Replication Repair Operations

  - Administering and Monitoring Enterprise Replication with SQL administration API
  - Improved SQL Tracing with the SQL administration API
- Performance
  - Changing the Size of the First Extent of a Table
- Application Development
  - Rolling Back SQL Transactions to a Savepoint
  - Server-Specific Audit Configuration File Functionality
  - Capturing Transactional Data with the Change Data Capture API
  - Basic Text Search DataBlade module Supports High-Availability Clusters
  - Querying XML Attributes with the Basic Text DataBlade module
  - Setting the Frequency of Error Checking for Smart Large Object Transmission

Performing Unattended Installations on Mac OS X

Now you can install IBM Informix Client Software Development Kit (Client SDK) and Informix Connect on Mac OS X by setting installation information in the new **bundle.ini** file, which is provided on the installation media as a template, and running an installation command. In earlier releases, to install these products on Mac OS X, you had to start the installation process and provide installation information interactively.

This installation process is documented in the *IBM Informix Installation Guide for UNIX, Linux, and Mac OS X*.

Setting up Data Source Names on Mac OS X

You can now set up IBM Informix ODBC Driver data source names (DSNs) by using the Apple ODBC Administrator on computers running Mac OS X.

This feature is documented in the *IBM Informix ODBC Driver Programmer's Manual*.

Using the SQL Administration API to Dynamically Update Configuration Parameters

You can dynamically set configuration parameters by using the new SQL administration API SET ONCONFIG commands. These new commands provide the same function as the **onmode -wf** or **onmode -wm** commands. Use the SET ONCONFIG or SET ONCONFIG PERMANENT commands to change configuration parameters in the ONCONFIG file. Use the SET ONCONFIG MEMORY command to configure parameters only for the current session.

The SET ONCONFIG commands are described in the *IBM Informix Guide to SQL: Syntax*.

Dynamically Updating the LTXEHWM, LTXHWM, and DYNAMIC_LOGS Configuration Parameters

You can now dynamically update the value of the LTXEHWM, LTXHWM, and DYNAMIC_LOGS configuration parameters by using the **onmode -wf** or **onmode**

-wm command. The **onmode -wf** command changes the value in the ONCONFIG file. The **onmode -wm** command changes the value for the current session.

These onmode commands are documented in the *IBM Informix Administrator's Reference*.

Dynamically Updating Enterprise Replication Configuration Parameters in the ONCONFIG File

Previously, the **cdr add onconfig**, **cdr change onconfig**, and **cdr remove onconfig** commands dynamically updated Enterprise Replication configuration parameters for the current session only. Now these commands create persistent updates to Enterprise Replication configuration parameters in the ONCONFIG file.

This feature is documented in the *IBM Informix Enterprise Replication Guide*.

Improved Consistency Reporting after Enterprise Replication Repair Operations

Consistency reports generated after repair operations might show temporary inconsistencies between servers until all replicated transactions are completed on the target server. The `-R` option for **cdr check replicate** and **cdr check replicateset** commands now re-checks the repaired rows. If any rows are inconsistent, the check is rerun every five seconds, up to a maximum of 12 times (for a total elapsed time of 60 seconds). If rows are still inconsistent after the 12th check, a description of those inconsistent rows is displayed. This feature is documented in the *IBM Informix Enterprise Replication Guide*.

Administering and Monitoring Enterprise Replication with the SQL Administration API

Now you can use the SQL administration API to administer and monitor Enterprise Replication with the same syntax elements that are in the **cdr** command-line syntax.

The syntax for using Enterprise Replication **cdr** commands with the admin() and task() functions is described in the *IBM Informix Administrator's Reference*.

The **cdr** command-line syntax is described in the *IBM Informix Enterprise Replication Guide*.

Improved SQL Tracing with the SQL Administration API

You can use these new commands to manage SQL tracing by databases, sessions, and users: SET SQL TRACING DATABASE, SET SQL TRACING SESSION, and SET SQL TRACING USER. Previously, you could only trace SQL for all databases at the server. Now you can control which databases to include in the SQL trace. You can also turn tracing on or off for a specific session, and specify whether you want to trace SQL statements run by specific users. You can also suspend and resume all tracing at the server, without deallocating any resources, by using the SET SQL TRACING SUSPEND and RESUME commands.

All SET SQL TRACING commands are described in the *IBM Informix Guide to SQL: Syntax*.

Changing the Size of the First Extent of a Table

When you create a table, you specify a first extent size based on the eventual estimated size of the table. If the table becomes larger or smaller than that estimate, you might want to change the size of the first extent to avoid either having too many extents or creating extents that are larger than necessary. You can now change the size of the first extent of a table in a dbspace. When you change the size of the first extent, Informix records the change (in the system catalog and on the partition page), but only makes the actual change when the table is rebuilt or a new partition or fragment is created.

This feature, including the syntax for the new MODIFY EXTENT SIZE clause that you use with the ALTER TABLE statement, is documented in the *IBM Informix Guide to SQL: Syntax*.

Rolling Back SQL Transactions to a Savepoint

You can now declare or reference savepoint objects in SQL statements. A savepoint identifies an arbitrary location within the statements of an SQL transaction. Within its transaction, the savepoint resembles a statement label to which the ROLLBACK statement of SQL can cancel any changes to the database that the transaction produced between the savepoint and the ROLLBACK statement. A client application that declares one or more savepoints within a transaction can implement error-handling logic that rolls back only the portion of the transaction that follows the specified savepoint, rather than cancelling the entire transaction if an exception occurs.

The following new SQL statements are implemented for savepoints:

- The SAVEPOINT statement creates a savepoint within the current SQL transaction.
- The RELEASE SAVEPOINT statement destroys a specified savepoint, as well as any other savepoints that exist between the RELEASE statement and the savepoint that it references.
- The ROLLBACK WORK TO SAVEPOINT statement discards changes to the schema of the database or to its data values by statements that follow the savepoint but the effects of DDL and DML statements that preceded the savepoint persist.

This release of Informix directly supports the SQL statements that declare or reference savepoint objects in DB-Access and in UDRs written in the SPL, C, or Java languages. Methods that support savepoints are also supported in JDBC applications. Applications written in ESQL/C that use the savepoint statements of SQL are also supported with this release of Client SDK.

This feature is documented in *IBM Informix Guide to SQL: Syntax* and *IBM Informix JDBC Driver Programmer's Guide*.

Capturing Transactional Data with the Change Data Capture API

Use the Change Data Capture (CDC) API with your client applications to capture transactional data from Informix databases. The API includes a new system database called syscdc, which includes built-in SQL functions that control data capture.

This feature is documented in a new publication, the *IBM Informix Change Data Capture Programmer's Guide*.

Basic Text Search DataBlade Module Supports High-Availability Clusters

You can now use the Basic Text Search DataBlade module to perform searches on high-availability cluster servers by creating indexes in sbspaces. Previously, the Basic Text Search DataBlade module only supported the creation of indexes in extspaces, and thus could not participate in any queries on high-availability secondary servers and in backup and restore operations.

The Basic Text Search DataBlade module enhancements are documented in the *IBM Informix Database Extensions User's Guide*.

Querying XML Attributes with the Basic Text DataBlade Module

The Basic Text Search DataBlade module now supports searches on XML attributes in a document repository. The new **all_xmlattrs** parameter enables searches on all attributes that are contained in the XML tags or paths in a column that contains an XML document.

This feature is documented in the *IBM Informix Database Extensions User's Guide*.

Setting the Frequency of Error Checking for Smart Large Object Transmission

Use the **IFX_LOB_XFERSIZE** environment variable to specify the number of bytes in a CLOB or BLOB to transfer from a client application to the database server before checking for errors. The error check occurs each time the specified number of bytes is transferred. If an error occurs, the remaining data is not sent and an error is reported. If no error occurs, the file transfer will continue until it finishes. For example, if the value of **IFX_LOB_XFERSIZE** is set to 10485760 bytes (10 MB), error checking will occur after every 10485760 bytes of the CLOB or BLOB are sent. If you do not set this environment variable, the error check occurs after the entire BLOB or CLOB is transferred.

You should adjust the value of **IFX_LOB_XFERSIZE** to suit your environment. Set **IFX_LOB_XFERSIZE** low enough so that transmission errors of large BLOB or CLOB data types are detected early, but not so low that excessive network resources are consumed.

This feature is currently documented in the documentation notes for the *IBM Informix Guide to SQL: Reference*.

New Features in Version 11.50.xC2 of IBM Informix

IBM Informix, Version 11.50.xC2 contains new functionality in the following areas:

- High Availability
  - Reconfiguring Connection Manager while it's running
- Administration
  - Multiple copies of Informix on one Windows operating system

  - Enhanced Informix installation application on Mac OS X
  - Limiting memory use during Enterprise Replication synchronization
  - Obtaining Informix version information from the **cdr** utility
  - Monitoring Enterprise Replication with new SMI tables
  - Monitor the whole Enterprise Replication domain
  - Prevent ATS or RIS file generation
  - The ISM Administrator Program is included with Storage Manager on Windows
  - Limiting the number of sessions that can connect to the database server
  - New format for backup filters
  - Enhancements to the OpenAdmin Tool for Informix
- Performance
  - Controlling I/O of B-Tree Indexes with Compression Levels
- Application Development
  - Subquery support in UPDATE and DELETE statements
  - Longer return strings from string manipulation functions
- Security
  - Server-Specific Audit Configuration File Functionality

Reconfiguring Connection Manager while It's Running

In past releases, you had to stop the Connection Manager to modify how it was configured, and then restart it for the new configuration to take effect. With the new reload option (`-r`), you can reconfigure the Connection Manager without stopping and restarting it.

Connection Manager is primarily documented in the *IBM Informix Administrator's Guide* and the *IBM Informix Administrator's Reference*.

Multiple Copies of Informix on the Same Windows Computer

In version 11.10, you could not install and run multiple copies of the same version of Informix on the same Windows computer. Now you can do so by using the graphical user interface or by supplying installation parameters in a file to perform a silent, noninteractive installation. If the installation program detects that the same version of Informix is already installed, you can choose to install a new copy in another directory or you can choose to modify the existing installation.

This feature is documented in the *IBM Informix Installation Guide for Windows*.

Enhanced Installation Application on Mac OS X

The Informix installation application can automatically tune the kernel settings to values that support a working instance of the database server on your computer. In addition, after setup of a demonstration database server is complete, a terminal icon appears inside the installation directory for easier navigation.

This feature is documented in the *IBM Informix Installation Guide for UNIX, Linux, and Mac OS X*.

Controlling Memory Use during Enterprise Replication Synchronization

Enterprise Replication uses the value of the CDR_QUEUEMEM configuration parameter as the size of the send queue during a synchronization operation. To specify a larger or smaller size for the send queue during a particular synchronization operation, use the **--memadjust** option for the **cdr sync replicate** and **cdr sync replicateset** commands. If you synchronize with the **cdr start replicate**, **cdr start replicateset**, or **cdr realize template** commands, you can specify that the synchronization operation is performed as a foreground process with the new **--foreground** option, and adjust the send queue size with the **--memadjust** option.

This feature is documented in the *IBM Informix Enterprise Replication Guide*.

Obtaining Informix Version Information From the cdr Utility

In previous releases, you could not obtain product version information by using the cdr utility. You can now use the **cdr -V** command to display the version of the database server that is currently running.

This feature is documented in the *IBM Informix Enterprise Replication Guide*.

Monitoring Enterprise Replication with New SMI Tables

SMI tables provide a way to obtain information by using SQL commands, either locally or from remote servers. The following new SMI tables contain information about Enterprise Replication that you can use to monitor status and diagnose problems:

- syscdr_state

  Contains information on whether Enterprise Replication, data capture, data apply, and the network between servers is active.
- syscdr_ddr

  Contains information about the status of log capture and the proximity or status of transaction blocking (DDRBLOCK) or transaction spooling.
- syscdr_nif

  Contains information about network connections and the flow of data between Enterprise Replication servers.
- syscdr_rcv

  Contains information about transactions being applied on target servers and acknowledgements being sent from target servers.
- syscdr_atsdir

  Contains information about the contents of the ATS directory.
- syscdr_risdir

  Contains information about the contents of the RIS directory.
- syscdr_ats

  Contains the first ten lines of the header of each ATS file.
- syscdr_ris

  Contains the first ten lines of the header of each RIS file.
- syscdr_rqmstamp

Contains information about which transaction is being added into each queue.

- syscdr_rqmhandle

  Contains information about which transaction is being processed in each queue.

This feature is documented in the *IBM Informix Enterprise Replication Guide*.

Monitoring the Whole Enterprise Replication Domain

You can now monitor the status of all Enterprise Replication servers from any one of those servers with the new **cdr view** command. Specify one or more subcommands, depending on what information you want to monitor. You can automatically rerun the **cdr view** command every specified interval by using the `--repeat` option.

This feature is documented in the *IBM Informix Enterprise Replication Guide*.

Preventing ATS or RIS File Generation

You can prevent generating ATS or RIS files by setting the ATS or RIS directory to **/dev/null** on UNIX and **NUL** on Windows.

This feature is documented in the *IBM Informix Enterprise Replication Guide*.

The ISM Administrator Program is Included with Storage Manager on Windows

You can use the ISM Administrator program to manage your ISM server and storage devices from a graphical interface on Windows platforms. You can monitor the progress of your backup and restore operations, manage backed-up data, and control your storage media and storage devices through the ISM Administrator program.

The ISM Administrator program is documented in the *IBM Informix Storage Manager Administrator's Guide*.

Limiting the Number of Sessions That Can Connect to Dynamic Server

You can now limit the number of sessions that can connect to the server. You do this by setting the LIMITNUMSESSIONS configuration parameter to the maximum number of sessions that you want connected to the database server. Optionally, you can also specify whether you want the server to print messages when the number of sessions approaches a specified maximum number. You can use **onmode -wm** and **onmode -wf** commands to turn this configuration parameter on or off or change the value of the configuration parameter.

This feature is primarily documented in the *IBM Informix Administrator's Reference*.

New Format for Backup Filters

Backups have a new format for version 11.50.xC2 if they are performed with the ontape command using a filter specified by the BACKUP_FILTER configuration parameter. The new backup format is not backwards compatible. Any backups that were made prior to version 11.50.xC2 must be restored with the earlier versions of the filter.

Enhancements to the OpenAdmin Tool for Informix

The OpenAdmin Tool for Informix Version 2.22 has the following enhancements:

- Enhanced security for the SQL Toolbox features
- Enterprise Replication Plug-in for OpenAdmin Tool for Informix Version 1.1:
  - A Routing Topology page for monitoring Enterprise Replication nodes at a domain level
  - New Node Details pages to view errors and configuration settings for individual ER nodes

The OpenAdmin Tool for Informix has enhanced security for the SQL Toolbox features.

See the OpenAdmin Tool for Informix release notes for more information.

Controlling I/O of B-Tree Indexes with Compression Levels

B-tree scanners can now compress indexes by merging two partially used index pages if the data on those pages totals a set level (low, medium, or high). You can specify the index compression level by modifying the value of the compression field of the BTSCANNER configuration parameter option, by running an **onmode -C compression *value*** command, or by running an SQL administration API function with a SET INDEX COMPRESSION command.

This feature is primarily documented in the*IBM Informix Performance Guide*.

Subquery Support in UPDATE and DELETE Statements

The FROM clause of a subquery in the WHERE clause of the DELETE or UPDATE statement can specify as a data source the same table or view that the FROM clause of the DELETE or UPDATE statement specifies. In this version, Dynamic Server supports DELETE and UPDATE operations with subqueries that reference the same table object if all of the following conditions are true:

- The subquery either returns only a single row, or else has no references to columns in any table that is not listed in its FROM clause.
- The subquery is specified in the WHERE clause of the outer DELETE or UPDATE statement, using the Condition with Subquery syntax.

If you call SPL routines within the same subquery, these cannot reference the table that is being modified.

This feature is documented in the *IBM Informix Guide to SQL: Syntax*.

Longer Return Strings from String Manipulation Functions

The following SQL string-manipulation functions now support operations where the returned string can be up to 32 KB in length:

- LPAD
- RPAD
- REPLACE

- SUBSTR
- SUBSTRING
- TRIM
- LTRIM
- RTRIM
- ENCRYPT_AES
- ENCRYPT_TDES
- DECRYPT_CHAR
- DECRYPT_BINARY
- CONCAT (and the || operator)

In addition, the **CONCAT** function and the || operator now provide native support for the LVARCHAR data type.

This feature is documented in the *IBM Informix Guide to SQL: Syntax*.

Server-Specific Audit Configuration File Functionality

Each server instance uses its own **adtcfg.***servernumber* file if the audit configuration parameters are changed with the **onaudit** utility. The new **-n** option for the **onshowaudit** utility reads the server-specific audit configuration (**adtcfg.***servernumber*) file.

This feature is documented in the *IBM Informix Security Guide*.

New Features in Version 11.50.xC1 of IBM Informix

IBM Informix, Version 11.50.xC1 contains new functionality in the following areas:

- High Availability
  - SQL administration API commands to configure high availability clusters added
  - Enhanced connection management for high-availability clusters
  - Update data on secondary servers
  - New options to troubleshoot high-availability clusters
  - Support for transient types on high-availability cluster secondary servers
- Administration
  - Enhanced configuration options during installation
  - Install as the Local System User ID support added (Windows)
  - Enhancements to the OpenAdmin Tool for Informix
  - Improved onconfig.std and new default values for configuration parameters
  - Enhanced Data Server client session information
  - Enhanced shared-memory dump file size control
  - Enhanced startup script customization
  - New Options for Configuring Storage Space Monitoring
  - Updating Table Statistics with AUS Scheduler Tasks
- Performance
  - Control external directives for a session
  - Improved Query Performance for Large Integers and Serial Data

  - Support for Obtaining Explain Output in XML Format Added
- Application Development
  - DRDA Protocol Configuration During Installation Added
  - SQL expressions with the IS [NOT] NULL predicate
  - Determine data currency with a version column
  - Distributed Relational Database Architecture (DRDA) Enhancements
  - J/Foundation Upgraded to JRE 5.0
  - Improvements to the Basic Text Search DataBlade Module
  - New Support for Dynamic SQL Statements in SPL Routines
  - New Support for Extensible Stylesheet Language Transformation (XSLT)
- Security
  - Single Sign-on Support (SSO) Added
  - Support for Encrypting Data by Using Secure Sockets Layer (SSL) Communications Added

SQL Administration API Commands to Configure High-Availability Clusters Added

You can now use SQL administration API commands to perform the equivalent of **onmode -d** commands to configure high-availability clusters.

SQL administration API commands are documented in the *IBM Informix Administrator's Reference*.

Enhanced Connection Management for High-Availability Clusters

The new Connection Manager dynamically routes client application connection requests to the most appropriate server in a high-availability cluster. Connection Manager connects to each of the servers in the cluster and gathers statistics about the type of server, unused workload capacity, and the current state of the server. From this information, the Connection Manager redirects the connection to the appropriate server. In addition, Connection Manager Arbitrator provides automatic failover logic for high-availability clusters. Using a configuration file, you specify which secondary server takes over if the primary server fails.

Connection Manager is installed with Client SDK. Connection Manager requires that applications connecting to it be compiled with Client SDK 3.50 or Data Server Driver for JDBC and SQLJ.

Connection Manager is primarily documented in the *Administrator's Guide* and the *Administrator's Reference*.

New Options to Troubleshoot High-Availability Clusters

The following new **onstat** commands provide diagnostic information about high-availability clusters:

- **onstat –g dri**
- **onstat –g sds**
- **onstat –g rss**
- **onstat -g proxy**

The following new **sysmaster** tables contain equivalent information as the new **onstat** commands:

- **sysproxyagents**
- **sysproxydistributers**
- **sysproxysessions**
- **sysproxytxnops**
- **sysproxytxns**

These options are documented in the *IBM Informix Administrator's Guide* and *IBM Informix Administrator's Reference*.

Update Data on Secondary Servers

You can configure secondary servers so that client applications can send them transactions that update data. The ability to update secondary servers is not enabled by default. To enable updates to secondary servers, set the UPDATABLE_SECONDARY configuration parameter to the number of SMX pipes between the primary and secondary servers. The maximum value of this parameter is two times the number of CPU virtual processors. To set the secondary server as read-only, which is the default behavior, set UPDATABLE_SECONDARY to 0.

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

Support for Transient Types on High-Availability Cluster Secondary Servers

Transient unnamed complex data types (ROW, SET, LIST, and MULTISET) can now be used on high-availability cluster secondary servers, whether the secondary servers are read-only or use updatable secondary functionality.

The following types of operations that use transient types are supported on secondary servers:

- SQL queries that explicitly use transient types
- SQL queries that use derived tables, collection subqueries, and XML functions (these statements implicitly use transient types)
- Temporary tables created with the CREATE TEMP statement that use transient types

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

Enhanced Configuration Options During Installation

You can use the new Instance Configuration Wizard to automatically create the database server configuration file (ONCONFIG) during installation. Access the wizard in the installation program with either GUI or console installation modes:

- UNIX and Linux: Select to create a demonstration database server and choose to customize the default configuration file
- Windows: Choose custom installation setup, select "Initialize Server," and select the option to enable a default configuration file for your hardware and user needs

Provide the information for the instance that you are installing, such as the number of CPUs, memory, disk space, and estimates of online transactions and query clients. The wizard ensures that your settings are valid, and it calculates values for other server configuration parameters based on your settings. Your custom configuration information is stored in the ONCONFIG file so that when you start the instance after the product is installed, the instance runs with your settings.

This feature is documented in the IBM Informix Installation Guide for Microsoft Windows and the *IBM Informix Installation Guide for UNIX, Linux, and Mac OS X*.

Install as the Local System Account support added (Windows)

In past releases, the InformixWindows Service was only allowed to log on as the **informix** user.

Starting with version 11.50, you can install the database server on Windows as the local system account. Select the Local System User option in the installation program. That option provides the same privileges as the **informix** user account; however, it uses an internal account that does not require a password. The local system account is used by the operating system and services running under Windows during the installation of Informix.

You can choose not to create an **informix** user account at all, but if you do so, you will not be able to use Enterprise Replication betweenInformix on UNIX and Windows platforms.

This feature is documented in the *IBM Informix Installation Guide for Windows*.

Enhanced Data Server Client Session Information

You can use the new **syssesappinfo** table in the **sysmaster** database to view Data Server client session information. The table shows the client session ID, session application name, and a session value in the **sesapp_sid**, **sesapp_name**, and **sesapp_value** columns.

You can now also display client session information using the **onstat -g ses** command.

The **syssesappinfo** table and the **onstat -g ses** command are documented in the *IBM Informix Administrator's Reference*.

Improved onconfig.std File and New Default Values for Configuration Parameters

The Informix configuration file, onconfig.std, has been improved in the following ways:

- The onconfig.std file is easier to read because the comments and the parameters are listed separately and grouped by functional areas.
- Most supported configuration parameters are now included in the file.
- Deprecated configuration parameters were removed.
- Some configuration parameters that specify sizes have now have higher values.
- Some configuration parameters that specify file locations now have more secure default locations under the $INFORMIXDIR directory.

Tip: If you use a utility like grep on the onconfig.std file, include a new-line character when you search for configuration parameters names so that you do not return the parameter description. For example, the following command returns both the description and the configuration parameter:

```
grep "MSGPATH" onconfig.std
# MSGPATH      - The name of the IDS message log file
MSGPATH $INFORMIXDIR/tmp/online.log
```

In contrast, this command returns only the configuration parameter:

```
grep "^MSGPATH" onconfig.std
MSGPATH $INFORMIXDIR/tmp/online.log
```

The following table lists the configuration parameters that are added to the onconfig.std file and their values.

Table 2-1. Configuration Parameters added to onconfig.std

Configuration Parameter	Value
ADMIN_USER_MODE_WITH_DBSA	none
BTSCANNER	num=1,priority=low,threshold=5000, rangesize=-1,alice=6
BACKUP_FILTER	none
BAR_DEBUG	0
CDR_SUPPRESS_ATSRISWARN	none
DD_HASHMAX	10
DD_HASHSIZE	31
DEF_TABLE_LOCKMODE	page
DS_HASHSIZE	31
DS_POOLSIZE	127
ENCRYPT_CDR	none
ENCRYPT_CIPHERS	none
ENCRYPT_HDR	none
ENCRYPT_MAC	none
ENCRYPT_MACFILE	none
ENCRYPT_SMX	none
ENCRYPT_SWITCH	none
EXT_DIRECTIVES	0
FAILOVER_CALLBACK	none
FASTPOLL	1
HA_ALIAS	none
LOG_INDEX_BUILDS	none
MAX_INCOMPLETE_CONNECTIONS	1024
PC_HASHSIZE	31
PC_POOLSIZE	127
PLCY_HASHSIZE	127
PLCY_POOLSIZE	31

Table 2-1. Configuration Parameters added to onconfig.std (continued)

Configuration Parameter	Value
PLOG_OVERFLOW_PATH	UNIX: $INFORMIXDIR/tmp Windows: none
RESTORE_FILTER	none
SBSPACETEMP	none
SDS_ENABLE	none
SDS_PAGING	none
SDS_TEMPDBS	none
SDS_TIMEOUT	20
SECURITY_LOCALCONNECTION	none
SQLTRACE	Commented out: # SQLTRACE level=low,ntraces=1000,size=2,mode=global
SSL_KEYSTORE_LABEL	none
STMT_CACHE	0
STMT_CACHE_HITS	0
STMT_CACHE_NOLIMIT	0
STMT_CACHE_NUMPOOL	1
STMT_CACHE_SIZE	512
TEMPTAB_NOLOG	0
UNSECURE_ONSTAT	none
UPDATABLE_SECONDARY	0
USRC_HASHSIZE	31
USRC_POOLSIZE	127
VPCLASS	cpu,num=1,noage Commented out: # VPCLASS aio,num=1 Commented out: #VPCLASS jvp,num=1

The following table lists configuration parameters whose value in the onconfig.std file has been updated.

Table 2-2. Configuration Parameters Updated in onconfig.std

Configuration Parameter	Previous value	New value
ADMIN_MODE_USERS	1	None
ALARMPROGRAM	UNIX: /usr/informix/etc/alarmprogram.sh Windows: None	UNIX: $INFORMIXDIR/etc/alarmprogram.sh Windows: $INFORMIXDIR\etc\alarmprogram.bat
BAR_ACT_LOG	/usr/informix/bar_act.log	UNIX: $INFORMIXDIR/tmp/bar_act.log Windows: $INFORMIXDIR\tmp\bar_act.log

Table 2-2. Configuration Parameters Updated in onconfig.std (continued)

Configuration Parameter	Previous value	New value
BAR_BSALIB_PATH	UNIX: $INFORMIXDIR/lib/ libsad001.so Windows: libbsa.dll	None
BAR_DEBUG_LOG	UNIX: /usr/informix/bar_dbug.log Windows: bar_dbug.log	UNIX: $INFORMIXDIR/tmp/ bar_dbug.log Windows: $INFORMIXDIR\tmp\ bar_dbug.log
BUFFERPOOL	Operating systems with 2K page size: default,buffers=5000,lrus=8, lru_min_dirty=50,lru_max_dirty=60 size=2k,buffers=5000,lrus=8, lru_min_dirty=50,lru_max_dirty=60 Operating systems with 4K page size: default,buffers=1000,lrus=8, lru_min_dirty=50,lru_max_dirty=60 size=4k,buffers=1000,lrus=8, lru_min_dirty=50,lru_max_dirty=60	Operating systems with 2K page size: default,buffers=10000,lrus=8, lru_min_dirty=50.00, lru_max_dirty=60.50 size=2k,buffers=50000,lrus=8, lru_min_dirty=50, lru_max_dirty=60 Operating systems with 4K page size: default,buffers=10000,lrus=8, lru_min_dirty=50.00, lru_max_dirty=60.50 size=4k,buffers=10000,lrus=8, lru_min_dirty=50, lru_max_dirty=60
CLEANERS	1	8
CONSOLE	UNIX: /dev/console Windows: console.log	UNIX: $INFORMIXDIR/tmp/ online.con Windows: online.con
DB_LIBRARY_PATH	commented out: # DB_LIBRARY_PATH $INFORMIXDIR/extend	commented out: # DB_LIBRARY_PATH
DRLOSTFOUND	UNIX: /usr/etc/dr.lostfound Windows: \tmp	UNIX: $INFORMIXDIR/etc/ dr.lostfound Windows: $INFORMIXDIR\tmp
DUMPDIR	UNIX: /usr/informix/tmp Windows: INFORMIXDIR\tmp	UNIX: $INFORMIXDIR/tmp Windows: $INFORMIXDIR\tmp
LISTEN_TIMEOUT	10	60
LOCKS	2000	20000
LOGBUFF	32	64
LOGSIZE	2000	10000
LTAPEDEV	UNIX: /dev/tapedev Windows: \\.TAPE1	UNIX: /dev/tapedev (same as previous value) Windows: NUL
MIRRORPATH	None	UNIX: $INFORMIXDIR/tmp/ demo_on.root_mirror Windows: none

Table 2-2. Configuration Parameters Updated in onconfig.std (continued)

Configuration Parameter	Previous value	New value
MSGPATH	UNIX: /usr/informix/online.log Windows: online.log	UNIX: $INFORMIXDIR/tmp/ online.log Windows: online.log
NETTYPE	UNIX: none Windows: onsoctcp,drsoctcp,1,NET	UNIX: ipcshm,1,50,CPU Windows: none
PHYSBUFF	32	128
PHYSFILE	2000	50000
RA_PAGES	None	64
RA_THRESHOLD	None	16
ROOTPATH	UNIX: /dev/online_root Windows: None	UNIX: $INFORMIXDIR/tmp/ demo_on.rootdbs Windows: None
ROOTSIZE	30000	200000
SHMVIRT_ALLOCSEG	0	0,3
SHMVIRTSIZE	8192	32656
SYSALARMPROGRAM	UNIX: /usr/informix/etc/ evidence.sh Windows: INFORMIXDIR\etc\ evidence.bat	UNIX: $INFORMIXDIR/etc/ evidence.sh Windows: Commented out: # SYSALARMPROGRAM $INFORMIXDIR\etc\evidence.bat
TAPEBLK	32	UNIX: 32 Windows: 16
TAPESIZE	10240	0

The following configuration parameters were removed from the onconfig.std file:

- AFF_NPROCS
- AFF_SPROC
- NOAGE
- NUMCPUVPS
- PHYSDBS
- JDKVERSION

The changes to configuration parameter default values are documented in the *IBM Informix Administrator's Reference*.

Enhancements to the OpenAdmin Tool for Informix

The OpenAdmin Tool for Informix has the following enhancements:

- Enhanced high availability cluster management (MACH) support:
  - Starting and stopping a remote server in the cluster
  - Encryption for idsd
  - Connection Manager Wizard

  - SD Secondary Disk Setup Wizard
- New security features:
  - Read-only user group support
  - User privileges administration
- New Task Scheduler to create and modify Scheduler tasks
- New server administration tools:
  - Automated update of statistics
  - Dynamic update of configuration parameters
  - System validation
  - Virtual processor administration
- Improved performance monitoring tools:
  - Improved historical performance data tracking
  - New system reports
  - Memory pool statistics
- Improved usability:
  - Menu restructuring and new look and feel
  - New contextual help topics
  - HOWTO.html task guide

See the OpenAdmin Tool release notes for more information.

Enhanced Shared-Memory Dump File Size Control

By using the new options for the DUMPSHMEM configuration parameter and the **onstat** utility, you can control how much memory is written to a dump file. These options exclude the buffer pool in the resident memory, which can result in a much smaller file.

Use the DUMPSHMEM configuration parameter to automatically create a dump file when an assertion fails. Set DUMPSHMEM to 2 to create a shared memory dump that excludes the buffer pool. You can dynamically change the value of DUMPSHMEM with **onmode -wm** and **onmode -wf**.

Use the new **nobuffs** option for the **onstat -o** command to write the contents of shared memory without the buffer pool to a specified file. If you do not specify a file, the contents are written to a file called **onstat.out** in the current working directory.

If you use **onstat -o** without an option, the **nobuffs** or **full** option is controlled by the DUMPSHMEM configuration parameter setting.

Any **onstat** options, such as **onstat -i** (interactive), that work with a full dump file will work with a dump file that excludes the buffer pool.

This feature is primarily documented in the *IBM Informix Administrator's Reference*.

Enhanced Startup Script Customization

You can customize startup scripts and automate startup with the new `-w` option for the **oninit** utility. The `-w` option forces the server to wait until it successfully initializes before returning a shell prompt. The `-w` option provides a return code so that you can check if the system started without incident. If the server fails to initialize within a configurable timeout period, the utility returns a 1 and writes a message to the online.log file. A return code of 0 indicates a successful initialization.

The –w option works with both disk-space initialization and shared-memory initialization `oninit` options.

In a high-availability environment, you can only use the **oninit -w** command on primary servers; it is not valid on secondary servers.

The -w option to the **oninit** utility is documented in the *IBM Informix Administrator's Reference*.

New Options for Configuring Storage Space Monitoring

You can configure how you are notified when a storage space or partition becomes full with the new STORAGE_FULL_ALARMS configuration parameter. By default, you will now receive alarms and messages in the online message log when a dbspace, sbspace, temporary dbspace, blobspace, or partition becomes full. You can specify the level of the alarms or disable them. You can also specify the time interval between alarms and messages. By default, level 3 alarms are enables with an interval of 10 minutes.

This features is documented in the *IBM Informix Administrator's Guide* and *IBM Informix Administrator's Reference*.

Updating Table Statistics with AUS Scheduler Tasks

The new Auto Update Statistics (AUS) scheduler tasks can improve the performance of the database server by regularly providing the query optimizer with updated table statistics as the basis for efficient query plans.

- Use the **Auto Update Statistics Evaluation** task to generate a prioritized set of UPDATE STATISTICS statements. Built-in policies specify criteria for identifying the tables whose column distributions should be recalculated, but you can modify these default policies.
- Use the **Auto Update Statistics Refresh** task to run the statements during a recurring time interval that you set.

This feature is primarily documented in the *IBM Informix Performance Guide*.

Control External Directives for a Session

You now can use the new EXTDIRECTIVES session environment option of the SET ENVIRONMENT statement to control whether external directives are enabled, disabled, or have default behavior during a session. Specify the default directives behavior with the EXT_DIRECTIVES configuration parameter and the client-side IFX_EXTDIRECTIVES environment variable.

For more information, see the SET ENVIRONMENT syntax diagram and the section on the EXTDIRECTIVES option in the *IBM Informix Guide to SQL: Syntax*.

Improved Query Performance for Large Integers and Serial Data

The ISO standard data types BIGINT and BIGSERIAL are provided as alternatives to INT8 and SERIAL8. BIGINT and BIGSERIAL data types can provide better performance than the INT8 and SERIAL8 data types.

This feature is primarily documented in the *IBM Informix Guide to SQL: Reference*.

Support for Obtaining Explain Output in XML Format Added

EXPLAIN_SQL() is a new routine that prepares a query and returns a query plan in XML. The IBM Data Studio Administration Console can use EXPLAIN_SQL to obtain the XML query plan, interpret the XML, and render the plan visually. Vendors can use this routine to obtain explain output in XML format.

For information on using IBM Data Studio, see IBM Data Studio documentation.

DRDA Protocol Configuration During Installation Added

It's easier now than in past releases to set up an instance to use a variety of database clients. When you install Informix 11.50, the installer enables you to configure a database server alias and a port for clients that use the Distributed Relational Database Architecture (DRDA) protocol. By default, those items are configured for you unless you deselect DRDA support. DRDA is for open development of applications that allow access of distributed data.

This feature is documented in the IBM Informix Installation Guide for Microsoft Windows and *IBM Informix Installation Guide for UNIX, Linux, and Mac OS X*.

SQL Expressions with the IS [NOT] NULL Predicate

Now you can use SQL expressions as operands of the IS NULL and IS NOT NULL predicate. Previously, you could only use column name operands with IS NULL and IS NOT NULL, which return a TRUE or FALSE result.

The use of IS NULL or IS NOT NULL with expressions enables you to provide a value for entries that otherwise are not computable because NULL is not a valid numerical value. For more information on using IS NULL or IS NOT NULL with expressions, and for an example of a CASE expression that allows NULL values to be treated as 0 for the purpose of arithmetic computations, see the *IBM Informix Guide to SQL: Syntax*.

Determine Data Currency with a Version Column

You can now add a version column to a table to contain both a checksum and a version number. Use a version column to detect if a row has been updated since it was originally queried. Also, version numbers enable you to detect differences if a row is deleted and another row is re-inserted into a table. To add a version column, use the ALTER TABLE *tablename* ADD VERCOLS statement.

Web applications can use a version column to determine if information about a previously retrieved object is still current. For example, a Web application can display an item to a customer, and then when the customer decides to buy the item, the application checks the version column of the item's row to determine if the information about the item has changed.

A version column might also improve the updatable secondary performance in a high-availability cluster because during collision detection, the primary server only needs to compare the version of the updated row instead of the entire contents of the row.

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

Distributed Relational Database Architecture Enhancements

Distributed Relational Database Architecture (DRDA) enables communication between applications and database systems on disparate platforms and enables relational data to be distributed among multiple platforms. Enhancements to DRDA functionality include:

- Support for retrieving ISAM errors
- Support for connecting to a database that has a name that is longer than 18 bytes
- Batching of multiple statements into one request
- Support for displaying DRDA session information in the new syssesappinfo table in the sysmaster database and with the onstat -g ses command
- Support for using IS NULL or IS NOT NULL with expressions
- Support for using the Secure Sockets Layer (SSL) protocol for DRDA communications
- Support for DRDA connections between primary and shared disk secondary servers in high availability clusters

These features are primarily documented in the *IBM Informix Administrator's Guide*.

J/Foundation Upgraded to JRE 5.0

The J/Foundation component of Informix has been upgraded from JRE 1.4.2 to JRE 5.0 (Java version 1.5.0). This upgrade supports JRE 5.0 and JDK 5.0 for server-based Java routines.

The JRE 5.0 upgrade has the following changes:

- New settings for the JVPJAVAVM configuration parameter
- The JDKVERSION configuration parameter has been deprecated

Depending on your application requirements, you should test and possibly increase the value for the JVM_MAX_HEAP_SIZE environment variable to configure the heap size for the new JVM.

This feature is documented in the *J/Foundation Developer's Guide*.

Improvements to the Basic Text Search DataBlade module

You have the following new facilities for using **bts** indexes:

- Obtain a list of fields in an index that you can search on with the new bts_index_fields() function

- Specify a custom stopword list with a new index parameter instead of using the default stopword list
- Index XML documents with new index parameters

To use these features, you must register this new version of the Basic Text Search DataBlade module in your databases.

These features are documented in the *IBM Informix Database Extensions User's Guide*.

New Support for Dynamic SQL Statements in SPL Routines

In earlier versions of the Informix SPL language, the form of SQL statements in a UDR was known when the SPL routine was written. Users can now specify queries dynamically at runtime in SPL routines, using the EXECUTE IMMEDIATE and PREPARE statements to execute dynamically constructed SQL statements and to define and manage cursors and prepared objects with the DECLARE, OPEN, FETCH, CLOSE, and FREE statements.

For some of these Dynamic SQL statements, the SPL syntax is a subset of the existing ESQL/C syntax, but character expressions that evaluate to SQL statement text are now supported by PREPARE and EXECUTE IMMEDIATE statements in SPL routines. In addition, Boolean conditions in SPL statements can specify SQLCODE as an expression that returns the SQLCA status from the most recently executed SQL statement.

This feature increases the flexibility of UDRs written in the SPL language, and simplifies the migration to IDS of stored procedures and applications written for other database servers that support Dynamic SQL operations. This feature is documented in the *IBM Informix Guide to SQL: Syntax*.

New Support for Extensible Stylesheet Language Transformation (XSLT)

You can now use built-in SQL functions to transform XML documents using extensible stylesheet language transformations (XSLTs). IDS supports the XSLT 1.0 standard for style sheets with the Xalan and Xerces libraries.

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

Single Sign-on Support Added

With single sign-on, users authenticate themselves once and the authentication is carried securely over trusted and nontrusted networks. The users provide a valid user ID and password when they log in to a client computer, and they can access the database server and other SSO-enabled services without having to log in again. In the past, users had to log in multiple times.

IBM Informix delivers support for SSO in the Generic Security Services Communications Support Module (GSSCSM). GSSCSM works with Kerberos, which is a network authentication protocol. You must deploy a Kerberos authentication protocol that supports the Generic Security Services Application Programming Interface (GSSAPI) before you can use SSO with IBM Informix. After you enable SSO, you benefit from centralized management of authentication.

This feature is primarily documented in the *IBM Informix Security Guide*.

Support for Encrypting Data by Using Secure Sockets Layer (SSL) Communications Added

You can now configure IBM InformixIDS to use the SSL protocol, which encrypts data in TCP/IP connections between two points over a network. The SSL protocol is an alternative to the IDS-specific encryption Communication Support Module (CSM) and simple password CSM for CSDK clients. You must use SSL to encrypt data in communications between IDS and DRDA clients.

This feature is primarily documented in the *IBM Informix Security Guide*.

New Features in Version 11.10.xC2 of IBM Informix

IBM Informix, Version 11.10.xC2 contains the following new features:

- Prevent Shared Memory Addresses from being used by Informix
- Preview Enterprise Replication ATS or RIS Repair Operations

Prevent Shared Memory Addresses from being used by Informix

You can specify virtual memory address ranges that Informix cannot use to attach shared memory with the SHMNOACCESS configuration parameter. SHMNOACCESS is used to avoid specific range process addresses, which in turn avoids conflicts with operating system libraries.

Each address in each range must start in hex format, each address in a range must be separated by a hyphen, and each range must be separated by a comma, as the following example shows:

```
SHMNOACCESS 0x70000000-0x75000000,
0x7A000000-0x80000000
```

The SHMNOACCESS configuration parameter is documented in the *IBM Informix Administrator's Reference*.

Preview Enterprise Replication ATS or RIS Repair Operations

You can preview the operations that an ATS or RIS repair operation performs by using the new **--check** option with the **cdr repair ats** or **cdr repair ris** command. The **--check** option specifies that the repair operations are displayed to stderr but not actually performed. For more information, issue the **cdr repair ats -h** or **cdr repair ris -h** command.

This feature is documented in the *IBM Informix Enterprise Replication Guide*.

New Features in Version 11.10 of IBM Informix

Scalability, High Availability, and Performance

- Multiple Remote Standalone Secondary Servers
- Multiple Shared Disk Secondary Servers
- Backup and Restore to Directories with ontape

- Continuous Logical Log Restore
- Encrypted Communications for HDR
- Improved Parallelism during Backup and Restore
- Automatic Ordering of dbspaces during Backup and Restore
- RTO Policy to Manage Server Restart
- Nonblocking Checkpoints
- Performance Improvements for Enterprise Replication
- ON-Bar Performance Report
- Transform Data During Backup and Restore
- Improved Performance for Cooked Files with Direct I/O
- Improved Performance of Online Index Creation

Administration
- SQL administration API
- Schedule Administrative Tasks
- Monitor and Analyze Recent SQL Statements
- Dynamically Change Enterprise Replication Configuration Parameters and Environment Variables
- Dynamically Rename Enterprise Replication Columns, Tables, and Databases
- Truncate Replicated Tables
- Improved Statistics Maintenance
- Installation Improvements on Windows Platforms
- Session Configuration Routines
- Multiple Users for Administration Mode
- OpenAdmin (a PHP-based admin utility)

Integrated Solutions
- Named Parameters in a JDBC CallableStatement
- Index Binary Data Types
- Trigger Enhancements
- Derived tables in the FROM Clause of Queries
- Index Self-Join Query Plans
- Optimizer Directives in ANSI-Compliant Joined Queries
- Deployment Wizard
- Enhanced Concurrency with Committed Read Isolation
- Enhanced Data Types and UDR Support in Cross-Server Distributed Operations
- XML Publishing
- Index Hierarchical Data
- Basic Text Search
- Improved Concurrency with Private Memory Caches for Virtual Processors
- Web Feature Services for Spatial Data
- Support for Common Clients with DRDA
- Statement Labels, GOTO, and LOOP Statements in the SPL language
- New SQL Functions
- Automatic Re-Compilation of Prepared Statements

Security

- Label-Based Access Control (LBAC)

Multiple Remote Standalone Secondary Servers

The high availability functionality currently available in IDS with HDR is extended with the capability to add multiple remote standalone (RS) secondary servers. This provides new high-availability configuration options that can be combined with HDR. RS secondary servers can be geographically distant from the primary server, serving as remote back up servers in disaster-recovery scenarios. Each RS secondary server maintains a complete copy of the database, with updates transmitted asynchronously from the primary server over secure network connections. In the event of a failover from the primary server to the HDR server, one of the RS secondary servers can be promoted to become the HDR secondary.

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

Multiple Shared Disk Secondary Servers

IBM Informix now provides multiple server access to a single shared disk for high availability and query workload distribution. The shared disk (SD) solution for secondary servers provides configuration options that can be combined with remote standalone secondary servers and HDR. The primary server has write access to a disk or disk array, while all SD secondary servers have read-only access. An SD secondary server does not maintain a copy of the physical database on its own disk space; rather, it shares disks with the primary server. A single copy of data is shared among the servers, lowering data storage costs. New secondary servers can be added dynamically to share query workload and expand availability options. A secondary server can be promoted to the primary as needed to maintain continuous availability.

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

Backup and Restore to Directories with ontape

You can use the **ontape** utility to backup and restore data from file system without interactive prompts. To enable this feature, set the TAPEDEV and LTAPEDEV configuration parameter to a valid directory of a local or remote mounted file system. The **ontape** utility generates file names automatically and performs physical and log backups.

This feature is documented in the *IBM Informix Backup and Restore Guide*.

Continuous Logical Log Restore

This feature lets you perform continuous restore of logical log backups using the **ontape** and ON-Bar utilities. Use continuous log restore to keep a second system (hot backup) available to replace the primary system if the primary system fails. Logical logs backed up on the primary system can be restored on the secondary system as they become available. If the primary system fails, the remaining available logical logs can be restored on the second system, which can then be brought online and function as the new primary system. You can suspend the log restore upon exit from the restore command and let the instance resume it with successive restore commands.

This feature is documented in the *IBM Informix Backup and Restore Guide*.

Encrypted Communications for HDR

You can encrypt communication between an HDR pair, to secure the transmission of data over unsecured networks, including the internet. You use new configuration parameters to enable encryption between the HDR servers and specify encryption options. After you enable encryption, the HDR primary database server encrypts the data before sending it to the secondary database server. The secondary database server decrypts the data. HDR encryption works in conjunction with Enterprise Replication encryption and operates whether Enterprise Replication encryption is enabled or not.

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

Improved Parallelism during Backup and Restore

This feature lets ON-Bar backup and restore a whole system using parallel I/O, which reduces the total time that is required to complete the backup or restore. Parallel whole system backups are still restorable to a consistent state without log backup and log restore.

This feature is documented in the *IBM Informix Backup and Restore Guide*.

Automatic Ordering of dbspaces during Backup and Restore

IDS now makes intelligent decisions regarding the ordering of dbspaces during backup and restore to achieve maximum parallelism, thus reducing the backup and restore time necessary. For example, if the largest dbspace is backed up in parallel to other smaller dbspaces, the complete system backup will take less time. During a restore, dbspaces are restored in the same order in which they were backed up, reducing restore time.

RTO Policy to Manage Server Restart

You can now create a recovery time objective (RTO) policy to set the amount of time, in seconds, that IBM Informix has to recover from a problem after you restart the server. You do this using a new configuration parameter, RTO_SERVER_RESTART. This configuration parameter enables failure recovery to meet the RTO policy by monitoring the workload and triggering checkpoints in a timely manner to ensure failure recovery will meet the policy.

This feature is documented in the *IBM Informix Administrator's Guide*, *IBM Informix Administrator's Reference*, and *IBM Informix Performance Guide*.

Nonblocking Checkpoints

IBM Informix has replaced its checkpoint algorithm with a virtually nonblocking checkpoint algorithm. IBM Informix now allows applications to continue to process transactions while checkpoint processing is occurring. IBM Informix monitors the workload and past checkpoint performance and triggers checkpoints more frequently to avoid running out of critical resources, like the physical or logical log, to make sure transactions do not experience blocking during checkpoint processing. For applications that are sensitive to response times, the old method of using aggressive LRU flushing to reduce checkpoint quiescent times can be changed. LRU flushing can be less aggressive since transaction processing is not blocked during checkpoint processing. Making LRU flushing less aggressive can improve transaction performance.

This feature is documented in the *IBM Informix Administrator's Guide*, *IBM Informix Administrator's Reference*, and *IBM Informix Performance Guide*.

Performance Improvements for Enterprise Replication

Enterprise Replication has increased the degree of parallelism when applying transactions on target servers, resulting in better performance.

ON-Bar Performance Report

This new feature provides a report of ON-Bar backup and restore performance. You can set the frequency, in minutes, of the progress messages with the BAR_PROGRESS_FREQ configuration parameter. You can configure the report to contain sub-second timestamps for ON-Bar processing, as well as the transfer rates between ON-Bar and the storage manager, and between ON-Bar and the IDS instance. You can set the level of reporting to be written to the ON-Bar Activity log by using the new BAR_PERFORMANCE configuration parameter.

This feature is documented in the *IBM Informix Backup and Restore Guide*.

Transform Data during Backup and Restore

This features provides the option of specifying external programs, or filters, to transform data during backup and restore with both ON-Bar and ontape. You can use filters for compression or other data transformations. The backup filter reads the data to be backed up, transforms it, and then returns the transformed data to the backup utility. The restore filter receives the restored data from disk, transforms it back to its original state, and then passes the data to the restore utility. For example, if you want to compress your archive data, the backup filter receives the data, compresses it, and then backs up the compressed data. During the restore, the restore filter decompresses the data before it is restored to the database. You specify the filters with two new configuration parameters, BACKUP_FILTER and RESTORE_FILTER.

This feature is documented in the *IBM Informix Backup and Restore Guide*.

Improved Performance for Cooked Files with Direct I/O on UNIX

You can now improve the performance of cooked files used for dbspace chunks by using direct I/O. IBM Informix allows you to use either raw devices or cooked files for dbspace chunks. In general, cooked files are slower because of the additional overhead and buffering provided by the file system. Direct I/O bypasses the use of the file system buffers, and therefore is more efficient for reads and writes that go to disk. You specify direct I/O with the new DIRECT_IO configuration parameter. If your file system supports direct I/O for the page size used for the dbspace chunk and you use direct I/O, performance for cooked files can approach the performance of raw devices used for dbspace chunks.

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

Improved Performance of Online Index Creation

The CREATE INDEX ONLINE statement now never places an exclusive lock over the source table for the final 10% of the index build.

If concurrent transactions apply changes to the table faster than online index creation can apply those same changes to the index, the server now automatically places a transient shared lock on the table from time to time to reduce the amount of concurrent insert, update, or delete activity in the table to allow the index build to catch up with new changes. Because it is a shared lock, read activity in the table is not affected. Consequently, online index creation no longer results in long transactions or running out of space to store records of the ongoing changes that need to be applied to the index.

SQL Administration API

A new SQL administration API enables the DBSA to perform administrative tasks remotely by issuing SQL statements. The DBSA can accomplish administrative tasks, which in previous releases required administrative utilities of IBM Informix, by invoking new built-in **admin( )** or **task( )** functions with arguments that emulate command-line arguments to the corresponding utility. For example, the following SQL statement, which is equivalent to the **oncheck -ce** command, instructs the database server to check the extents: EXECUTE FUNCTION admin('check extents');

The **admin()** and **task()** functions are equivalent, except that their return values have different data types. The return values indicate the result of the operation.

Some options to these functions can also accomplish tasks for which no corresponding utility exists. The effects of calling the **admin( )** and **task( )** functions with the same argument list are equivalent, but their return values, which indicate the result of the operation, have different data types.

Information about the execution of SQL administration API functions is stored in the **command_history** table of a new **sysadmin** database of the IBM Informix instance.

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

Schedule Administrative Tasks

The new Scheduler allows you to manage and run scheduled maintenance, monitoring, and administration tasks at predefined times or as determined internally by the server. You can monitor activities (for example, space management) and create automatic corrective actions. Scheduler functions collect information and monitor and adjust the server, using an SQL-based administrative system and a set of tasks. A set of task properties, which define what needs to be collected or executed, control the Scheduler.

The task properties are stored in the **ph_task** table in the **sysadmin** database. Each row in this table is a separate task and each column is a task property. You can modify task properties, and you can set up new tasks by inserting rows into the table.

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

Monitor and Analyze Recent SQL Statements

You can now monitor the performance of recently executed SQL statements by configuring SQL statement tracing. This feature provides statistical information about each SQL statement executed on the system. The statistical information is stored in a circular buffer, which the DBA can resize. By default this features is turned off. The feature can be enabled for all users or just a specific set of users.

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

Dynamically Change Enterprise Replication Configuration Parameters and Environment Variables

You can add, change, and remove in-memory values for Enterprise Replication configuration parameters and environment variables while the server is running. This feature does not update the onconfig file; the values are only valid for the current Enterprise Replication session.

This feature is documented in the *Enterprise Replication Guide*.

Dynamically Rename Enterprise Replication Columns, Tables, and Databases

You can now rename a replicated column, table, or database while Enterprise Replication is active. Use the RENAME statement to rename a column, table, or database on every participant in the replicate.

This feature is documented in the *IBM Informix Enterprise Replication Guide*.

Truncate Replicated Tables

You can now use the TRUNCATE statement on replicated tables while replication is active. The truncate operation is useful prior to direct synchronization with target tables are significantly inconsistent.

This feature is documented in the *IBM Informix Enterprise Replication Guide*.

Improved Statistics Maintenance

IBM Informix now automatically collects index statistics, equivalent to the statistics gathered by UPDATE STATISTICS in LOW mode, when you create a B-tree index on a UDT column of an existing table or if you create a functional or VII index on a column of existing table. Statistics that are collected automatically by this feature are available to the query optimizer, removing the necessity of manually running UPDATE STATISTICS. When B-tree indexes are created, column statistics are collected on the first index column, equivalent to what UPDATE STATISTICS generates in HIGH mode, with a resolution is 1% for tables of fewer than a million rows, and 0.5% for larger tables.

You can now view statistics about completed queries in the new Query statistics section in SET EXPLAIN.

In explicit UPDATE STATISTICS operations in MEDIUM mode, a new SAMPLING SIZE option in the Resolution clause can specify the minimum number of rows to sample for column distributions.

You can use the SET EXPLAIN statement to designate a nondefault output file to capture the information that normally goes to the "sqexplain.out" file.

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

Installation Improvements on Windows Platforms

IBM Informix installation on Windows platforms is now accomplished using an industry-standard installation tool, which provides a shorter and easier installation using a substantially smaller footprint.

This feature is documented in the *IBM Informix Installation Guide for Windows* and *Quick Beginnings for IBM Informix Dynamic Server Express Edition*.

Session Configuration Routines

New built-in SPL procedures enable the Database Administrator to execute SQL and SPL statements automatically when a user connects to or disconnects from the database. These built-in procedures can be useful in setting the session environment or performing tasks like activating a role for users of information management applications whose code cannot easily be modified, or in automating operations that need to be performed after the application terminates.

If the DBA specifies the login ID of a user as the owner of a procedure whose name is **sysdbopen( )**, IBM Informix executes that procedure when the specified user connects to or disconnects from the database. If the DBA specifies PUBLIC as the owner, that routine is automatically executed when a user who is not the owner of any of these built-in session configuration procedures connects to the database. The **sysdbopen( )** routine is not invoked, however, when a user who is already connected to a database performs a distributed operation, such as a cross-database or cross-server query, that references an object in another database.

Similarly, another built-in procedure, **user.sysdbclose( )** or **public.sysdbclose( )**, if no **user.sysdbclose( )** is registered in the database for that user, is called automatically when the user closes the connection to the database.

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

Multiple Users for Administration Mode

The new administration mode enhances and replaces single-user mode as a way to temporarily restrict access to the database server to perform administrative tasks. Single-user mode allowed only the user **informix** or a member of the DBSA group to connect to the database server. The user **informix** or a DBSA can now dynamically give one or more specific users the ability to connect to the database server in administration mode. Administration mode is enabled using a new **onmode** command option, a new **oninit** command option, or the new ADMIN_MODE_USERS configuration parameter.

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

PHP-based OpenAdmin Tool for IBM Informix

A new PHP-based Web browser administration tool, the OpenAdmin Tool for IBM Informix, provides the ability to administer multiple database server instances from a single location.

OpenAdmin is an open-source program that you can download from the following Web site: http://www.openadmintool.com.

Named Parameters in a JDBC CallableStatement

A CallableStatement provides a way to call a stored procedure on the server from a Java program. This feature provides support for named parameters in a CallableStatement, which was introduced in the JDBC 3.0 specification. Using named parameters in a CallableStatement adds the convenience of being able to identify parameters by name instead of by ordinal position. If the stored procedure is unique, you can omit parameters that have default values and you can enter the parameters in any order. Named parameters are especially useful for calling stored procedures that have many arguments and some of those arguments have default values.

This feature is documented in the *IBM Informix JDBC Driver Programmer's Guide*.

Index Binary Data Types

The new Binary UDT DataBlade module module provides two new data types allow you to store binary-encoded strings, which can be indexed for quick retrieval. The binaryvar data type is a variable-length opaque type with a maximum length of 255 bytes. The binary18 data type is the same as the binaryvar data type except it holds a fixed value of 18 bytes. As part of a new DataBlade module, these data types come with string manipulation functions to validate the data types and bitwise operation functions that allow you to perform bitwise logical AND, OR, XOR, and NOT comparisons.

This feature is documented in the *IBM Informix Database Extensions User's Guide*.

Trigger Enhancements

Several new features expand the syntax and the functionality of triggers on tables and on views:

- You can now define multiple INSERT, DELETE, UPDATE, and SELECT triggers on a table and multiple INSTEAD OF triggers for the view.
- When a table, view, or column list has multiple triggers for a DML event type, IBM Informix executes all BEFORE triggered actions before the FOR EACH ROW actions, and executes all FOR EACH ROW actions before the AFTER actions.
- You can create SPL procedures which can refer to applicable OLD and NEW trigger correlated values. In this procedure you can access applicable OLD and NEW values and modify the NEW values: e.g. Using LET statements. From FOR EACH ROW trigger action, you can execute this SPL procedure [syntax: execute procedure foo() with trigger references].
- New Boolean operators (DELETING, INSERTING, SELECTING, and UPDATING) can be used in procedures executed from trigger action statements. These test whether the currently executing triggered action was triggered by the specified type of DML event and return a boolean value. The IF statement of SPL and the CASE expression of SQL can specify these operators as the condition in a trigger routine.

These features make it easier to incorporate IBM Informix triggers on tables and on views within a heterogeneous information management system where multiple applications need to share the table or view.

You can find examples about using multiple triggers in demo directory: **%INFORMIXDIR%\demo\dbaccess\demo_ud\cr_trig.sql**.

This feature is primarily documented in the *IBM Informix Guide to SQL: Syntax* and *IBM Informix Guide to SQL: Tutorial*.

Derived tables in the FROM Clause of Queries

The SELECT statement can now include syntax that complies with ISO/IEC 9075:1992, the SQL-92 standard, to specify full select subquery in the FROM clause as a data source for the query. These subqueries are called derived tables or table expressions, can be simple, UNION, or joined subqueries, including OUTER joins, and can include the ORDER BY clause. In addition, AS correlation specifications in the FROM clause can declare temporary names for columns within the query. Informix-extension syntax, such as the FUNCTION keyword with iterator functions or the TABLE (MULTISET (SELECT ...)) keywords for collection-derived tables, can now be replaced in the FROM clause by SQL-92 syntax. This feature expands the capability of IBM Informix to run without modification queries that are interoperable on other database servers that support industry-standard SQL syntax.

You can find examples about using derived tables in the demo directory: **%INFORMIXDIR%\demo\dbaccess\demo_ud\sel_sql99.sql**.

This feature is primarily documented in the *IBM Informix Guide to SQL: Syntax* and *IBM Informix Guide to SQL: Tutorial*.

Index Self-Join Query Plans

In earlier IBM Informix versions, queries of tables with composite indexes perform inefficiently if the ratio of duplicate values to the number of distinct values is much higher for the leading columns than for subsequent columns of the index. A new feature of the query optimizer supports a new type of index scan, called an index self-join path, that uses only subsets of the full range of a composite index. The table is logically joined to itself, and the more selective nonleading index keys are applied as index bound filters to each unique combination of the leading key values. By default, the optimizer considers this type of scan.

The optimizer also supports two new access-method directives, INDEX_SJ and AVOID_INDEX_SJ. The INDEX_SJ directive forces an index self-join path using the specified index, or choosing the least costly index in a list of indexes, even if data distribution statistics are not available for the leading index key columns. The AVOID_INDEX_SJ directive prevents a self-join path for the specified index or indexes. This feature can improve performance for queries of tables on which composite indexes are defined.

This feature is primarily documented in the *IBM Informix Guide to SQL: Syntax* and *IBM Informix Performance Guide*.

Optimizer Directives in ANSI-Compliant Joined Queries

Earlier IBM Informix versions supported optimizer directives in Informix-extension joined queries, but not in queries that used ANSI/ISO syntax to specify joins. For both inline directives and external directives, this release extends support in ANSI/ISO joined queries to the following classes of optimizer directives:

- Access-method directives (FULL, AVOID_FULL, INDEX, AVOID_INDEX, INDEX_SJ, AVOID_INDEX_SJ)
- Explain-mode directives (EXPLAIN, AVOID_EXECUTE)
- Optimization-goal directives (ALL_ROWS, FIRST_ROWS).

The join-order directive (ORDERED) is supported only in ANSI/ISO-compliant LEFT OUTER joins and INNER joins. Because of ordering requirements for OUTER joins, in ANSI-compliant joined queries that specify the RIGHT OUTER JOIN or FULL OUTER JOIN keywords, the ORDERED join-order directive is ignored, but it is listed under Directives Not Followed in the sqexplain.out file.

This feature does the not support the join-method directives (USE_NL, AVOID_NL, USE_HASH, AVOID_HASH, /BUILD, and /PROBE) in ANSI/ISO joined queries, except in cases where the optimizer rewrites the query so that it is no longer uses the ANSI/ISO syntax.

This feature is documented in the *IBM Informix Guide to SQL: Syntax*.

Deployment Wizard

The new Deployment Wizard allows you to perform a custom installation of IBM Informix by the components and features you do not currently wish to install. This enables database administrators and independent software vendors to minimize the disk space (footprint) required for a custom installation of IBM Informix. All installation methods (console, GUI, and silent) use the wizard to enforce dependencies between components and provide estimated total footprint of selected components prior to actual file loading. You can selectively install or uninstall components at any time. You can use the generated response file to automate future replication of the installation setup in other installation locations.

This feature is documented in the IBM Informix installation guides, including *Quick Beginnings for IBM Informix Dynamic Server Express Edition*.

Enhanced Concurrency with Committed Read Isolation

In Committed Read isolation level, exclusive row-level locks held by other sessions can cause SQL operations to fail when attempting to read data in the locked rows. This release introduces a new LAST COMMITTED keyword option to the SET ISOLATION COMMITTED READ statement to reduce the risk of locking conflicts when attempting to read a table. This new syntax instructs IBM Informix to return the most recently committed version of the rows, even if another concurrent session holds an exclusive row-level lock.

This behavior can be extended to the Dirty Read, Read Uncommitted, and Read Committed isolation levels by setting the new USELASTCOMMITTED configuration parameter or through new options to the SET ENVIRONMENT statement.

This feature supports B-tree indexes and functional indexes, but not R-tree indexes. It does not support tables that are being accessed by the DataBlade module, tables with columns of collection data types, tables created using a Virtual Table Interface, tables with page-level locking, tables with exclusive table-level locks, unlogged tables, or tables in databases with no transaction logging.

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

Enhanced Data Types and UDR Support in Cross-Server Distributed Operations

This release extends support for UDRs in cross-database and cross-server distributed operations to most contexts where a UDR is valid in the local database. In addition, external routines written in the C or Java languages are now valid in any distributed operation where an SPL routine is valid.

This release also extends the data types that are valid as parameters or return values of cross-server UDRs, which were formerly restricted to nonopaque built-in SQL data types, by supporting these additional data types:

- BOOLEAN
- LVARCHAR
- DISTINCT of nonopaque built-in types
- DISTINCT of BOOLEAN
- DISTINCT of LVARCHAR,
- DISTINCT of the DISTINCT types that are listed above

These data types can be returned by SPL, C, or Java language UDRs that use these data types as parameters or as return values, if the UDRs are defined in all the participating databases. Any implicit or explicit casts defined over these data types must be duplicated across all the participating IDS instances. The DISTINCT data types must have exactly the same data type hierarchy defined in all databases that participate in the distributed query.

This feature does not relax existing restrictions on other opaque and DISTINCT types or on large-object, serial, and collection data types in locally or remotely executed SPL routines or external routines.

This feature is primarily documented in the *IBM Informix Guide to SQL: Syntax*.

XML Publishing

You can now perform XML publishing with IBM Informix. Built-in functions let you transform results of an SQL query to XML for use in XML applications or in a heterogeneous database environment. Other built-in functions let you use XPATH expressions to extract elements and values from XML documents.

This feature is documented in the *XML User's Guide*.

Index Hierarchical Data

The node data type is part of the new Node DataBlade module which, with its supporting functions, gives you the ability to represent hierarchical data within the relational database. The advantage to this new data type is that it allows for searches within the hierarchy with a single SELECT statement using traditional operators without recursion. Represented as an ordinal number followed by either a single .0 or a set of ordinal numbers separated by dots, the node data type corresponds to a position in a tree structure similar to the way a books table of contents represents chapter, section, and subsection information.

This feature is primarily documented in the *IBM Informix Database Extensions User's Guide*.

Basic Text Search

The Basic Text Search DataBlade module allows you to search words and phrases in an unstructured document repository stored in a column of a table. The column can be a BLOB, CHAR, CLOB, LVARCHAR, NCHAR, NVARCHAR, or VARCHAR data type. Search strategies include single and multiple character wildcard searches, fuzzy and proximity searches, and AND, OR and NOT Boolean operations. This feature is included with the database server at no extra cost.

This feature is primarily documented in the *IBM Informix Database Extensions User's Guide*.

Improved Concurrency with Private Memory Caches for Virtual Processors

You can now configure a private memory cache for every CPU virtual processor to decrease the time of server memory allocation on large multiprocessor computers by using the VP_MEMORY_CACHE_KB configuration parameter.

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

Web Feature Service for Geospatial Data

The new Web Feature Service DataBlade Module implements an Open Geospatial Consortium Web Feature Service (OGC WFS) in IBM Informix to act as a presentation layer for the Spatial and Geodetic DataBlade module. The OGC WFS interface allows requests for geographical features across the web using platform-independent calls. The XML-based GML (Geography Markup Language) is used as the encoding for transporting the geographic features.

This feature is compatible with the 8.21 version of the Spatial DataBlade module. See your release notes for the Geodetic DataBlade module for compatibility information.

This feature is primarily documented in the *IBM Informix Database Extensions User's Guide* and the *IBM Informix Spatial DataBlade Module User's Guide*.

Support for Data Server Clients with DRDA

You can use IBM Data Server JDBC Driver and the IBM Data Server .NET Provider to communicate with IDS as well as DB2. IBM Informix now supports DRDA, the communications protocol used by DB2. As a result, application developers can create a solution using this API and enable their customers to deploy on the IBM Data server they prefer to use.

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

Statement Labels, GOTO, and LOOP Statements in the SPL language

The new support for statement labels and the GOTO and LOOP statements provide greater flexibility in iterating and in exiting from statement loops in SPL routines. They also facilitate migration to IBM Informix of routines written in the procedural languages of other database servers that support GOTO and LOOP statement syntax. This release introduces the new SPL statements and constructs for statement loops in SPL routines:

- In the lexicographic sequence of statements in an SPL routine, a statement label can precede an executable statement, a program block, or a LOOP statement within the same SPL routine to which the GOTO label , END LOOP label, or EXIT label statement can transfer control.
- The GOTO label statement can unconditionally exit from a loop and transfer control to the executable statement or statement block that follows the specified statement label.
- The LOOP statement executes a statement block for an unspecified number of iterations. This can be within a WHILE condition LOOP or a FOR condition LOOP statement, or independent of any FOR or WHILE statement. LOOP statements can also be nested.
- To avoid infinite iterations of LOOP statements, new EXIT and CONTINUE options to the IF statement are valid within LOOP statements.
- The EXIT label WHEN condition statement can terminate a loop that has a label.
- The EXIT label statement can terminate a loop that has a label.

This feature is documented in the *IBM Informix Guide to SQL: Syntax*.

New SQL Functions

IDS now supports the following new built-in SQL functions to perform common mathematical, casting, and bitmap operations, and for manipulating character string, date, and datetime values:

- ADD_MONTHS()
- ASCII()
- BITAND()

- BITANDNOT()
- BITNOT()
- BITOR()
- BITXOR()
- CEIL()
- FLOOR()
- FORMAT_UNITS()
- LAST_DAY()
- LTRIM()
- MONTHS_BETWEEN ()
- NEXT_DAY ()
- NULLIF()
- POWER()
- ROUND()
- RTRIM()
- TO_CHAR()
- TO_NUMBER()
- TRUNC()

These built-in SQL functions can simplify the migration to IBM Informix of applications that have been developed for other database servers.

These functions are documented in the *IBM Informix Guide to SQL: Syntax*.

Automatic Re-Compilation of Prepared Statements

In this release, when you execute a prepared statement, IBM Informix detects changes to underlying objects and re-prepares (recompiles) the statement if needed. After a statement has been prepared (compiled) by the application, the statement objects -- tables, indices, statistics -- can be altered. Some operations like ALTER TABLE require no cursor to be open, while others like CREATE INDEX ONLINE allow cursors on the table. To account for these changes, the statement has to be re-prepared. In previous releases, when you tried to open a statement that's using an altered object, IBM Informix raised -710 error so the client program can catch the exception, re-prepare the statement before proceeding.

After a DDL operation modifies the schema of a database table, the database server now automatically performs the following actions that previously had to be performed manually before executing any SPL routines or prepared objects that reference the modified table:

- The database server automatically issues the UPDATE STATISTICS statement to recalculate routine statistics for all SPL routines that reference the table.
- The database server automatically issues the PREPARE statement to update any prepared objects that reference the table.

Cursors associated with automatically updated routines or prepared statements can be used in dynamic SQL operations without the user manually issuing the PREPARE or UPDATE STATISTICS statement. For schema changes where automatic recompilation is not desired, the legacy behavior can be restored with the AUTO_REPREPARE configuration parameter or within a session by resetting a new SET ENVIRONMENT option.

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

Label-Based Access Control

This feature implements Mandatory Access Control, a security requirement that the United States Federal government and public sector agencies in other countries impose on some information management systems that handle sensitive or classified information. This feature provides mechanisms by which a hierarchy of security labels can be defined and assigned to database objects and to users of the database. In a database that implements label-based access control (LBAC), there are two fundamental access rules:

- Users cannot have Read access to database objects that are at a higher security level than the user.
- Users cannot have Write access to database objects that are at a lower security level than the user.

This feature is supported by various enhancements to the SQL syntax of IBM Informix, and to the security of existing IBM Informix features, including secure auditing, high-availability data replication, backup and restore features, and administrative utilities:

- The database server administrator can grant a new built-in role, DBSECADM, to a database security administrator.
- The DBSECADM can issue new data definition language statements to create, drop, alter, and rename security policies, security labels, and security label components.
- The DBSECADM can grant exemptions to individual users to bypass LBAC access rules.
- Each untyped permanent database table can receive LBAC protection through a security label.
- A new built-in data type can store row labels in protected tables.
- The system catalog is enhanced to store LBAC information.

This feature is documented in the *IBM Informix Guide to SQL: Reference*, *IBM Informix Guide to SQL: Syntax*, and *IBM Informix Security Guide*.

New Features in Version 10.00 of IBM Informix

New Features in Version 10.00.xC4

IBM Informix, 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 *IBM Informix 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 *IBM Informix 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 *IBM Informix 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 *IBM Informix 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 *IBM Informix Administrator's Reference*.

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 *IBM Informix 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, 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 Informix 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 IBM Informix 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 IBM Informix 10.00.xC3 to an earlier release that does not support the implementation of the ANSI-compliant syntax such queries might show reduced performance because the Informix instance from which the query originates performs the joins locally.

Transaction Support for XA-Compliant External Data Sources

The IBM Informix 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 Function Reference* and the *IBM Informix DataBlade API Programmer's Guide*.

MQ DataBlade module

The MQ DataBlade module provides mechanisms for data exchange between IBM Informix 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 Administrator's Guide* and *IBM Informix 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

ForIBM Informix version 10.00.xC1, table-level restore supported all built-in, BOOLEAN, and LVARCHAR data types for physical and logical restore. For IBM Informix 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 *IBM Informix 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 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 *IBM Informix Enterprise Replication Guide* documentation notes.

CSDK Included in IBM Informix Installation

You can install IBM Informix Client Software Development Kit (Client SDK) with the IBM Informix installation program. CSDK version 2.90.xC3 is included as part of the IBM Informix 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 Installation Guide for UNIX, Linux, and Mac OS X*.

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 IBM Informix 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 DBSA 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 value
1

takes effect
when the database server is shut down and restarted

New Features in Version 10.00.xC1

IBM Informix, 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

Security Enhancements

IBM Informix 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 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 Administrator's Guide*.

Restricting Registration of External Routines

The database server administrator (DBSA) can use the new built-in role called EXTEND to specify which users can register UDRs that include the EXTERNAL NAME clause. User-defined routines written in the C or Java languages use shared-object files that are external to the database server and that could potentially contain harmful code. Similarly, Java UDRs are stored in class or jar files outside the DBMS. 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 Administrator's Guide*, *IBM Informix Administrator's Reference*, and *IBM Informix Guide to SQL: Syntax*.

Preventing Denial-of-Service Attacks

Informix 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 Administrator's Guide* and *IBM Informix 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 might want to specify a nondefault page size if you want a longer key length than is available for the default page size. The root dbspace is the default page size. If you want to specify a page size, the size must be an integral multiple of the default page size, not greater than 16 kilobytes.

You can also use the new BUFFERPOOL configuration parameter to create a buffer pool that corresponds to the page size of the dbspace.

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

Defining Buffer Pools:

You can use the new BUFFERPOOL configuration parameter or the **onparams** utility to define a buffer pool for pages that correspond to each unique page size in use by your dbspaces. When you use the BUFFERPOOL configuration parameter or the **onparams** utility to define a buffer pool, you specify information about the buffer pool including its size, the number of LRUS in the buffer pool, the number of buffers in the buffer pool, and lru_min_dirty and lru_max_dirty values.

The BUFFERS, LRUS, LRU_MAX_DIRTY, and LRU_MIN_DIRTY configuration parameters are no longer used. Information that was specified with the BUFFERS, LRUS, LRU_MAX_DIRTY, and LRU_MIN_DIRTY configuration parameters prior to Version 10.0 is now specified using the BUFFERPOOL configuration parameter or the onparams utility. Information you enter using the BUFFERPOOL configuration parameter or onparams utility supersedes any information previously specified with the deprecated parameters.

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

Managing the tblspace tblspace: You have enhanced flexibility in managing the tblspace **tblspace**. The tblspace **tblspace** is a set of pages that describe the location and structure of all tblspaces in a given dbspace. Use the **onspaces** utility to move or drop the chunk containing the tblspace **tblspace**.

You can also specify the size of the first extent and of subsequent extents when dbspaces are created:

- Use the TBLTBLFIRST and TBLTBLNEXT configuration parameters to specify the size of the extents in the root dbspace. You must set these configuration parameters before creating the root dbspace.
- Use the onspaces utility to specify the size of nonroot dbspaces during creation.

This feature allows you to reduce the number of tblspace **tblspace** extents and reduce the frequency of cases where those extents are placed in nonprimary chunks.

These features are primarily documented in the *IBM Informix Administrator's Guide* and *IBM Informix Administrator's Reference*.

Administering the Database Server in Single-User Mode:

As a database administrator, you can use a new single-user mode that is an intermediary mode between quiescent mode and online mode. While the server is in single-user mode, new connections are accepted only for user **informix**. Use this mode to perform any maintenance task, including tasks requiring the execution of SQL and DDL statements, when no other users are connected to the database server. Administrators can also perform all other functions available in online mode. You can set this mode using the **oninit** or **onmode** utilities, or the IBM Informix Server Administrator (ISA).

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

Managing Database Permissions Through Default Roles:

You can create a default role and assign that role to individual users or to PUBLIC on a per-database level. Each user who is assigned to a default role receives the privileges of that role as well as whatever other privileges are granted to the user individually. The syntax of the GRANT, REVOKE, and SET ROLE statements support this feature. This feature allows client applications to reset permissions (to the default role) of the user running the application, instead of requiring the DBA to reset permissions manually.

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

Renaming Dbspaces:

If you are user **informix** or have DBA privileges and the database server is in quiescent mode, you can rename a previously defined standard dbspace. You might want to rename standard dbspaces if you are reorganizing data in an existing dbspace and see a need to change the dbspace name. The rename dbspace operation only changes the dbspace name; it does not reorganize data. The database server automatically updates the system catalog so that database objects residing in a renamed dbspace are registered with the new name, but subsequent DDL statements referencing that dbspace must specify the new name, rather than the old name.

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

Specifying Which Events Trigger the Alarm Program:

You can use the new alarm configuration parameter, ALRM_ALL_EVENTS, to specify whether the event alarm program operates for all events that are logged in the MSGPATH or only specified noteworthy events.

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

Specifying Shared Memory Size Greater Than 4 GB:

You can now specify that segments for shared memory be created as large as your operating system platform or the SHMMAX parameter allows.

Restriction: ON-Monitor does not support Shared Memory Size greater than 4 GB. Therefore, the size specified for SHMVIRSIZE, SHMADD and SHMTOTAL in the "Shared-Memory screen" cannot be greater than 4 GB.

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

Setting Up HDR with External Backup and Restore:

You can set up High-Availability Data Replication using standard ON–Bar or **ontape** commands for external backup and restore. Doing so can significantly reduce initial setup time.

This feature is primarily documented in the *IBM Informix Administrator's Guide* and *IBM Informix Backup and Restore Guide*.

Replicating Indexes to HDR Secondary Servers:

You can replicate an index to a secondary server in an HDR pair that contains a corrupted index. Replicating an index does not require a lock on the table. You can manually replicate an index when you detect a corrupt index with the **onmode -d index** command. Alternatively, you can configure automatic index replication with the **onmode -d idxauto** command or the DRIDXAUTO configuration parameter. This feature increases the availability of the HDR primary server because replicating an index is quicker than dropping and then rebuilding the index on the primary server.

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

Automating HDR Failover:

You can automate switching servers for High-Availability Data Replication if the primary server fails by using the DRAUTO configuration parameter. If DRAUTO is set to either RETAIN_TYPE or REVERSE_TYPE, the secondary database server switches to type standard automatically when an HDR failure is detected. If DRAUTO is set to RETAIN_TYPE, the original secondary database server switches back to type secondary when the HDR connection is restored. If DRAUTO is set to REVERSE_TYPE, the original secondary database server switches to type primary when the HDR connection is restored, and the original primary switches to type secondary.

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

Determining Version Information:

You can now use the new `-version` option with all server utilities to provide detailed information on the build operating system, build number, and build date. The `-version` option provides more information than the existing `-V` option. This feature enables DBAs and IBM Technical Support personnel to track version- and build-machine information, access documents before installing a product, be sure they are ready to install a product, and diagnose problems.

This feature is primarily documented in the *IBM Informix Administrator's Reference*.

IBM Informix Server Administrator Enhancements:

IBM Informix Server Administrator (ISA) supports the following new features:

- **Single-user mode.** See "Administering the Database Server in Single-User Mode" on page 2-67.
- **Enterprise Replication templates.** See "Replicate Templates" on page 2-71.

IPv6 Format for IP Addresses Support:

You can use the IPv6 format for IP addresses with Informix. The IBM Informix JDBC Driver, Version 3.0, supporting the JDK 1.4, is IPv6 aware. That is, the code that parses the connection URL can handle the longer (128-bit mode) IPv6 addresses (as well as IPv4 format). This IP address can be an IPv6 literal.

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

Performance Enhancements

The performance enhancements include improved query performance and recovery time. In addition to the topics discussed below, enhancements have been made to improve performance in the following areas:

- XA transactions
- Nested ANSI-compliant left-outer joins
- Subqueries
- Full-outer joins

Allocating Memory for Non-PDQ Queries:

This feature was first available in 9.40.xC4. You can specify how much memory is allocated to non-PDQ queries. The default of 128K can be insufficient for queries that specify ORDER BY, GROUP BY, hash joins, or other memory-intensive options. Use the new configuration parameter, DS_NONPDQ_QUERY_MEM, to specify more memory than the 128K that is allocated to non-PDQ queries by default.

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

Storing and Applying External Optimizer Directives:

You can create, save, and reuse external optimizer directives. In previous releases of Informix, external optimizer directives existed as text strings within DML statements, but were not stored as separate entities. External optimizer directives are useful when it is not feasible to rewrite a query for a short-term solution to a problem, for example, when a query starts to perform poorly. In this release, the DBA (as user **informix**) can create external optimizer directives and apply them to subsequent queries, without changing existing application code. This feature is implemented as a new SQL statement, SAVE EXTERNAL DIRECTIVES, to create and register external optimizer directives in a new **sysdirectives** table of the system catalog. Use the new **IFX_EXTDIRECTIVES** environment variable and the EXT_DIRECTIVES configuration parameter to enable this feature.

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

Storing Multiple Table or Index Fragments in a Single Dbspace:

You can store multiple fragments of the same table or index in a single dbspace, reducing the total number of dbspaces needed for a fragmented table. You must specify a name for each fragment that you want to store in the same dbspace. Storing multiple table or index fragments in a single dbspace improves query performance over storing each fragmented expression in a different dbspace. This feature improves performance and simplifies management of dbspaces.

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

Recovering Quickly with Fuzzy Checkpoints:

You can set two new configuration parameters (FAST_RESTART_PHYSLOG and FAST_RESTART_CKPT_FUZZYLOG) to reduce the time required for database server recovery. This supports high availability by improving recovery performance when using fuzzy checkpoints.

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

Dynamically Setting the OPTCOMPIND Environment Variable: This feature was first available with Version 9.40.UC3. You can use the new SET ENVIRONMENT OPTCOMPIND statement to set the value of the **OPTCOMPIND** environment variable dynamically for the current session. For example, you might want to change the value for different kinds of queries. For a DSS query, you should set the value of **OPTCOMPIND** to 2 or 1, and you should be sure the isolation level is not set to REPEATABLE READ. For an OLTP query, you can set the value to 0 or 1 with the isolation level not set to REPEATABLE READ. The value that you enter using this statement takes precedence over the current setting specified in the ONCONFIG file. The default setting of the **OPTCOMPIND** environment variable is restored when your current session terminates. No other user sessions are affected by SET ENVIRONMENT OPTCOMPIND statements that you execute.

This feature is primarily documented in the *IBM Informix Guide to SQL: Syntax* and *IBM Informix Performance Guide*.

SQL Enhancements

The SQL enhancement improves database availability.

Creating and Dropping Indexes without Locking Tables

The SQL syntax of CREATE INDEX and DROP INDEX now supports the new ONLINE keyword. When you use the ONLINE keyword, DDL operations execute without applying an exclusive lock to the table on which the specified index is defined. If you use this syntax to create an index on a table that other users are accessing, the index is not available until no user is updating the table. After you issue the new syntax to drop an index, no one can reference the index, but concurrent DML operations can use the index until they terminate. Dropping the index is deferred until no user is using the index. This feature maintains the availability of the table within a production environment after an existing index has ceased to be efficient.

This feature is primarily documented in the *IBM Informix Guide to SQL: Syntax*.

Enterprise Replication Enhancements

The Enterprise Replication enhancements ease administration, improve data integrity, and allow additional SQL operations. These enhancements are documented in the *IBM Informix Enterprise Replication Guide*.

Master Replicates:

A master replicate is a replicate that uses a specified participant as a master against which all other participants are tested for consistency.

Creating a replicate as a master replicate provides several advantages:

- Ensures data integrity by verifying that all participants in the replicate have table and replicated column attributes that match the master replicate definition. Verification occurs when the replicate is defined and when the replicated is started, thus avoiding runtime errors.
- Provides automatic table generation on participants that do not already contain the table specified in the master replicate.
- Allows alter operations on the replicated tables. For more information, see "Altering Replicated Tables" on page 2-72.

Replicate Templates:

Enterprise Replication has a replicate template option that greatly simplifies setting up the replication system. Replication templates contain schema information about a database, a group of tables, column attributes, and the primary keys that identify rows. You first define the template by specifying schema information and then apply the template on the database servers within the replication domain. If the tables and primary keys do not exist on the server during the realization of the template, then the tables and primary keys are created according to the template definitions. Table attributes are verified during the realization of the template to ensure that proper replication can be performed on that database. Replicates created as part of the realization of a replicate template are created as mastered replicates and grouped as a replicate set.

You can create, realize, view, and delete templates using the Enterprise Replication command line syntax or IBM Informix Server Administrator (ISA).

Performing Synchronization and Repair:

Enterprise Replication can perform an initial synchronization on data to be replicated as well as repair a system in which data may have failed to be applied for any reason. Initial synchronization is performed at the startup of a new replicate or when a new participant is added to an existing replicate. Repair jobs reconciles the differences when the normal replication process has failed. Repair processes can be optimized to only compare rows found in ATS or RIS files: if the row still exists in the source, it is re-replicated; if it does not, then it is deleted on the target.

Altering Replicated Tables:

Beginning with version 10.00 you can alter a replicated table in any of these ways:

- Add or drop UNIQUE, DISTINCT or FOREIGN KEY constraints
- Alter next extent size
- Alter the locking granularity of a table.
- Add or drop default values
- Add or drop SQL checks
- Alter serial columns
- Add or drop fragments (new in 10.00)
- Attach or detach fragments (new in 10.00)
- Add, modify, or drop columns (new in 10.00)
- Create a clustered index or recluster an existing index (new in 10.00)

Before attaching fragments place the table in alter mode. Alter mode is a new state of a replicated table. In this mode only DDL and select operations are allowed. No insert, update, or delete operations are allowed in alter mode. After attaching the fragments, unset alter mode.

When performing any sort of alter other than attaching fragments, the database server automatically sets alter mode before altering the table and unsets it after the table is altered.

To set/unset the alter mode on a table, use the CDR CLI interface.

ALTER TABLE and ALTER FRAGMENT statements are allowed only on master replicates.

Remastering:

An existing replicate can be redefined by “remastering the replicate”. Remastering allows you to change what columns are in a replicate definition without interrupting the replication. Also an existing nonmaster replicate can be converted to a master replicate using remastering.

Detecting Event Alarms with the Event Alarm Program:

Enterprise Replication event alarms are now detected uniquely by the event alarm program. You can specify what actions are triggered by specific Enterprise Replication event alarms.

Suppressing DataSync Warning and Error Messages:

You can selectively suppress any DataSync warning or error message that the ATS or RIS files contain with the CDR_SUPPRESS_ATSRISWARN configuration parameter.

Backup and Restore Enhancements

The backup and restore enhancements improve performance and debugging. These enhancements are documented in the *IBM Informix Backup and Restore Guide*.

Performing Point-In-Time Restores of Tables with Archecker:

You can recover specific tables from an archive using the **archecker** utility. Tables can be restored up to a specific point in time. This allows you to restore specific pieces of data without having to perform a lengthy restore of the entire archive. Data can be restored without restoring the full database server on another instance. To perform a table level restore, **archecker** supports a command file which uses an SQL-like syntax to specify the source and destination table schemas.

View Logical Logs Backed Up by ON-Bar:

You can now view the logical-logs backed up by ON-Bar, similar to using the **onlog** utility to view the logical-logs backed up by the **ontape** utility. If you need to perform a restore, you can view the old logical-log files that were backed up by ON-Bar to find out the exact problem.

Enhanced Debugging for ON-Bar:

You can now change the ON-Bar debugging level while ON-Bar is running. You can save a large amount of time and disk space by setting high debugging levels only when you need them. You set the debugging level with the BAR_DEBUG configuration parameter. The value of BAR_DEBUG ranges from 0-9, with 0 being no debugging information and 9 being the most detailed debugging information. You can change the value of BAR_DEBUG as frequently as you want. Similar to the **onstat -m** command, you can use ON-Bar to print the recent ON-Bar activity from the **onbar** activity log file.

The ontape Utility Can Use Standard I/O:

You can now specify that the **ontape** utility uses standard I/O instead of a tape device or disk file. During a backup, **ontape** writes the data to stdout (standard output). During a restore, **ontape** reads data from stdin (standard input). Specifying stdout or stdin allows **ontape** to use pipes (an OS provided buffer mechanism to connect separate programs to a data stream) for archives and restores. Using pipes, the data can be processed by other programs without requiring that the data be saved in files or tape devices. For example, you can use compression to save media space, use cloning to duplicate the archive for safety reasons, or restore the data onto another server instance. This feature is especially efficient for setting up High-Data Availability Replication by restoring the data to the secondary server while skipping the intermediary step of saving the data to a file or disk.

External Backup and Restore Using ontape:

You can use the **ontape** utility to perform external backup and restore procedures.

Storage Enhancements

The storage enhancements improve ease of use.

Using Long Identifiers with the High-Performance Loader

The High-Performance Loader utilities **onpload** and **onpladm** include support for long object names up to 128 characters, but the **ipload** GUI does not. If you use long database, table or column names and create jobs using **onpladm**, you cannot run these jobs using **ipload**. For **ipload**, database, table and column names cannot exceed 18 characters.

This feature is documented in the documentation notes for the*IBM Informix High-Performance Loader User's Guide*.

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 Informix 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 Remote 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, MERGE, 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 Installation Guide for UNIX, Linux, and Mac OS X*.

IBM Software Electronic Licensing

IBM Informix, 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 Informix:

- Release notes
- Machine notes
- Documentation notes
- Informix installation guide (in PDF format)

New Installation Program on UNIX and Linux

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

New Features in Version 9.4

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

- "Security Enhancement" on page 2-76
- "Database Server Usability Enhancements" on page 2-76
- "Performance Enhancements" on page 2-78
- "Enterprise Replication Enhancements" on page 2-79
- "Extensibility Enhancements" on page 2-81

- "SQL Enhancements" on page 2-82
- "GLS Enhancements" on page 2-86
- "Reliability, Availability, and Supportability Features" on page 2-87
- "DataBlade API Enhancements" on page 2-87
- "High-Performance Loader Enhancements" on page 2-89
- "Backup and Restore Enhancements" on page 2-89
- "Installation Enhancements" on page 2-89

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

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

This option provides complete data encryption with a standard cryptography 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 Administrator's Guide*.

Enterprise Replication implements encryption with configuration parameters instead of the ENCCSM. For more information, see "Enterprise Replication Security" on page 2-79.

Database Server Usability Enhancements

Informix Version 9.4 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 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 Administrator's Guide*. For more information on setting event alarms, see the *IBM Informix 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 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 2-84)
- The **onspaces** utility
- The **ontape** utility (see "Backup and Restore Enhancements" on page 2-89)
- 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 "Backup and Restore Enhancements" on page 2-89)
- 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 2-89.

In previous releases, the user was required to specify a nonzero 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 Administrator's Reference*.

Chunk Reserve Pages in Non-Root Chunks

Chunk reserve pages are stored in the root chunk. In previous releases of Informix, 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 nonroot chunks in the root dbspace.

For more information, see the *IBM Informix 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 Administrator's Guide*. For more information on the PLOG_OVERFLOW_PATH configuration parameter, see the *IBM Informix 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 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 Informix.

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 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 Performance Guide*. For information on how to configure the B-tree scanner with the **onstat -C** command, see the *IBM Informix 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 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 Performance Guide*. For more information on setting these configuration parameters, see the *IBM Informix Administrator's Reference*.

Enterprise Replication Enhancements

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

All Enterprise Replication features are documented in the *IBM Informix 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 2-80.

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. (Informix 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 that the database server 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 renegotiated.

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 Informix.

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 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 2-84.

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 2-81, the following additional changes to the IBM Informix dialect of the Structured Query Language (SQL) have been implemented in Version 9.4 of Informix.

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 2-81, 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 Informix 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 Informix 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 Informix 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 (SLVs) 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 DDL 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, 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 2-79 describes additional new environment variables that can affect Enterprise Replication.

GLS Enhancements

Informix 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 2-84.

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 Informix.

For more information on these features, see the *IBM Informix 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	Publication
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	Publication
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	Publication
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	Publication
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	Publication
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 Installation Guide for UNIX, Linux, and Mac OS X*.

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 Installation Guide for UNIX, Linux, and Mac OS X*.

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 Installation Guide for UNIX, Linux, and Mac OS X* and the *IBM Informix Installation Guide for Windows*.

New Features in Version 9.3

The new features for Informix, 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 Installation Guide for UNIX, Linux, and Mac OS X*.

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 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 the **onconfig.std** file, 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 Informix 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	Publication
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	Publication
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	Publication
Determine the cardinality of a collection.	*IBM Informix DataBlade API Programmer's Guide*
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	Publication
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 DataBlade Modules

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	Publication
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*.

Informix, 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 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, SERIAL8, and BIGSERIAL 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 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 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 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 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	Publication
Prepare tables for conflict resolution.	*IBM Informix 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 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 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	Publication
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 a 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	Publication
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	Publication
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 2-102.

JVM 1.3 Support in J/Foundation

Informix 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	Publication
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	Publication
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	Publication
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 Informix, 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*.

New Features in Informix, Version 9.21

These features were introduced in IBM Informix, 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	**Publication**
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 Informix. These tables are also called raw tables. Raw tables do not support primary constraints, unique constraints, and rollback. However, these tables can be indexed and updated. You can create either a standard or raw table and change tables from one type to another.

Task	**Publication**
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.

The **onbar -b -l** command replaces **onbar -l**, which has been deprecated.

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	Publication
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**	*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

Informix 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 Informix.

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. Informix 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

Informix supports the following GLS features:

- CLIENT_LOCALE, DBCENTURY, 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

Informix supports the JVM_MAX_HEAP_SIZE, JAR_TEMP_PATH, and JAVA_COMPILER environment variables.

Partial Support for Variable-Length Opaque-Types

You can now write UDRs and DataBlade modules in Java.

Informix 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 publications.

Task	Publication
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 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 to support greatly increased numbers of client connections. Informix 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: Informix MaxConnect and the *IBM Informix MaxConnect User's Guide* ship separately from IBM Informix, 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 Informix MaxConnect with two new network protocols: **ontliimc** and **onsocimc**.
- New utility options to monitor Informix MaxConnect
  - **onstat -g imc**
  - **imcadmin**
  - **ISA options**
- New environment variables for Informix MaxConnect
  - **IMCADMIN**
  - **IMCCONFIG**
  - **IMCSERVER**

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

Chapter 3. Using Existing Informix Features

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

Dynamic Scalable Architecture

Informix 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 3-12.

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 Informix, 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 publications, 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
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)
- Informix 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 1-7.

For information about client/server configurations that the database server supports, see the *IBM Informix Administrator's Guide*.

High Performance

Informix achieves high performance through the following mechanisms:

- Memory management
- Fragmentation
- Parallelization
- Query optimization

Memory Management

Informix 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 2-99. For information on transaction logging, see the *IBM Informix Administrator's Guide*.

Using NFS-Mounted Directories

An Informix storage space can reside on an NFS-mounted file system using regular operating-system files.

Fragmentation

Informix 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.

Informix 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 these 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

Informix 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 Clusters
- 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* recreates 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:

- Backing up and restoring storage spaces and logical logs.
- Performing point-in-time restores.
- Starting continuous logical-log backups.
- Verifying backups with the **archecker** utility.
- Performing 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.
- Renaming chunks to different path names 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 tracks backed-up data through a data life cycle that the database or system administrator determines and also manages storage devices and storage volumes.

For information about ISM, see the *IBM Informix Storage Manager Administrator's Guide*.

The archecker Utility

When you use the **onbar -v** command to verify **ON–Bar** backups, it calls the **archecker** utility.

Fast Recovery

Fast recovery is an automatic procedure that restores the database server to a consistent state after it goes offline under uncontrolled conditions. Fast recovery also rolls forward all committed transactions since the last checkpoint and rolls back any uncommitted transactions.

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

For information about fast recovery, see the *IBM Informix Administrator's Guide*.

Mirroring

When you use disk mirroring, the database server writes each piece of data to two locations. *Mirroring* is a strategy that pairs a *primary chunk* of one storage space with an equal-sized *mirrored chunk*. Every write to the primary chunk is automatically accompanied by an identical write to the mirrored chunk. If a failure occurs on the primary chunk, mirroring lets you read from and write to the mirrored chunk until you can recover the primary chunk, all without interrupting user access to data.

It is recommended that you mirror the following data:

- Root dbspace
- Dbspaces that contain the physical log and logical-log files
- Frequently queried data

For information about mirroring, see the *IBM Informix Administrator's Guide*.

Data Replication

Data replication generates and manages multiple copies of data at one or more sites, which allows an enterprise to share corporate data throughout its organization. Data replication provides a backup system in case of a catastrophic failure.

High-Availability Clusters

IBM Informix provides several high-availability cluster configuration options. A high-availability cluster consists of a primary server and one or more secondary servers on which data from the primary server is replicated. Data replication provides a way to duplicate database objects at more than one distinct site. Dynamic Server provides several secondary server configuration options, including:

- High-availability data replication (HDR) secondary server provides synchronous data replication for Dynamic Server. Use an HDR secondary server if you require a hot standby. Configuring an HDR secondary server provides a way to maintain a backup copy of the entire database server that applications can access quickly in the event of a catastrophic failure of the primary server.
- Shared-Disk (SD) secondary server is a server that shares disk space with a primary server. The primary server has write access to a disk or disk array, while all SD secondary servers have updatable secondary access. An SD secondary server does not maintain a copy of the physical database on its own disk space; rather, it shares disks with the primary server.

- Remote Standalone (RS) secondary server is a server that is updated asynchronously from the primary server. RS secondary servers can be geographically distant from the primary server, serving as remote backup servers in disaster-recovery scenarios. Each RS secondary server maintains a complete copy of the database, with updates transmitted asynchronously from the primary server over secure network connections.

For information about HDR secondary servers, RS secondary servers, and SD secondary servers, see the *IBM Informix Administrator's Guide*

Any of the above configurations can be combined with Enterprise Replication. For more information, see the *IBM Informix Enterprise Replication Guide*.

Enterprise Replication

Enterprise Replication captures transactions to be replicated throughout the enterprise. On the source database server, Enterprise Replication reads the logical log and transmits each transaction to the target database servers. At each target database server, Enterprise Replication receives and applies each transaction to the appropriate databases and tables. Enterprise Replication can be combined with other data replication solutions.

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

Types of Data That You Can Replicate

Table 3-1 shows the types of data that you can replicate with data replication or Enterprise Replication.

Table 3-1. Data Types that high-availability clusters and ER Replicate

Data Type	Cluster Support	ER Support
Atomic data types such as numeric, character, varying character, time, Boolean	Yes	Yes
Simple large objects in dbspaces	Yes	Yes
Simple large objects in blobspaces	No	Yes
User-defined data types	Yes	Yes [1]
DataBlade types (text, image, video, web, and geodetic)	Yes	Yes
Smart large objects	Yes	Yes [2]

Notes:
1. To replicate user-defined data types, the required **streamwrite()** and **streamread()** functions must exist. For information on writing and registering support functions, see the section on writing Enterprise Replication stream support functions in the *IBM Informix DataBlade API Programmer's Guide*.
2. For information on restrictions for replicating smart large objects, see the *IBM Informix Enterprise Replication Guide*.

Database Server Security

The *IBM Informix Security Guide* contains information that your database administrator needs to know.

Database security features include the following types of tasks:

- Secure server utilities
- Encrypt data across the network
- Encrypt column-level data
- Secure connections
- Control user privileges
- Control user access to data

Auditing Database Events

You can audit database events by tracking activities that users perform on particular objects at distinct times. You can use this information to monitor database activity for suspicious use, detect unauthorized access attempts, assess potential security damage, unscrupulous users, and provide evidence of database server abuse.

For information on auditing, see the *IBM Informix Security Guide*.

Informix RDBMS Features

This section discusses the database components and extensibility features.

Structured Query Language (SQL)

You can use SQL statements to retrieve, insert, update, and delete data from a database. To retrieve data from a database, you perform a query, which is a SELECT statement that specifies the rows and columns to be retrieved from the database.

You can write programs that exchange data with the database server. You can also write programs that take data from any source in any format, prepare it, and insert it into the database.

Use Informix ESQL/C to embed SQL statements directly into a C program. DB–Access lets you execute SQL statements interactively. Use JDBC to embed SQL statements directly into a Java program.

For information about database management, see the *IBM Informix Database Design and Implementation Guide*. For information about how to create and use SQL, see the *IBM Informix Guide to SQL: Tutorial* and the *IBM Informix Guide to SQL: Syntax*. For information about embedded SQL, see the *IBM Informix ESQL/C Programmer's Manual* and *J/Foundation Developer's Guide*. For information about how to use DB–Access, see the *IBM Informix DB-Access User's Guide*.

Stored Procedure Language (SPL)

Informix Stored Procedure Language (SPL) is an extension to SQL that provides flow control such as looping and branching. Consider using SPL procedures and routines for SQL-intensive tasks. An *SPL procedure* is a routine written in SPL and SQL that does not return a value. An *SPL function* is a routine written in SPL and SQL that returns a single value, a value with a complex data type, or multiple values.

You can write user-defined routines in the SPL, C, and Java languages and store them in the database.

For information on how to create and use SPL routines, see the *IBM Informix Guide to SQL: Tutorial*. For the syntax diagrams of SPL statements, see the *IBM Informix Guide to SQL: Syntax*. For performance aspects, see your *IBM Informix Performance Guide*.

System Catalog Tables

Sometimes called the "data dictionary," the system catalog tables describe the structure of the database. The database server automatically generates the system catalog tables when you create a database. Each system catalog table contains specific information about elements in the database.

System catalog tables track database objects such as the following:

- Tables, views, sequences, synonyms, and sequence objects
- Columns, constraints, indexes, and fragments
- Triggers
- Procedures, functions, routines, and associated messages
- Authorized users and privileges
- User-defined routines
- Data types and casts (IDS)
- Aggregate functions (IDS)
- Access methods and operator classes (IDS)
- Inheritance relationships (IDS)
- External optimizer directives (IDS)

For information about the system catalog, see the *IBM Informix Guide to SQL: Reference*.

Data Types

Every column in a table is assigned a data type. The data type precisely defines the values that you can store in that column. Informix supports the data types that Figure 3-1 shows.

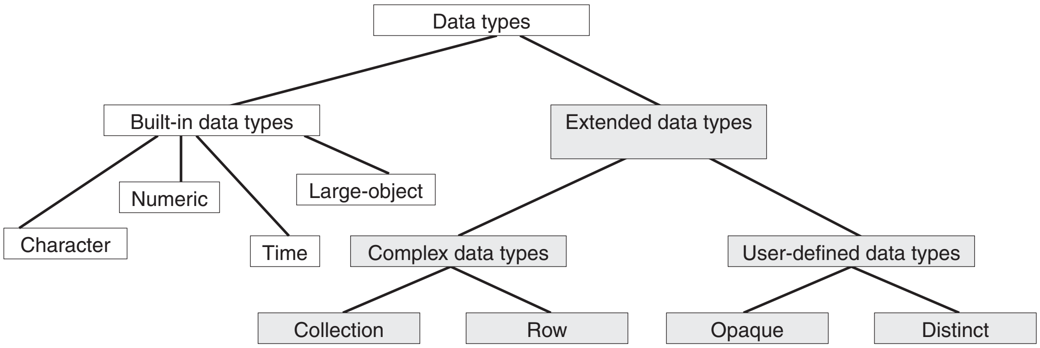

Figure 3-1. Overview of Supported Data Types

For a description of the data types and data type conversions, see the *IBM Informix Guide to SQL: Reference*. For information about how to choose data types for your relational or object-relational database, see the *IBM Informix Database Design and Implementation Guide*. For information on how to extend existing data types, create new casts, and define new data types for a database, see *IBM Informix User-Defined Routines and Data Types Developer's Guide.*

Table 3-2 describes the data types that you can define for a database.

Table 3-2. Data Types

Data Type	Explanation	Examples
Built-in data types	Fundamental data types that cannot be broken into smaller pieces Serve as building blocks for other data types.	• BIGINT • BIGSERIAL • BLOB • BOOLEAN • BYTE • CHAR(*n*) • CHARACTERVARYING(*m*,*r*) • CLOB • DATE • DATETIME • DECIMAL or NUMERIC(*p*,*s*) • DOUBLE PRECISION • FLOAT • IDSSECURITYLABEL • INTEGER • INTERVAL • LVARCHAR(*m*) • MONEY(*p*,*s*) • NCHAR(*n*) • NVARCHAR(*m*,*r*) • REAL or SMALLFLOAT • SERIAL • SERIAL8 • SMALLINT • TEXT • VARCHAR(*n*,*r*)
Complex data types	Combination of other data types An SQL statement can access individual components within the complex type.	
Collection types	Complex data types Include groups of elements of the same data type, which can be any built-in or complex data type.	• SET • LIST • MULTISET
Row types	Complex data types Include groups of related data *fields* of any data type that form a template for a record.	• Named row type • Unnamed row type
User-defined data types	Include distinct and opaque types	

Table 3-2. Data Types (continued)

Data Type	Explanation	Examples
Distinct data types	Have the same internal structure as existing data types They have distinct names and functions that make them different from the source type.	CREATE DISTINCT TYPE birthday AS DATE The data type is **birthday**.
Opaque data types	User-defined types The internal structure is not known to the database server.	CREATE OPAQUE TYPE fixlen_typ (INTERNALLENGTH=8, CANNOTHASH) The data type is **fixlen_typ**.
DataBlade data types	New data types from IBM Informix DataBlade module A DataBlade module is a collection of functions that describe special-purpose data types and all of their support functions. A DataBlade module can contain any or all of the previously described data types.	Examples of DataBlade modules include: IBM Informix • Image Foundation DataBlade module • 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 • Informix Video Foundation DataBlade Module • IBM Informix Web DataBlade Module

Application Types

Two main classes of applications operate on data in an Informix database:

- Online transaction processing (OLTP) applications
- Decision-support system (DSS) applications

OLTP Applications

OLTP applications are often used to capture new data or update existing data. An order-entry system is a typical example of an OLTP application.

OLTP applications have the following characteristics:

- Transactions that involve small amounts of data
- Indexed access to data
- Many users
- Frequent queries and updates
- Fast response times

DSS Applications

DSS applications often report on or consolidate data that OLTP operations have captured over time. These applications provide information that is often used for accounting, strategic planning, and decision making. Data in the database is typically queried but not updated during DSS operations. Typical DSS applications include payroll, inventory, and financial reports.

For more information on how to manage DSS systems, see your *IBM Informix Performance Guide*.

Database Support

Informix supports the following types of databases:

- Relational
- ANSI compliant
- Object-relational
- Dimensional (data warehouse)
- Distributed

Relational Databases

Relational database management systems (RDBMS) are designed for online transaction processing (OLTP), although you can use an RDBMS for DSS processing. An RDBMS focuses on high-speed, short-running queries and transactions on the following types of simple data:

- Integer
- Floating-point number
- Character string, fixed, or variable length
- Date and time, time interval
- Numeric and decimal
- Simple large objects (TEXT and BYTE data)

For information about relational databases, see the *IBM Informix Database Design and Implementation Guide* and the *IBM Informix Guide to SQL: Syntax*.

ANSI-Compliant Databases

You create an ANSI-compliant database when you use the MODE ANSI keywords in the CREATE DATABASE statement. You can use the same SQL statements with both ANSI-compliant databases and non-ANSI-compliant databases. You might want to create an ANSI-compliant database for the following reasons:

- Privileges and access to objects

  ANSI rules govern privileges and access to objects such as tables and synonyms.
- Name isolation

  The ANSI table-naming scheme allows different users to create tables in a database without having to worry about name conflicts.
- Transaction isolation
- Data recovery

ANSI-compliant databases enforce unbuffered logging and automatic transactions for Informix.

For information about ANSI-compliant databases, see the *IBM Informix Database Design and Implementation Guide* and the *IBM Informix Guide to SQL: Syntax*.

Object-Relational Databases

Object-relational database management systems (ORDBMS) combine relational and object-oriented capabilities. Choose an object-relational database if you need greater flexibility in the types of data that the database server can store and manipulate. An example of an object-relational database is an online store catalog.

You can extend the capability of the database server by defining new data types and user-defined routines (UDRs) that let you store, access, and manage images, audio, video, large text documents, and so on.

An object-relational database supports the following data types and extensibility:

- Alphanumeric data (such as character strings, integers, decimal, floating point, and date)
- Simple large objects (TEXT and BYTE data types)
- Smart large objects (BLOB and CLOB data types)
- User-defined types (opaque and distinct types)
- Complex data types (composites of existing data types)
- User-defined routines
- Operator functions
- User-defined casts
- User-defined aggregates
- Type and table inheritance
- DataBlade modules
- User-defined virtual processors
- User-defined access methods (see "Access Methods" on page 3-18)

For information about object-relational databases, see the *IBM Informix Database Design and Implementation Guide* and *IBM Informix Guide to SQL: Syntax*. For more information about extending the database server, see *IBM Informix User-Defined Routines and Data Types Developer's Guide* and *J/Foundation Developer's Guide*.

Simple and Smart Large Objects

The database server supports simple large objects and smart large objects for storing large chunks of binary or text data in a database. A *large object* is a data object that is logically stored in a table column but physically stored independently of the column. Large objects are stored separately from the table because they typically store very large amounts of data.

For more information on simple and smart large objects, see the *IBM Informix Guide to SQL: Reference* and *IBM Informix Guide to SQL: Tutorial*.

Simple Large Objects (TEXT and BYTE Data Types): The database server stores simple large objects in a dbspace or blobspace. Simple large objects do not support random access to the data. When you transfer a simple large object between a client application and the database server, you must transfer the entire BYTE or TEXT value.

Smart Large Objects (CLOB and BLOB Data Types): You can use smart large objects to store user-defined types such as video and audio clips, pictures, large text documents, and spatial objects such as drawings and maps.

The database server stores smart large objects in *sbspaces*. You can control the logging characteristics of smart large objects and sbspaces independently from the logging characteristics of the database. Use a *temporary sbspace* to store *temporary smart large objects* without any logging.

Programmers can use functions similar to UNIX and Windows functions to read, write, and seek smart large objects. Informix provides the smart-large-object API in the DataBlade API and the Informix ESQL/C programming interface.

For information on sbspaces, see the *IBM Informix Administrator's Guide*. For information on how to create an sbspace, see the discussion of **onspaces** in the *IBM Informix Administrator's Reference*. For information on how to calculate space and tune sbspaces, see your *IBM Informix Performance Guide*. For information on how to access a simple large object or a smart large object from a client application, see the *IBM Informix ESQL/C Programmer's Manual*. For information on using the DataBlade API with smart large objects, see the *IBM Informix DataBlade API Programmer's Guide*.

User-Defined Data Types

You can create *user-defined data types* (UDTs) to extend the database server and provide greater flexibility in the types of data that you can store and manipulate. User-defined data types can be opaque or distinct.

You create a *distinct data type* with the CREATE DISTINCT TYPE statement. A distinct type has the same internal structure as an existing data type. However, it has a distinct name and therefore distinct functions that make it different from its source type. Once you create the distinct type, you can use it anywhere that other data types are valid.

You create and register an *opaque data type* with the CREATE OPAQUE TYPE statement. An opaque type stores a single value and cannot be divided into components by the database server. It is implemented as a structure and a set of routines that allow the database server to support the data type.

Complex Data Types

A *complex data type* is a composite of existing data types. It can be a named row type, unnamed row type, or collection type. For example, you might create a complex type whose components include built-in types, opaque types, distinct types, or other complex types.

A *collection type* is a group of elements of the same data type. Collection data types let you store and manipulate collections of data within a single row of a table.

A *row type* is a sequence of one or more fields. Each field has a name and a data type. The fields of a row are comparable to the columns of a table, but there are important differences. You cannot define a default value for a field, you cannot define constraints on a field, and you cannot use fields with tables, only with row types. Row types can be named or unnamed:

- A *named row type* is a group of fields that are defined under a single name. A field refers to a component of a row type. Once you create a named row type, the name that you assign to the row type represents a unique data type within the database.

- An *unnamed row type* is a group of fields that are defined by their structure. Unlike a named row type, which you can use to define a table, you cannot use an unnamed row type to define a table. Use an unnamed row type to define a column, field, or variable.

User-Defined Routines

A *routine* is a collection of program statements that perform a particular task. A *user-defined routine* (UDR) is a routine that you can define and that can be invoked within an SQL statement or within another UDR. A UDR can either return values or not, as follows:

- A user-defined function returns one or more values and therefore can be used in SQL expressions.
- 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 Informix DataBlade Developers 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 DataBlade Developers Kit User's Guide*.

Dimensional Databases

Informix 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 Queries and Multiphase Transactions

Informix supports distributed queries for transactions that involve only built-in data types, and certain built-in opaque and DISTINCT types, across the following server instances:

- Multiple databases of a single server instance (cross-server distributed queries)
- Multiple database server instances (cross-database distributed queries)

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. Informix 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. Informix 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.

Informix 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. Informix 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

Informix 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

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.
- **Publication**. 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 (DBSA)	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:

1. Set up and configure system hardware and software.
2. Install the database server and client applications.
3. Migrate data from an earlier version of the database server (if needed).
4. Configure the environment.
   a. Set required environment variables.
   b. Prepare connectivity files.
   c. Prepare the configuration file.
   d. Allocate and initialize disk space.

5. Choose a database type.
6. Create the demonstration database (optional).

Planning, Installing, and Configuring Tasks

Table 4-2 describes the planning, installing, and configuring tasks.

Table 4-2. Planning, Installing, and Configuring Tasks

If You Want To	Publication
Learn about the new features in Informix.	Chapter 2, "Using New Features in Informix," on page 2-1
Acquaint yourself with terms used in IBM Informix publications.	*IBM Informix Glossary*
Plan a database server installation.	*IBM Informix Administrator's Guide*
Plan to migrate to Dynamic Server from an earlier database server version.	*IBM Informix Migration Guide*
Plan and configure: • Operating system • Hardware and system software upgrades • Network capacity • Integration with other vendor products and applications • Disk and storage media	• *IBM Informix Installation Guide for Windows*, *IBM Informix Installation Guide for UNIX, Linux, and Mac OS X*, or the *Quick Beginnings for IBM Informix Dynamic Server Express Edition;* • *IBM Informix Administrator's Guide*
Determine optimal memory configuration. Determine optimal disk layout and striping.	• *IBM Informix Performance Guide* • System documentation • Machine notes
Install Informix on UNIX or Linux: • Typical installation • Silent installation • Custom installation • Modify an installation • Uninstall • Multiple residency	• *IBM Informix Installation Guide for UNIX, Linux, and Mac OS X* • *Quick Beginnings for IBM Informix Dynamic Server Express Edition*
Install Informix on Windows: • Typical installation • Silent installation • Custom installation • Modify an installation • Uninstall • Multiple residency • Cluster installation	• *IBM Informix Installation Guide for Windows* • *Quick Beginnings for IBM Informix Dynamic Server Express Edition*
Initialize the database server. Manage database server operating modes.	*IBM Informix Administrator's Guide*

Table 4-2. Planning, Installing, and Configuring Tasks (continued)

If You Want To	Publication
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*
Install and configure Informix 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 the OpenAdmin Tool for IDS (OAT) to configure the database server.	OAT online help OAT *How do I?* document
Use 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 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 Informix 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 Informix to an earlier database server version.	*IBM Informix Migration Guide*
Interpret error messages.	*IBM Informix Error Messages* or **finderr** utility

Administering the Database Server

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

1. Prepare the operating system to automatically start and stop the database server when the system is shut down or rebooted.
2. 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.
3. Check that users have set the correct environment variables.
4. Review the database server configuration parameters.
5. Transfer data that was created on other IBM Informix database servers.

Administration Tasks

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

Table 4-3. Administration Tasks

If You Want To	Publication
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 OpenAdmin Tool for IDS to administer and monitor the database server.	OpenAdmin Tool online help

Table 4-3. Administration Tasks (continued)

If You Want To	Publication
Use the following utilities to perform administrative tasks: • **genoncfg** • **oncheck** • **onclean** • **oncmsm** • **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*
• 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. • Perform a table-level restore. • Set up HDR with external backup and restore.	*IBM Informix Backup and Restore Guide*
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*

Table 4-3. Administration Tasks (continued)

If You Want To	Publication
• Encrypt network data. • Encrypt column-level data. • Prevent unauthorized connections to the database server. • Control access to database objects. • Control access to data.	*IBM Informix Security 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 Security 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 IBM Informix database server and pass that information to a network manager.	*IBM Informix SNMP Subagent Guide*
Design, define, monitor, and control your Enterprise Replication system.	*IBM Informix 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:

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

Performance Tuning Tasks

Table 4-4 on page 4-7 lists performance-related tasks and where to find information on those tasks.

Table 4-4. Performance Tuning Tasks

If You Want To	Publication
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*
Use OpenAdmin Tool for IDS performance tools.	OpenAdmin Tool online help
Adjust the database server configuration. • 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 query drill down. • 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.	*IBM Informix Performance Guide*

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 Informix.

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	Publication
• 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 Enterprise Replication Guide*

Chapter 5. Designing, Maintaining, and Extending the Database

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	Publication
Work with the tables in the **sysmaster** database.	*IBM Informix Administrator's Reference*
• Place tables on disk. • 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.	*IBM Informix Performance Guide*
Find corrective actions to error messages.	*IBM Informix Error Messages*
Design a database (choose whether to implement the relational, object-relational, or dimensional database model). 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.	*IBM Informix Database Design and Implementation Guide*

Table 5-1. Database Tasks (continued)

If You Want To	Publication
• Invoke the **DB–Access** utility. • 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.	*IBM Informix DB-Access User's Guide*
• Learn how GLS affects database server migration. • Load and unload data. • Display the database schema with **dbschema**.	*IBM Informix Migration Guide*
• Use the system catalog tables to track objects. • Set environment variables. • Find a description of the tables in the **stores_demo** or **superstores_demo** database. • Look up definitions in the glossary.	*IBM Informix Guide to SQL: Reference*
• 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	Publication
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*
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 IBM Informix SQL APIs and database servers handle different languages, cultural conventions, and code sets.	*IBM Informix GLS User's Guide*

Table 5-2. Application Development Tasks (continued)

If You Want To	Publication
• 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 Informix 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 Informix ESQL/C functions with the DataBlade API.	*IBM Informix DataBlade API Function Reference*

Table 5-2. Application Development Tasks (continued)

If You Want To	Publication
• Use Java to create client applications or applets that run against IBM Informix. • 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 IBM 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 IBM 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*
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 Publications" on page 6-4

Table 5-2. Application Development Tasks (continued)

If You Want To	Publication
Develop existing .NET applications that access IBM Informix databases using the IBM Informix .NET Provider provided with Client SDK.	*IBM Informix .NET Provider Reference Guide*
Develop new. NET applications that access IBM Informix or DB2 databases using the IBM Data Server Provider for .NET.	*IBM Data Server Provider for .NET Programmer's Guide, Informix Edition*
Develop Java or SQLJ applications that access IBM Informix or DB2 databases using the IBM Data Server Driver for JDBC and SQLJ.	*IBM Data Server Driver for JDBC and SQLJ for Informix*
Develop PHP applications with the PDO_INFORMIX extension to connect to IBM Informix, or develop PHP applications with the PDO_IBM extension to connect to IBM Informix or DB2 databases.	http://publib.boulder.ibm.com/infocenter/idshelp/v115/topic/com.ibm.conn.doc/php.htm
Develop applications in the Ruby programming language. Two Ruby Gems are available: the Ruby Informix Gem to access IBM Informix databases and the Ruby Gem for IBM Data Servers to access IBM Informix or DB2 databases.	http://publib.boulder.ibm.com/infocenter/idshelp/v115/topic/com.ibm.conn.doc/ruby.htm
Learn about and install the Call Level Interface (CLI) component of the IBM Data Server Driver for ODBC and CLI. This driver is a prerequisite for the PDO for Data Server clients for PHP connections and for the Ruby Gem for Data Server clients.	http://publib.boulder.ibm.com/infocenter/idshelp/v115/topic/com.ibm.conn.doc/installcli.htm

Chapter 6. Using the Documentation

This chapter contains an alphabetical list of IBM Informix publications.

The IBM Informix Documentation Set

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

IBM Informix Publications

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

Table 6-1. Database Server Publications

Book Title	**Description**
IBM Informix Backup and Restore Guide	This publication 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 publication describes the API functions.
IBM Informix DataBlade API Programmer's Guide	This publication describes the API, the C-language application programming interface that is provided with Informix. Use the API to develop client and server applications that access data stored in a Informix 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 Administrator's Guide	This user guide for system and database server administrators discusses the concepts and procedures for managing Informix. It is intended to help you understand, configure, and use the database server. The short publication name is *IBM Informix Administrator's Guide*.
IBM Informix Administrator's Reference	This reference publication 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 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.
IBM Informix 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 Installation Guide for UNIX, Linux, and Mac OS X	This guide contains instructions for installing Informix on UNIX, Linux, and Mac OS X. It also describes how to solve common installation problems.

Table 6-1. Database Server Publications (continued)

Book Title	**Description**
IBM Informix Installation Guide for Windows	This guide contains instructions for installing Informix on Windows.
IBM Informix Performance Guide	This guide explains how to configure and operate Informix to achieve optimum performance and optimize SQL queries. The short publication 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 publication 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 publication 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 publication 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 publication 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 publication 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.
IBM Informix Security Guide	This guide describes the security features to secure data and the secure auditing facility.
IBM Informix User-Defined Routines and Data Types Developer's Guide	This guide explains how to define new data types and create UDRs on Informix. 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.

Table 6-1. Database Server Publications (continued)

Book Title	Description
IBM Informix Virtual-Index Interface Programmer's Guide	This publication 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 publication describes the syntax, API function calls, and data structures.
IBM Informix Virtual-Table Interface Programmer's Guide	This publication 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 publication 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 the database server.

Client SDK and Connectivity Publications

Table 6-2 lists the IBM Informix Client Software Development Kit and connectivity publications that you can use when you work with Informix.

Table 6-2. Client Publications for Informix

Book Title	Description
IBM Informix Client Products Installation Guide	This guide contains instructions for installing the IBM Informix Client Software Development Kit and IBM Informix Connect on Linux, UNIX, Mac OS X, and Windows.
IBM Informix Embedded SQLJ User's Guide	This publication 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 publication explains how to use Informix ESQL/C, the IBM 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 IBM 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 publication describes how to install, administer, and tune performance for Informix MaxConnect. Informix 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.
IBM Informix ODBC Driver Programmer's Manual	This publication explains how to use the IBM Informix ODBC Driver to access IBM Informix databases and the database server. The IBM Informix ODBC Driver is the IBM Informix implementation of the Microsoft Open Database Connectivity (ODBC) interface.

Table 6-2. Client Publications for Informix (continued)

Book Title	Description
IBM Informix OLE DB Provider Programmer's Guide	This publication 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 Data Server Provider for .NET Programmer's Guide, Informix Edition	This guide describes how to use the IBM Data Server Provider for .NET to access and manipulate data in IBM Informix databases. The provider, which is based on the Distributed Relational Database Architecture (DRDA) protocol, is also integrated with DB2. This capability means that the common features of the IBM Data Server Provider for .NET allow you to write client applications that can use both DB2 and Informix data servers.
IBM Data Server Driver for JDBC and SQLJ for Informix	This guide describes how to install, load, and use IBM Data Server Driver for JDBC and SQLJ to connect to an IBM Informix database from a Java application or applet. The driver, which is based on the Distributed Relational Database Architecture (DRDA) protocol, is also integrated with DB2. This capability means that the common features of the IBM Data Server Driver for JDBC and SQLJ allow you to write client applications that can use both DB2 and Informix data servers.

DataBlade Publications

Table 6-3 lists the publications that you can use when you develop or use DataBlade modules and web-based applications with IBM Informix.

Table 6-3. DataBlade and Tools Publications

Book Title	Description
IBM Informix Database Extensions User's Guide	This guide explains how to use the following IBM Informix DataBlade modules that come with Informix: • Large Object Locator, a foundation DataBlade module for large objects management that can be used by other modules that create or store large-object data. • MQ DataBlade module, to allow IBM Informix database applications to communicate with other MQSeries® applications with MQ messaging. • Binary DataBlade module, which includes binary data types to store binary-encoded strings that can be indexed for quick retrieval. • Basic Text Search DataBlade module, to search words and phrases in an unstructured document repository stored in a column of a table. • Node DataBlade module for the hierarchical data type, to represent hierarchical data within the relational database. • Web Feature Service DataBlade module, to add an Open Geospatial Consortium (OGC) web feature service as a presentation layer for the Spatial and Geodetic DataBlade modules.
IBM DataBlade Developers Kit User's Guide	This guide describes how to develop and package DataBlade modules using BladeSmith and BladePack.
IBM Informix DataBlade Module Development Overview	This publication provides an overview of DataBlade module development.

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

Book Title	Description
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 IBM Informix databases. BladeManager runs on client computers.
Excalibur Text Search DataBlade Module User's Guide	This publication explains how to perform text searches and retrieval using the Informix 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 publication 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 publication explains how to use the IBM Informix Geodetic DataBlade Module to store and use spatio-temporal data such as maps.
IBM Informix Image Foundation DataBlade Module User's Guide	This publication 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 Data User's Guide	This publication 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 publication 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 publication describes how to use the Informix Video Foundation DataBlade Module to store video technology in a media management system.
IBM Informix Web DataBlade Module Administrator's Guide	This publication describes how to administer web applications that use the IBM Informix Web DataBlade Module to dynamically retrieve data from IBM Informix databases.
IBM Informix Web DataBlade Module Application Developer's Guide	This publication explains how to use the IBM Informix Web DataBlade Module to develop web applications that dynamically retrieve data from IBM Informix databases.

Appendix A. Database server utilities

IBM Informix 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 publication of the IBM Informix documentation set, as shown in the final column of the table.

Table A-1. Informix Utilities

Utility	Description
archecker utility	Verifies backups and performs table-level restores.
chkenv utility	Checks the validity of shared or private environment-configuration files. **chkenv** validates the names of the environment variables in the file, but not their values.
cdr utility	Controls Enterprise Replication operations.
ClassGenerator utility	Generates a Java class for a named row type defined in the system catalog.
clusterIT_a	Configures Informix on the primary node for use in a cluster environment.
clusterIT_b	Configures Informix on the secondary node for use in a cluster environment.
dbexport utility	Unloads a database into text files for later import into another database and create a schema file.
dbimport utility	Creates and populates a database from text files. Use the schema file with dbimport to recreate the database schema.
dbload utility	Loads data into databases or tables.
dbschema utility	Creates a file that contains the SQL statements needed to replicate a specified table, view, or database, or view the information schema.
GenMacKey	Improves the reliability of message verification using encryption.
genoncfg utility	Expedites the process of customizing the default Informix configuration file (**onconfig.std**) to the host environment and your planned usage of a database server instance.
geovalidate	Validates GeoObject data.
glfiles utility	Generates a list of GLS-related files on the **UNIX** operating system.
HPL utility	Migrates optical TEXT and BYTE data by either loading or unloading a complete or partial table.
ifxdeploy utility	Deploy a snapshot or removes a snapshot that you have already deployed.
ifx_getversion utility	On the **UNIX** operating system: outputs the complete version name of an IBM Informix library.
ILOGIN utility	On the Windows operating system: tests the connection to IBM Informix.
imcadmin	Starts or stops Informix MaxConnect, or gather statistics on Informix MaxConnect.
infoshp utility	Reports information extracted from headers of the `.shp`, `.shx`, and `.dbf` files that make up ESRI shapefiles.
ipload utility	Manages the onpload database and create and the components of load and unload jobs using a UNIX-based GUI.
ism utility	Manages IBM Informix Storage Manager, storage devices, and media volumes.
ism_add utility	Adds administrative users and storage devices to the ISM server.
ism_catalog utility	Recreates entries in the ISM catalog for a volume that has been removed from the ISM catalog, finds the ISM server bootstrap, creates a new ISM bootstrap, and recovers the entire ISM catalog.

Table A-1. Informix Utilities (continued)

Utility	Description
ism_clone utility	Initiates a clone operation on a storage volume or specific save set from the command line.
ixpasswd.exe utility	On the Windows operating system: Changes the logon password for all services which log on as user **informix**.
ixsu.exe utility	On the Windows operating system: Launches a command-line window that runs as the specified user.
loadshp utility	Loads spatial features and associated attributes from an ESRI shapefile into a table in an IBM Informix database.
ntchname.exe utility	On the Windows operating system: Changes the registry entries for IBM Informix from the old host name to the new host name.
onaudit utility	Manages audit masks and auditing configurations.
onbar utility	Backs up and restores storage spaces and logical logs.
oncheck utility	Checks specified disk structures for inconsistencies, repair inconsistent index structures, and display information about disk structures.
onclean utility	Forces a shut down of the database server when normal shut down with the onmode utility fails or when you cannot restart the server.
oncmsm utility	Starts the Connection Manager, which manages and redirects client connection requests based on service level agreements configured by the system administrator.
ondblog utility	Changes the logging mode.
oninit utility	Brings the database server online.
onload utility	Loads data that was created with **onunload** into the database server.
onlog utility	Displays the contents of logical-log files.
onmode utility	Changes the database server operating mode and perform various other operations on shared memory, sessions, transactions, parameters, and segments.
ON-monitor utility	Performs administrative tasks using the ON–monitor menus.
onparams utility	Modifies the configuration of logical logs or physical logs.
onpassword utility	Encrypts and decrypts password files for Enterprise Replication and Connection Manager.
onperf utility	Monitors database server performance (create graphs, query trees, show status and metrics).
onpladm utility	Writes scripts and create files that automate data load and unload jobs.
onpload utility	Manages load and unload jobs directly from the command line.
onrestorept utility	Restore a server instance back to its original state just before the start of an upgrade.
onsecurity utility	Checks the security of a file, directory, or path and troubleshoots any existing problems.
onshowaudit utility	Extracts information from an audit trail.
onsymsync utility	Synchronizes the sysutils database and emergency boot file with the storage-manager catalog.
onsnmp utility	Provides information that lets an SNMP Network Manager monitor all the IBM Informix databases that are on a network.
onspaces utility	Modifies dbspaces, blobspaces, sbspaces, or extspaces.
onstat utility	Monitors the operation of the database server.
ontape utility	Logs, backs up, and restores data.
onunload utility	Unloads data from the database server.

Table A-1. Informix Utilities (continued)

Utility	Description
setnet32 utility	Sets or modifies environment variables and network parameters that IBM Informix products use at run time.
snmpdm utility	On the **UNIX** operating system: ensures that both the SNMP master agent and the onsrvapd server-discovery daemon are running on a host.
SqlhDelete utility	Deletes the `sqlhosts`entries from the LDAP server.
SqlhUpload utility	Loads the `sqlhosts` entries from a flat ASCII file to the LDAP server in the prescribed format.
unloadshp utility	Copies spatial features and associated attributes from a table in an IBM Informix database into an ESRI shapefile.
w4gl utility	Publishes, deploys, packages, and subscribes to Web services.

Appendix B. Accessibility

IBM strives to provide products with usable access for everyone, regardless of age or ability.

Accessibility features for IBM Informix products

Accessibility features help a user who has a physical disability, such as restricted mobility or limited vision, to use information technology products successfully.

Accessibility features

The following list includes the major accessibility features in IBM Informix products. These features support:

- Keyboard-only operation.
- Interfaces that are commonly used by screen readers.
- The attachment of alternative input and output devices.

Tip: The information center and its related publications are accessibility-enabled for the IBM Home Page Reader. You can operate all features by using the keyboard instead of the mouse.

Keyboard navigation

This product uses standard Microsoft Windows navigation keys.

Related accessibility information

IBM is committed to making our documentation accessible to persons with disabilities. Our publications are available in HTML format so that they can be accessed with assistive technology such as screen reader software.

You can view the publications in Adobe Portable Document Format (PDF) by using the Adobe Acrobat Reader.

IBM and accessibility

See the *IBM Accessibility Center* at http://www.ibm.com/able for more information about the IBM commitment to accessibility.

Dotted decimal syntax diagrams

The syntax diagrams in our publications are available in dotted decimal format, which is an accessible format that is available only if you are using a screen reader.

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, that element 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 refers 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 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 can write HOST STATE, but you cannot 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 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

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other 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 grant 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:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
1623-14, Shimotsuruma, Yamato-shi
Kanagawa 242-8502 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 websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites 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 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 document 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 is for planning purposes only. The information herein is subject to change before the products described become available.

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. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

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

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at "Copyright and trademark information" at http://www.ibm.com/legal/copytrade.shtml.

Adobe, the Adobe logo, and PostScript are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

Intel, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, and Windows NT are trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other company, product, or service names may be trademarks or service marks of others.

Index

Printed in USA

GI13-1803-02

Spine information:

Informix Product Family Informix **Version 11.50** **IBM Informix Getting Started Guide**