




# Release Notes

---

InterBase® 7.5  
November 2004



VERSION 7.5

Borland®  
**InterBase®**

Borland Software Corporation  
100 Enterprise Way, Scotts Valley, CA 95066-3249  
[www.borland.com](http://www.borland.com)

Borland Software Corporation may have patents and/or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents.

COPYRIGHT © 2004 Borland Software Corporation. All rights reserved. All Borland brand and product names are trademarks or registered trademarks of Borland Software Corporation in the United States and other countries. Other product names are trademarks or registered trademarks of their respective holders.

# Contents

## Chapter 1

### General Information

What's in these Release Notes . . . . .	1-1
Chapter 1: General Information . . . . .	1-1
Chapter 2: Recent InterBase features . . . . .	1-1
Chapter 3: Bugfix list . . . . .	1-1
Documentation . . . . .	1-2
Installing the InterBase document set. . . . .	1-2
Accessing the InterBase docs . . . . .	1-2
All platforms. . . . .	1-2
Windows . . . . .	1-2
Linux and Solaris . . . . .	1-2
What's in the doc set . . . . .	1-3
Acrobat Reader with Full Text Search . . . . .	1-4
Contacting Borland. . . . .	1-5
Migration issues . . . . .	1-5
External tables. . . . .	1-5
Migrating from older databases. . . . .	1-5

## Chapter 2

### Recent InterBase features

New in InterBase 7.5 . . . . .	2-1
Multi-Instance. . . . .	2-1
Automatic rerouting of databases. . . . .	2-1
Manual routing of databases. . . . .	2-2
Server side database alias . . . . .	2-2
Embedded database user authentication. . . . .	2-2
New ODS . . . . .	2-2
Global temporary tables . . . . .	2-2
CASE, COALESCE, and NULLIF . . . . .	2-2
Index optimization for NULL/non-NULL values . . . . .	2-2
Memory management allocation algorithms . . . . .	2-3
Stored procedure and trigger cache management. . . . .	2-3
Sort buffer cache management . . . . .	2-3
Greater SMP scalability. . . . .	2-3
Database page buffer cache . . . . .	2-3
Thread-private Latch Cache . . . . .	2-3
New in InterBase 7.1 . . . . .	2-4
New cross-platform installer. . . . .	2-4
New registration . . . . .	2-4
Precision of exact numerics . . . . .	2-4
New drivers . . . . .	2-4
New ODS . . . . .	2-4
Savepoints . . . . .	2-4
Savepoints in SQL. . . . .	2-4
Savepoints in the InterBase API . . . . .	2-5

Savepoints in triggers and stored procedures . . . . .	2-5
A SAVEPOINT example . . . . .	2-5
New keywords . . . . .	2-6
Performance monitoring now accessible in IBConsole . . . . .	2-6
New character sets . . . . .	2-6
Enhanced SMP support . . . . .	2-7
Hyperthreading support on Intel processors . . . . .	2-7
Change in gbak functionality . . . . .	2-7
Hyperthreading support for Intel processors . . . . .	2-7
New SQL command: DROP GENERATOR. . . . .	2-7
Enhanced garbage collection/index handling . . . . .	2-8
IBConsole displays additional object dependencies. . . . .	2-8
Using the InterBase Install API . . . . .	2-8
Documentation fixes and changes. . . . .	2-8
UDF library documentation has been moved . . . . .	2-8
Declaring Blob UDFs . . . . .	2-8
Calling convention for UDFs. . . . .	2-9
Portable UDFs . . . . .	2-9
Correction for YEARDAY range . . . . .	2-9
New in InterClient 4.0 . . . . .	2-9
DataSource properties for InterBase . . . . .	2-9
Standard properties . . . . .	2-9
Extended properties. . . . .	2-10
InterClient connection pooling . . . . .	2-11
InterClient scrollability. . . . .	2-12
The Connection class . . . . .	2-12
The ResultSet class . . . . .	2-12
New InterClient methods . . . . .	2-13
Methods for the Statement and PreparedStatement . . . . .	2-13
classes . . . . .	2-13
The BatchUpdateException class . . . . .	2-14
The DatabaseMetaData.supportsBatchUpdates . . . . .	2-15
function . . . . .	2-15
Additional functions . . . . .	2-15
Code examples. . . . .	2-15
InterClient and the Borland Enterprise Server. . . . .	2-16
Other InterClient enhancements . . . . .	2-16
New in InterBase 7.0 . . . . .	2-16

## Chapter 3

### Bugfix list

Bugs fixed in InterBase 7.5. . . . .	3-1
Bugs fixed in InterBase 7.1. . . . .	3-2
Bugs fixed in InterBase 7.0. . . . .	3-4



# General Information

**Installation, registration, and licensing** These Release Notes address issues relevant to the installed Borland® InterBase® product. For detailed information on installing, registering, and licensing InterBase and for pointers to getting started, please read *IBsetup.html*. You can find this file at the root of the your InterBase CD-ROM, in your InterBase trial download file, or at the root of the InterBase install directory.

**Links** Throughout this document, bold blue type indicates links that are clickable if you are reading this in Acrobat Reader.

## What's in these Release Notes

---

These Release Notes contain the following topics:

### Chapter 1: General Information

**Documentation** Information about installing the InterBase doc set, accessing it, and installing the enhanced Acrobat Reader needed for Full Text Search. In addition, there is a list the topics included in each of the books, to help you find your way around the doc set.

**Contacting Borland** Mailing address, website links, and email.

**Migration issues** A brief discussion of migrating from recent versions of InterBase with a pointer to an extensive discussion of migrating from versions older than 6.

### Chapter 2: Recent InterBase features

**New in InterBase 7.5** Features new to InterBase 7.5.

**New in InterClient 4.7** Features new to InterClient 4.7.

**New in InterBase 7.1** Features new to InterBase 7.1.

**New in InterClient 4.0** Features new to the InterClient 4.0 JDBC driver.

**New in InterBase 7.0** A review of features introduced in InterBase 7.0.

### Chapter 3: Bugfix list

**Bugs fixed in InterBase 7.5.**

**Bugs fixed in InterBase 7.1**

## Bugs fixed in InterBase 7.0

# Documentation

---

This section describes the InterBase document set and how to install Acrobat Reader With Search.

## Installing the InterBase document set

The document set in PDF format requires about 13MB of space on your disk. They install by default when “Client and Server” or “Client” is chosen during the InterBase install process. However, in a custom install, it is possible to choose an install that does not include the document set. If you want to install documents at a later time, run the InterBase install, choose Custom, and select the documentation.

You can also copy them from the Documentation directory of the InterBase CD-ROM or download file.

The books are available for purchase in printed form from <http://shop.borland.com>. For point releases, the PDFs that distribute with InterBase may be more current than the printed books.

## Accessing the InterBase docs

### *All platforms*

#### **2AllBooks.pdf**

The *2AllBooks.pdf* file is a single-page document that contains links to all the books. It is located in the directory with the other PDF files.

#### **CD-ROM and download files**

The document set is available in unarchived form in the *Doc* directory of your InterBase CD-ROM or download files.

#### **HTML reference files**

The API Function Reference and the SQL Statement and Function Reference are both available in HTML form in the *<interbase\_home>/HtmlRef* directory. In the InterBase CD-ROM or download files, they are in the *Doc* directory.

### *Windows*

There are links to each of the books in Start menu | Programs | InterBase | Documentation. You can, of course, copy these links to a more accessible location, or create your own links.

### *Linux and Solaris*

To access the PDF document set, go to *<interbase\_home>/Doc*. You can then display any one of the six books or the Release Notes. *2AllBooks.pdf* is a single-page file that has links to all the other documentation files. To access the HTML reference files, see [HTML reference files](#) above.

## What's in the doc set

The InterBase document set consists of the following books and files:

Document Title	Contents
<i>Release Notes</i>	This document: contact information, new feature descriptions, list of bugs fixed
<i>Operations Guide</i>	A general guide to working with InterBase databases, including: <ul style="list-style-type: none"> <li>• Using IBConsole</li> <li>• Configuring and operating the InterBase server</li> <li>• Network configuration</li> <li>• Database security, configuration, and maintenance</li> <li>• Licensing</li> <li>• Database backup and restore</li> <li>• Database statistics and connection monitoring</li> <li>• Interactive queries, including a language reference</li> <li>• Database and server performance</li> <li>• Migration</li> </ul>
<i>Data Definition Guide</i>	Designing and building InterBase databases, including: <ul style="list-style-type: none"> <li>• General guidelines</li> <li>• Specifying datatypes</li> <li>• Working with domains, tables, indexes, views, stored procedures, triggers, and generators</li> <li>• Planning security</li> <li>• Character sets and collation orders</li> </ul>
<i>Developer's Guide</i>	Developing InterBase database applications using Borland development tools, JDBC, and ODBC, including: <ul style="list-style-type: none"> <li>• Working with UDFs and blob filters, including the complete list of InterBase UDFs and their declarations</li> <li>• Introduction to IBX</li> <li>• Designing and building database applications</li> <li>• Connecting to databases</li> <li>• Understanding datasets</li> <li>• Working with tables, queries, stored procedures, cached updates, and events</li> <li>• Debugging with SQL monitor</li> <li>• Importing and exporting data</li> <li>• Working with InterBase services</li> <li>• Writing install wizards</li> </ul>
<i>Embedded SQL Guide</i>	Developing InterBase database applications using embedded SQL, including: <ul style="list-style-type: none"> <li>• Application requirements</li> <li>• Working with databases, transactions, data definition statements, data, dates and times, blob data, arrays, stored procedures, and events</li> <li>• Error handling</li> <li>• Dynamic SQL</li> <li>• Preprocessing, compiling, and linking</li> </ul>

Document Title	Contents
<i>API Guide</i>	Developing InterBase database applications using the InterBase Application Programming Interface, including: <ul style="list-style-type: none"> <li>• Application requirements</li> <li>• Programming with the InterBase API</li> <li>• Working with databases, transactions, dynamic SQL, blob data, arrays, conversions, events, and services</li> <li>• Handling error conditions</li> <li>• Using the install and licensing APIs</li> <li>• Exporting XML</li> <li>• The InterBase API function reference</li> </ul>
<i>Language Reference</i>	A reference guide to InterBase elements, including: <ul style="list-style-type: none"> <li>• The SQL statement and function reference</li> <li>• Procedures and triggers</li> <li>• Keywords</li> <li>• Error codes and messages</li> <li>• System tables, temporary tables, and views</li> <li>• Character sets and collation orders</li> </ul>
<i>SQL Reference (SqlRef.html)</i>	The complete SQL reference from the “SQL Statement and Function Reference” chapter of the <i>Language Reference</i> .
<i>API Function Reference (APIFunctionRef.html)</i>	The complete API function reference from the <i>API Guide</i> .

## Acrobat Reader with Full Text Search

You need Adobe Acrobat Reader 4 or 5 to view the \*.pdf document files. The InterBase PDF document set has Full Text Search enabled, which allows you to search across the entire document set. To take advantage of this feature, you must have the enhanced version of Acrobat Reader, which has a Search button in addition to the usual Find button. This Search button searches across multiple documents and is available only in the enhanced version of Acrobat Reader. The Find button that is available in the “plain” version of Acrobat Reader searches only a single document at a time.

If you do not already have the enhanced version of Acrobat Reader 5, the English-language version installer is available in the *Documentation/Adobe* directory on the InterBase CD-ROM or download file and in the `<interbase_home>/Doc/Adobe` directory after the InterBase installation.

The enhanced Acrobat Reader is also available for free in many languages from <http://www.adobe.com/products/acrobat/readstep2.html>. In Step 1, choose your desired language and platform. In step 2, be sure to check the “Include the following options...” box.

For information about using Full Text Search, see “Using Full-Text Search” in Chapter One of the *Operations Guide*.



## Contacting Borland

---

### Mailing address

Borland Software Corporation  
 100 Enterprise Way  
 Scotts Valley, CA 95066-3249  
 Phone: 1-831-431-1000

### Websites and email

- For technical support options, see:  
<http://www.borland.com/support>
  - Borland maintains an Internet site on the world-wide web for general InterBase information. The URL of this site is:  
<http://www.borland.com/interbase>
  - Technical information, such as white papers and FAQs, can be found at:  
<http://bdn.borland.com/interbase>
  - For installation and presales questions, visit:  
<http://shop.borland.com>
  - To discuss issues, with other InterBase users, visit:  
<http://info.borland.com/newsgroups>
  - For information about how to contact InterBase representatives outside the U.S. and Canada, look at the following web page:  
<http://www.borland.com/bww/>
  - For issues pertaining to content and presentation on our web site, send email to: [webmaster@borland.com](mailto:webmaster@borland.com)
- 

## Migration issues

---

Migrating databases from InterBase 6, 6.5, or 7.0 to InterBase 7.5 is a simple process:

- 1 Before installing InterBase 7.1, back up each database using your current version of **gbak**.
- 2 Uninstall your current version of InterBase and install InterBase 7.1.
- 3 Restore each database using the InterBase 7.1 **gbak**.

**Note** Upgrading a 7.1 database to 7.5 is automatic as soon as a 7.5 server attaches to a 7.1 database. Converting upgraded databases back to a lower version is not recommended.

### External tables

The only additional step required when migrating from InterBase 6.x occurs when you have external tables. For security reasons, InterBase 7 requires that one of the following two things be true for external tables:

- The tables must be in `<interbase_home>/ext`.

Or

- Each directory that contains an external database file must be listed in `ibconfig` using the `EXTERNAL_FILE_DIRECTORY` entry:

```
EXTERNAL_FILE_DIRECTORY directory1
```

```
EXTERNAL_FILE_DIRECTORY C:\data\mydata\
```

### Migrating from older databases

After attaching to an InterBase 7.0 ODS 11.0 database, or an InterBase 7.1 ODS 11.1 database with InterBase 7.5 the ODS (On Disk Structure) of the database will be automatically upgraded to ODS 11.2.

Migrating databases from InterBase versions 5.x and older to InterBase 6 or later requires planning. Please read the Migration appendix of the InterBase *Operations Guide* for details on migrating clients and databases from InterBase 5.x to InterBase 7.

## Recent InterBase features

This chapter describes what's new in InterBase 7.5 and reviews features introduced in InterBase 7.1 and 7.0.

### New in InterBase 7.5

- Multi-Instance
- Automatic rerouting of databases
- Manual routing of databases
- Server side database alias
- Embedded database user authentication
- New ODS
- Global temporary tables
- CASE, COALESCE, and NULLIF
- Memory management allocation algorithms
- Index optimization for NULL/non-NULL values

- Stored procedure and trigger cache management
- Sort buffer cache management
- Greater SMP scalability
- Database page buffer cache
- Thread-private Latch Cache
- Error reporting improved in interbase.log
- New in InterClient 4.7
- New in InterBase 7.1
- New in InterClient 4.0
- New in InterBase 7.0

## New in InterBase 7.5

---

InterBase 7.5 includes the following new features:

### Multi-Instance

InterBase 7.5 now allows multiple instances of InterBase servers to run simultaneously. In the past multiple versions of the InterBase server could not be run on the same machine. Previously when an application that utilized one version of InterBase, another application that utilized another version of InterBase could not be run. Now with InterBase 7.5 Borland has added the ability run multiple instances of InterBase on the same machine.

With InterBase 7.5 one previous version (major release) of InterBase, i.e. InterBase 7.1, or InterBase 6.x, etc. will be able to be run simultaneously. Multiple instances of InterBase 7.5 can be run simultaneously.

**Note** Separate licensing is required for all instances of InterBase.

## Automatic rerouting of databases

Now that InterBase 7.5 allows multiple instances of InterBase to run on the same machine this feature will allow configurations where some database connections can be rerouted to a different InterBase server instance on the same machine. See the Operations Guide for implementation details.

## Manual routing of databases

This solution allows application developers to explicitly specify a unique INTERBASE environmental variable for a local connection, or a unique TCP/IP protocol name for remote connections. This feature is useful if application developers want to isolate their application from other versions of InterBase installed on the same machine. See the Multi-Instance section in the Operations Guide for implementation details.

## Server side database alias

Database alias renames a database file within the context of the server. This beneficial feature enables clients to connect to databases regardless of the knowledge of its exact location. See the Operations Guide for implementation details.

## Embedded database user authentication

This is a security enhancement new in InterBase 7.5. Now that InterBase 7.5 can manage multiple databases for unrelated applications the embedded database user authentication feature allows custom user account management that is not shared with other InterBase applications. See the Operations Guide for implementation details.

**Note** There is no embedded SQL support for DDL for embedded database user authentication.

## New ODS

InterBase 7.5 uses ODS (On Disk Structure) 11.2. This new ODS is required to accommodate new system tables, changes to existing system tables, and embedded database user authentication. InterBase 7.0 ODS 11.0 databases and InterBase 7.1 ODS 11.1 databases are automatically upgraded to ODS 11.2 when an InterBase 7.5 server attaches to these databases. To migrate databases with an ODS less than 11.0 or 11.1, backup these database versions with the older version of InterBase and restore them using InterBase 7.5 IBConsole or **gbak**.

**Note** Since InterBase 7.5 automatically upgrades ODS 11.0 and 11.1 databases to ODS 11.2 it is recommended that you retain a copy of your database in an earlier ODS if you plan on using databases with InterBase 7.0 or 7.1.

## Global temporary tables

This feature implements the functionality of SQL global temporary tables in InterBase 7.5. Previously InterBase developers simulated temporary tables with permanent base tables. The developer was responsible for the application dropping those tables and performing any housekeeping to empty those tables if the application or database server abnormally terminated. With this new SQL temporary table feature in InterBase 7.5 all of the namespace and life cycle issues are transparently managed once the temporary table is declared to the database schema, thus making application development much easier. See the Language Reference for implementation details.

**Note** There is no embedded SQL support for DDL on temporary tables.

## CASE, COALESCE, and NULLIF

New language features have been added in InterBase 7.5; CASE, COALESCE, and NULLIF. SQL dialect 1-3 applications should be able to use this functionality except if the evaluation of a value expression yields a value of a data type only supported by SQL dialect 3. An InterBase server should raise an SQL dialect exception in this case. For syntax of these new language implementations, please refer to the Language Reference.

**Note** There is no embedded SQL support for CASE, COALESCE and NULLIF.

## **Index optimization for NULL/non-NULL values**

Nulls are sorted high in indices, meaning that they are located at the end of the index. When a query with greater than is matched to that index, previous versions of InterBase would unnecessarily go out and retrieve all the null-valued record versions from the index even though there is no way the nulls will satisfy the query. This new feature in InterBase 7.5 will optimize such queries by not involving the NULL key values where they are not required. This will improve the overall performance of the Interbase server, and give a better response to many SQL queries. Having NULL values in index keys is quite common (more so in composite keys), and hence this will have a wider impact on all InterBase customers who will see better performance from the product.

## **Memory management allocation algorithms**

In InterBase 7.5 the memory management allocation algorithms have been improved. In previous versions of InterBase the memory management allocation algorithms were designed for the Classic single process architecture. Those outdated algorithms constrained memory resources in the Super Server architecture; the best-fit search algorithm for heap memory allocation will be changed to a first-fit algorithm. Additionally, separate memory heaps will be managed for ad-hoc memory allocation and block-based demands from the SQL and relational engine components to prevent memory fragmentation. This will also enable a degree of SMP parallelization by allowing simultaneous memory allocations. The first-fit algorithm will minimize search latency due to soft/hard page faulting resulting from searching the entire server address space for free memory.

## **Stored procedure and trigger cache management**

The cache management for stored procedures and triggers in InterBase 7.5 has been refined to prevent the server from locking large amounts of memory to maintain the residency of triggers and stored procedures. The cache management will operate on two levels: 1) Deallocate individual clones not in use, and 2) deallocate the primary requests which are not used.

## **Sort buffer cache management**

This feature in InterBase 7.5 improves the page faulting that occurs when 1MB sort buffers are immediately released. This phenomenon can be observed in rapidly executing statements or procedures which perform operations requiring a sort buffer, or during the index build phase of database restoration.

## **Greater SMP scalability**

This feature in InterBase 7.5 improves performance for higher numbers of CPU both physical and logical. Through the use of spin locks and thread-private latch caches the InterBase ATOM synchronization architecture reduces the number and frequency of synchronization points.

## **Database page buffer cache**

In InterBase 7.5 the largest page size has been increased from 8,192 to 16,384 bytes. The largest buffer cache has been increased from 65,000 to 131,000 pages.

## **Thread-private Latch Cache**

The Thread-private Latch Cache (TLC) reduces the number and frequency of atom synchronization locks needed to safely address and pin down resident pages in the database page buffer cache. In doing so, performance is enhanced because of a shorter code path and the avoidance of thread context-switching due to contention over atom synchronization variables.

TLC enhances InterBase performance on SMP servers and single CPU servers when the IBCONFIG parameter MAX\_THREADS is not equal to 1. That is, when multiple threads are allowed to execute concurrently inside the InterBase engine. If MAX\_THREADS is equal to 1 then atom synchronization and TLC are both disabled.

## Error reporting improved in interbase.log

With improved error reporting in interbase.log database names are no longer reported in the 8.3 format. Index names instead of numbers are now reported when verifying a database, and index errors are reported. Also, improved errors are reported when errors are found in ib\_license.dat and the InterBase server is started.

## New in InterClient 4.7

---

InterClient 4.7 includes the following new features:

### Savepoints

Savepoints were introduced to InterBase 7.1, InterClient 4.7 now surfaces this as a JDC standard implementation with the class IBSavepoint

The following example shows how to use IBSavepoint:

```
package bo_blob;
import java.sql.*;

public class testSavepoints
{
    public static void main(String[] args)
    {
        try
        {
            System.out.println("Interclient version: " +
                interbase.interclient.Driver.getInterClientVersionInfo());

            String url = "jdbc:interbase://localhost/c:\\smistry\\foo.ib";
            Connection con;
            Class.forName("interbase.interclient.Driver");
            con = DriverManager.getConnection(url, "sysdba", "masterkey");

            con.setAutoCommit(false);
            Statement stmt = con.createStatement();

            try
            {
                stmt.executeUpdate("create table test (a int)");
            }
            catch (SQLException se)
            {
                System.out.println(se.getMessage());
            }
            con.commit();
            Savepoint savepoint = con.setSavepoint("sean");
            Savepoint savepoint = con.setSavepoint();
            PreparedStatement pstmt = con.prepareStatement("insert into test values (?)");
            stmt.executeUpdate("insert into test values (1)");
            ResultSet rs = stmt.executeQuery("Select a from test");
            while (rs.next())
            {
```

```

        System.out.println("a = " + rs.getInt("a"));
    }
    con.rollback(savepoint);
    rs = stmt.executeQuery("Select a from test");
    while (rs.next())
    {
        System.out.println("a = " + rs.getInt("a"));
    }
    con.commit();
    stmt.executeUpdate("drop table test");
}
catch (Exception se)
{
    se.printStackTrace();
}
}
public testSavepoints()
{
}
}

```

## ParameterMetaData

InterClient had always implemented a version of ParameterMetaData, as it was not specified by JDBC it was a InterClient extension to the specification. With JDK 1.4 this is now a part of JDBC 3.0 specification and InterClient surfaces this as a JDBC call. In order to use this new JDBC specified class users will need to use `java.sql.ParameterMetaData`, the previous class `interbase.interclient.ParameterMetaData` is the older implementation and is now deprecated.

**Note** The usability of these 2 new classes is based on the JDBC 3.0 specification and visiting Sun's JDBC 3.0 web site for detailed information on what these new interfaces include is recommended.

## New in InterBase 7.1

---

InterBase 7.1 includes the following new features:

### New cross-platform installer

InterBase 7.1 has a new all-Java installer that is available for all InterBase platforms. For information about installation, please see the *IBsetup.html* file located at the root of your InterBase CD-ROM or download file. This file is also accessible from the main screen of the installer.

### New registration

InterBase now joins other Borland products in requiring product registration. As part of the install process, you are asked to register and will be given four options for doing this, discussed below.

**Preparing for the install and registration** If you have already completed the InterBase installation and registration, you can skip this section. Before you begin the installation process have the following handy:

- The Serial Number and Key that are provided on the jacket of your InterBase CD-ROM.
- Your Borland Developer Network membership information. You can supply either your BDN user name or the email that you used to sign up plus the password. If you have ever registered a Borland product, you are a member of the BDN network. In addition, you might have signed up for membership on one of the Borland websites. If you are not a member of the Borland Developer Network, you have an opportunity to join during the registration process.

*IBsetup.html* provides detailed information about the installation and registration process.

## Precision of exact numerics

InterBase now returns the precision of exact numeric datatypes back to the client using the XSQLDA structure.

## New drivers

For Windows platforms, InterBase 7.1 includes a Borland Data Provider (BDP) for ADO.NET programming.

**Old drivers** The IBX drivers for Delphi 5, Delphi 6, and C++Builder 5 are still available on your InterBase 7.1 CD-ROM, but they are no longer listed on the driver install menu.

## New ODS

InterBase 7.1 uses ODS 11.1 rather than the ODS 11.0 used by InterBase 7.0. This new ODS is required to accommodate reporting the precision of exact numerics. To migrate databases, back them up with the older version of InterBase and restore them using InterBase 7.1 IBConsole or **gbak**.

## Savepoints

InterBase 7.1 implements savepoints as defined in the SQL 1999 standard.

### Savepoints in SQL

In DSQL and ESQL the following SQL statements are available:

- 1 To create a savepoint:

```
SAVEPOINT <savepoint_name>
```

A savepoint name can be any valid SQL identifier. Savepoint names must be unique within their atomic execution context. If you assign a name that is already in use, the existing savepoint is released and the name is applied to the current savepoint. An application, for example, is an execution context, as is each trigger and stored procedure. Thus, if you have an application with several triggers, you can have a savepoint named SV1 within the application and also within each trigger and stored procedure.

- 2 To release a savepoint:

```
RELEASE SAVEPOINT <savepoint_name>
```

Releasing a savepoint destroys that savepoint without affecting any work that has been performed subsequent to its creation.

- 3 To roll back to a savepoint:

```
ROLLBACK [WORK] [TO SAVEPOINT <savepoint_name>]
```

Issuing a ROLLBACK TO SAVEPOINT command rolls back all work performed since the creation of the named savepoint. If other savepoints were created after the named savepoint, those later savepoints are also rolled back.

### Savepoints in the InterBase API

The InterBase API supports savepoints with the following functions:

- 1 To create a savepoint:

```
ISC_STATUS isc_start_savepoint(
    ISC_STATUS *status_vector,
    isc_tr_handle *trans_handle,
    char *savepoint_name);
```

- 2 To release a savepoint:

```
ISC_STATUS isc_release_savepoint(
    ISC_STATUS *status_vector,
```



```
isc_tr_handle *trans_handle,
char *savepoint_name);
```

### 3 To roll back to a savepoint:

```
ISC_STATUS isc_rollback_savepoint(
ISC_STATUS *status_vector,
isc_tr_handle *trans_handle,
char *savepoint_name
short option);
```

The *option* parameter is reserved for future use. Pass a value of zero for this parameter.

### **Savepoints in triggers and stored procedures**

Savepoints are implemented in stored procedures and triggers.

#### **A SAVEPOINT example**

The following code snippet is a simple example of how to use savepoints:

```
CREATE PROCEDURE add_emp_proj2 (emp_no SMALLINT, emp_name VARCHAR(20), proj_id CHAR(5)) AS
BEGIN
BEGIN
SAVEPOINT EMP_PROJ_INSERT;
INSERT INTO employee_project (emp_no, proj_id) VALUES (:emp_no,:proj_id);
WHEN SQLCODE -530 DO
BEGIN
ROLLBACK TO SAVEPOINT EMP_PROJ_INSERT;
EXCEPTION unknown_emp_id;
END
END
SUSPEND;
END;
```

### **New keywords**

The savepoint functionality adds the following new keywords:

```
SAVEPOINT          RELEASE
```

### **Performance monitoring now accessible in IBConsole**

You can now access the performance monitoring features that were introduced in InterBase 7.0 through IBConsole, the graphical Windows interface for InterBase.

### **New character sets**

InterBase 7.1 implements several new character sets and collation orders.

For the Latin 2 character set, InterBase implements Polish and Czech. More languages will be implemented in the future. A longish list of languages is implemented for Latin 9, listed in the table below. Finally, Russian is implemented for the KOI8-R character set.

These new character sets are defined as follows:

Character set	Char. set ID	Max. char. size	Min. char. size	Collation orders
ISO8859_2 (Latin2)	22	1 byte	1 byte	ISO8859_2 CS_CZ PL_PL
ISO8859_15 (Latin9)	39	1 byte	1 byte	ISO8859_15 DA_DA9 DE_DE9 DU_NL9 EN_UK9 EN_US9 ES_ES9 FI_FI9 FR_CA9 FR_FR9 IS_IS9 IT_IT9 NO_NO9 PT_PT9 SV_SV9
KOI8-R	58	1 byte	1 byte	RU_RU

**Note** Databases can optionally have a default character set defined for them. Character sets can also optionally be defined for specific table columns. If you are connecting to a database from a platform whose default code page is different from that of the database you are connecting to, you must specify the default code page of the client platform when making the connection to the database. To do this from IBConsole, select the database from the Tree Pane and choose Connect As from the Connect menu or the mouse context menu. The resulting Database Connect dialog box contains a Character Set field where you can specify the client platform character set from the pull-down list.

## Enhanced SMP support

Support for multiprocessor machines has been improved. Among other changes, the `MAX_THREADS` parameter in the `ibconfig` configuration file now defaults to 1,000,000 when two or more CPUs are present and licensed. This means that there are never threads waiting to execute and improves the speed with which they release any resources that they hold. When only one CPU is licensed or if only one CPU is present, `MAX_THREADS` defaults to 1. For the purpose of determining this default value, InterBase counts a hyperthreaded processor as a single CPU. You can change the number of simultaneous active server threads by editing the `MAX_THREADS` entry in the `ibconfig` configuration file.

## Hyperthreading support on Intel processors

InterBase can support hyperthreading on Intel processors that support logical processors using Intel's hyperthreading technology. To enable this support in the InterBase server, you must make a setting in the InterBase configuration file, `ibconfig`. If you are running the InterBase server on a machine with hyperthreaded processors, edit the `ENABLE_HYPERTHREADING` parameter in the configuration file. By default, this parameter is set to zero. Set the value to 1 to allow the InterBase server to use hyperthreaded processors.

## Change in gbak functionality

When restoring a database, `gbak` no longer automatically performs constraint checking the database during the restore process. This improves the speed of database restores and ensures that users can always restore their databases from backup files even when the backup files contain data that violates constraints such as `NOT NULL`, `CHECK`, `PRIMARY` and `UNIQUE` indexes, or `REFERENTIAL` constraints.

InterBase 7.1 provides new switches and parameters to provide the former capability of validating a database when restoring it.

<i>Command line</i>	There is a new command-line switch: <code>-VA[LIDATE]</code> . For example: <code>gbak -r -user joe -pass blurf@ C:\archive\foo.ibk jupiter:/foo.ib -validate</code>
<i>InterBase Services API</i>	There is a new parameter that enables validation during a restore: <code>isc_spb_res_validate</code> .
<i>DPB</i>	There is a new DPB parameter, <code>isc_dpb_gbak_validate</code> that instructs the server to include validation checks during a database restore.

## Hyperthreading support for Intel processors

InterBase now recognizes and responds to hyperthreading technology in Intel processors. In InterBase 7.0, there was no way to exploit hyperthreading without purchasing additional SMP licenses. InterBase 7.1 now unlocks the additional processing power of hyperthreading transparently, without requiring additional SMP licenses. These same changes also insure that an SMP license is applied to a physical processor and not a logical processor, for maximum price/performance benefit.

## New SQL command: DROP GENERATOR

InterBase now supports a DROP GENERATOR SQL statement:

```
DROP GENERATOR generator_name
```

The statement fails if *generator\_name* is not the name of a generator defined on the database. This command checks for any existing dependencies on the generator—for instance in triggers or UDFs—and fails if such dependencies exist. An application that tries to call a deleted generator returns runtime errors.

DROP GENERATOR is implemented for DSQL and `isql`.

In previous versions of InterBase that lacked the DROP GENERATOR command, users were told to issue a SQL statement to delete the generator from the appropriate system table. This approach is strongly discouraged now that the DROP GENERATOR command is available, since modifying system tables always carries with it the possibility of rendering the entire database unusable as a result of even a slight error or miscalculation.

## Enhanced garbage collection/index handling

Users will see significant performance improvement as a result of InterBase 7.1's more efficient garbage collection of duplicate index nodes. New algorithms have been added that minimize computational overhead and memory consumption during garbage collection.

## IBConsole displays additional object dependencies

IBConsole now displays object dependencies on generators and UDFs in addition to all the dependencies it formerly displayed.

## Using the InterBase Install API

Certain components of the InterBase Install API point to an InterBase file structure that is no longer in use. If you are writing or updating an install application using this API, you need to have the current InterBase files arranged in the file structure required by the InterBase Install API. To facilitate this, InterBase supplies a file, *silent\_install.zip*, that contains all the current files arranged in the structure required by the API.

If you are writing an install application, extract *silent\_install.zip*, and place your compiled install application at the root of the resulting file structure.

If you have an existing install application that does not bundle InterBase files within the binary, you can update it by just extracting *silent\_install.zip*, and copying the resulting files over the InterBase file structure that you previously used.

If your existing install application includes changed InterBase files within the binary, you need to refresh the file structure with the files in *silent\_install.zip*, and then recompile the application.

*Licensing:* VARs are now provided with an additional file, which must be included with their InterBase installs in order to provide valid product registration. If you are embedding or reselling InterBase, you should have received instructions about how to manage this file. If you need more information, contact your Borland InterBase representative.

## Documentation fixes and changes

Some errors have been corrected in the documentation for InterBase 7.1. These changes are included in the PDF documents that ship with InterBase 7.1. They are not yet included in the printed documents.

### ***UDF library documentation has been moved***

In order to make all UDF information available in one place, the UDF chapter has been removed from the *Language Reference* and folded into the “Working with UDFs and Blob Filters” chapter of the *Developer’s Guide*.

### ***Declaring Blob UDFs***

The documentation now contains a more complete description of how to declare a UDF that returns a Blob.

To specify that a UDF should return a Blob, use the RETURNS PARAMETER *n* statement to specify which input Blob is to be returned. For example, if the Blob to be returned is the third input parameter, specify RETURNS PARAMETER 3. The Blob\_PLUS\_Blob UDF concatenates two Blobs and returns the concatenation in a third Blob. The following statement declares this UDF to a database, specifying that the third input parameter is the one that should be returned:

```
DECLARE EXTERNAL FUNCTION Blob_PLUS_Blob
  Blob,
  Blob,
  Blob
  RETURNS PARAMETER 3
  ENTRY_POINT 'blob_concatenate' MODULE_NAME 'ib_udf';
COMMIT;
```

For more information about UDFs and Blobs, see the chapter “Working with UDFs and Blob Filters” in the *Developer’s Guide*.

### ***Calling convention for UDFs***

Previous versions of InterBase documentation said that UDFs should be called using `_stdcall`. This is not correct. InterBase uses the CDECL calling convention, so all UDFs must be declared using the CDECL calling convention.

### ***Portable UDFs***

It has always been the case that UDFs could be written for Unix and Linux platforms as well as for Windows. However, the documentation did not make this clear. This has now been corrected. In addition, examples of declaring UDFs now show the portable form, in which the extension of the module name is not included. For example:

```
DECLARE EXTERNAL FUNCTION LOWERS VARCHAR(256)
  RETURNS CSTRING(256) FREE_IT
  ENTRY POINT 'fn_lower' MODULE_NAME 'udflib';
```

### ***Correction for YEARDAY range***

In the *Language Reference*, the range for EXTRACT(YEARDAY) should be 0–365. This will be corrected in the next version of the InterBase document set. It is incorrect in the set that ships with InterBase 7.1.

## New in InterClient 4.0

### **DataSource** properties for InterBase

#### **Standard properties**

**Table 2.1** *DataSource* standard properties

Name	Type	Description	Default Value
<i>databaseName</i>	String	The name of the database to connect to	null
<i>serverName</i>	String	The InterBase server name	localhost
<i>user</i>	String	The InterBase user who is connecting	null
<i>password</i>	String	The InterBase user password	null
<i>networkProtocol</i>	String	The InterBase network protocol; this can only be jdbc:interbase: for InterClient.	jdbc:interbase
<i>port Number</i>	int	The InterBase port number	3050
<i>roleName</i>	String	The InterBase role	null
<i>dataSourceName</i>	String	The logical name for the underlying XADataSource or Connection Pool; used only when pooling connections for InterBase (XA is not supported)	null
<i>description</i>	String	A description of this data source	null

#### **Extended properties**

**Table 2.2** *DataSource* Extended properties

Name	Type	Description	Default Value
<i>charSet</i>	String	<p>Specifies the character encoding for the connection; used for sending all SQL and character input data to the database and for all output data and InterBase messages retrieved from the database.</p> <p>The encoding specified by <i>charSet</i> must match one of the supported IANA character-encoding names detailed in the <i>CharacterEncodings</i> class.</p> <p>If <i>charSet</i> is set to <i>NONE</i>, InterClient uses the default system encoding obtained by the <i>System.getProperty("file.encoding")</i> method if that default encoding is supported by InterBase. If the default system encoding is not supported by InterBase, it is recommended that you use the <i>charSet</i> property to set the InterClient <i>charSet</i> to one of the InterBase-supported encodings.</p> <p>InterClient messages do not utilize <i>charSet</i>, but derive from the resource bundle in use, which is based on the locale-specific encoding of the client.</p>	No default value
<i>sqlDialect</i>	int	The client SQL dialect. If the value is set to 0 then the database's dialect is used for the client dialect.	0
<i>createDatabase</i>	Boolean	If set, the database is created if it does not exist.	false
<i>serverManagerHost</i>	String	Ignored.	null

**Table 2.2** *DataSource* Extended properties (*continued*)

Name	Type	Description	Default Value
<i>sweepOnConnect</i>	boolean	If set, forces garbage collection of outdated record versions immediately upon connection  See the <i>InterBase Operations Guide</i> for more details. Sweep does not require exclusive access, but there is some data and transaction state information that can be updated only where there are no active transactions on the database.	false
<i>suggestedCachePages</i>	int	The suggested number of cache page buffers to use for this connection  This is a transient property of the connection and is overridden by the database-wide default set by <i>ServerManager.setDatabaseCachePages(database, pages)</i> . It takes precedence over the server-wide default set by DATABASE_CACHE_PAGES in the InterBase <i>ibconfig</i> startup file or by <i>ServerManager.startInterBase(defaultCachePages, defaultPageSize)</i> .  On SuperServer, if a database cache already exists due to another attachment to the database, then the cache size can be increased but not decreased. So, although this is a transient property, once the cache size is increased, it stays that way as long as there are active connections. Once all connections to the database are closed, then subsequent connections use the database-wide or server-wide defaults.  Note: Using this connection property can jeopardize the performance of the server because an arbitrary user can connect and reserve 200MB for <i>foo.ib</i> while <i>corporate.ib</i> is forced to accept less.  InterBase code sets an absolute limitation on MAX_PAGE_BUFFERS of 65,535 pages. So the cache memory size for a database cannot go beyond a maximum of MAX_PAGE_BUFFERS* <i>PageSize</i> bytes, which is 512MB for an 8K page size. 8K is the maximum database page size currently allowed. If this property is zero or unspecified and there is no server-wide or database-wide default set, the default pages used is 2048 cache pages.  Also see <i>DatabaseMetaData.getPersistentDatabaseCachePages()</i> , and <i>DatabaseMetaData.getActualCachePagesInUse()</i> .	0

## InterClient connection pooling

InterClient now works with Container Managed Persistence (CMP) 2.0, which is supplied with the Borland Enterprise Server. This enables JDBC *DataSource* 2.x connectivity to InterBase databases. The following *jndi-definition.xml* file shows how it can be used through an application server:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE jndi-definitions PUBLIC "-//Borland Corporation//DTD JndiDefinitions//EN" "http://www.borland.com/devsupport/appserver/dtds/jndi-definitions.dtd">
<jndi-definitions>
  <visitransact-datasource>
    <jndi-name>serial://datasources/DataSource</jndi-name>
    <driver-datasource-jndiname>serial://datasources/driverDataSource</driver-datasource-jndiname>
    <property>
      <prop-name>connectionType</prop-name>
      <prop-type>Enumerated</prop-type>
      <prop-value>Direct</prop-value>
    </property>
  </visitransact-datasource>
</jndi-definitions>
```

```

    <property>
      <prop-name>dialect</prop-name>
      <prop-type>Enumerated</prop-type>
      <prop-value>interbase</prop-value>
    </property>
  </visittransact-datasource>
<driver-datasource>
  <jndi-name>serial://datasources/driverDataSource</jndi-name>
  <datasource-class-name>interbase.interclient.JdbcConnectionFactory</datasource-class-name>
  <property>
    <prop-name>user</prop-name>
    <prop-type>String</prop-type>
    <prop-value>SYSDBA</prop-value>
  </property>
  <property>
    <prop-name>password</prop-name>
    <prop-type>String</prop-type>
    <prop-value>masterkey</prop-value>
  </property>
  <property>
    <prop-name>serverName</prop-name>
    <prop-type>String</prop-type>
    <prop-value>agni</prop-value>
  </property>
  <property>
    <prop-name>databaseName</prop-name>
    <prop-type>String</prop-type>
    <prop-value>c:/admin.ib</prop-value>
  </property>
  <property>
    <prop-name>sqlDialect</prop-name>
    <prop-type>int</prop-type>
    <prop-value>3</prop-value>
  </property>
  <property>
    <prop-name>createDatabase</prop-name>
    <prop-type>boolean</prop-type>
    <prop-value>true</prop-value>
  </property>
</driver-datasource>
</jndi-definitions>

```

## InterClient scrollability

### **The Connection class**

To achieve JDBC 2.0 core compliance, InterClient now allows a value of `TYPE_SCROLL_INSENSITIVE` for the `resultSetType` argument for the following `Connection` methods:

```
public java.sql.Statement createStatement (int resultSetType, int resultSetConcurrency)
```

```
public java.sql.CallableStatement prepareCall (String sql, int resultSetType, int resultSetConcurrency)
```

```
public java.sql.PreparedStatement prepareStatement (String sql, int resultSetType, int resultSetConcurrency)
```

Previously, the only allowable value for `resultSetType` was `TYPE_FORWARD_ONLY`. Currently, the only type not allowed is the `TYPE_SCROLL_SENSITIVE`

**The ResultSet class**

The `resultSetType` property of the `ResultSet` class can now have a value of `TYPE_SCROLL_INSENSITIVE`. Previously, the only allowable value for `resultSetType` was `TYPE_FORWARD_ONLY`. Currently, the only type not allowed is the `TYPE_SCROLL_SENSITIVE`.

The following methods now return a valid value when the resultSets that are of the new `resultSetType.TYPE_SCROLL_INSENSITIVE`:

```
public boolean isBeforeFirst()
public boolean isAfterLast()
public boolean isFirst()
public isLast()
public void beforeFirst()
public void afterLast()
public boolean first()
public boolean last()
public int getRow()
public boolean absolute(int row)
public boolean relative(int rows)
public boolean previous()
```

**New InterClient methods**

`InterClient` is `InterBase`'s JDBC driver. In `InterBase 7.0`, we introduced a new type 4 JDBC driver. For `InterBase 7.1`, we have added a large collection of methods to this driver to bring it into compliance with the JDBC 2.0 standard.

**Methods for the Statement and PreparedStatement classes**

The following methods have been added to both the `Statement` and the `PreparedStatement` classes. The methods listed below now work according to the JDBC specifications.

**Table 2.3** Methods for the `Statement` and `PreparedStatement` classes

Method	Functionality
<code>void Statement.addBatch(String sql)</code>	Adds <code>sql</code> to the current list of commands.
<code>void Statement.clearBatch()</code>	Empties the list of commands for the current statement object.
<code>int[] Statement.executeBatch()</code> throws <code>BatchUpdateException</code>	Submits the list of commands for this statement's objects to the database for execution as a unit. The returned integer array contains the update counts for each of the SQL commands in the list.
<code>void PreparedStatement.addBatch()</code>	Adds a set of parameters to the list of commands for the current <code>PreparedStatement</code> object's list of commands to be sent to the database for execution.



### The `BatchUpdateException` class

A new `BatchUpdateException` class has been implemented in order to support JDBC Batch update functionality. Here is the list of methods and constructors in the new class:

**Table 2.4** Methods and constructors for the new `BatchUpdateException` class

Method/Constructor	Functionality
<pre>public BatchUpdateException(     String reason,     String SQLState,     int vendorCode,     int [] updateCounts)</pre>	<p>Constructs a <code>BatchUpdateException</code> object where:</p> <ul style="list-style-type: none"> <li>• <i>reason</i> is a string describing the exception,</li> <li>• <i>SQLState</i> is an object containing Open Group code identification,</li> <li>• <i>vendorCode</i> identifies the vendor-specific database error code</li> <li>• <i>updateCounts</i> contains an array of INTs where each element indicates the row count for each SQL UPDATE command that executed successfully before the exception was thrown.</li> </ul>
<pre>public BatchUpdateException(     String reason,     String SQLState,     int [] updateCounts)</pre>	<p>Constructs a <code>BatchUpdateException</code> object where:</p> <ul style="list-style-type: none"> <li>• <i>reason</i> is a string describing the exception,</li> <li>• <i>SQLState</i> is an object containing the InterBase error code</li> <li>• <i>updateCounts</i> contains an array of INTs where each element indicates the row count for each SQL UPDATE command that executed successfully before the exception was thrown.</li> <li>• The vendor code is implicitly set to zero.</li> </ul>
<pre>public BatchUpdateException(     String reason,     int [] updateCounts)</pre>	<p>Constructs a <code>BatchUpdateException</code> object where:</p> <ul style="list-style-type: none"> <li>• <i>reason</i> is a string describing the exception,</li> <li>• <i>updateCounts</i> contains an array of INTs where each element indicates the row count for each SQL UPDATE command that executed successfully before the exception was thrown.</li> <li>• The following values are implicitly set: the <i>vendorCode</i> is set to zero and the Open Group code identification is set to null.</li> </ul>
<pre>public BatchUpdateException(int [] updateCounts)</pre>	<ul style="list-style-type: none"> <li>• Constructs a <code>BatchUpdateException</code> object where <i>updateCounts</i> contains an array of INTs in which each element indicates the row count for each SQL UPDATE command that executed successfully before the exception was thrown.</li> <li>• The following values are implicitly set: <i>reason</i> is set to null, <i>vendorCode</i> is set to zero, and the Open Group code identification is set to null.</li> </ul>
<pre>public BatchUpdateException()</pre>	<p>The following values are implicitly set:</p> <ul style="list-style-type: none"> <li>• <i>updateCounts</i> is set to a zero-length integer array,</li> <li>• <i>reason</i> is set to null,</li> <li>• <i>vendorCode</i> is set to zero,</li> <li>• the Open Group code identification is set to null.</li> </ul>
<pre>public int [] getUpdateCounts()</pre>	<p>Retrieves an array of INTs where each element indicates the row count for each SQL UPDATE command that executed successfully before the exception was thrown.</p>

**The DatabaseMetaData.supportsBatchUpdates function**

The `DatabaseMetaData.supportsBatchUpdates` function has changed as follows:

Function	Functionality
<code>boolean DatabaseMetaData.supportsBatchUpdates()</code>	Can now return TRUE.

**Additional functions**

Additional functions that implement the JDBC 2.x API functionality are listed below.

Function	Functionality
<code>int Statement.getResultSetType()</code>	Returns the type if <code>resultSet</code> is open, otherwise throws an exception
<code>int Statement.getResultSetConcurrency()</code>	Returns the concurrency if <code>resultSet</code> is open.
<code>int Statement.getFetchDirection()</code>	Returns the fetch direction if <code>resultSet</code> is open, the return value is always <code>FETCH_FORWARD</code> for InterBase.
<code>int ResultSet.getFetchDirection()</code>	Returns <code>FETCH_FORWARD</code> in all cases
<code>int ResultSet.getFetchSize()</code>	Returns the fetch size for the statement's result set.
<code>int ResultSet.setFetchSize()</code>	Allows you to set the fetch size of the resultset and the statement.
<code>int ResultSet.setFetchDirection()</code>	Throws an exception; it can only work with <code>TYPE_SCROLL_SENSITIVE</code> and <code>TYPE_SCROLL_INSENSITIVE</code> . Neither of these are supported by InterBase, since InterBase does not support scrollable cursors. The only <code>ResultSet</code> type allowed by InterClient/InterBase is <code>TYPE_FORWARD_ONLY</code> .

**Code examples**

Code example for the batch update functions:

```
Statement con = con.createStatement();
con.setAutoCommit(false);
Statement stmt = con.createStatement();
stmt.addBatch("INSERT INTO foo VALUES (1, 10)");
stmt.addBatch("INSERT INTO foo VALUES (2, 21)");
int[] updateCounts = stmt.executeBatch();
con.commit();
```

Code example for the `PreparedStatement` class:

```
PreparedStatement pstmt = con.prepareStatement("UPDATE employee set emp_id = ? where emp_id = ?")
pstmt.setInt(1, newEmpId1);
pstmt.setInt(2, oldEmpId1);
pstmt.addBatch();
pstmt.setInt(1, newEmpId2);
pstmt.setInt(2, oldEmpId2);
pstmt.addBatch();
int[] updateCounts = pstmt.executeBatch();
```

Code example for the *BatchUpdateException* class and *getUpdateCounts()* method

```
try
{
    int[] updateCounts = pstmt.executeBatch();
}
catch (BatchUpdateException b)
{
    int [] updates = b.getUpdateCounts();
    for (int i = 0; i < updates.length; i++)
    {
        System.err.println ("Update Count " + updates[i]);
    }
}
```

## InterClient and the Borland Enterprise Server

InterClient 4 now works with Container Managed Persistence (CMP) 2.0, which is provided with Borland Enterprise Server (BES) 5.x.

## Other InterClient enhancements

- The JDBC Timestamp data type now matches the InterBase SQL TIMESTAMP data type and allows fractions of seconds.
- The *Resources\_ru.class* has been removed from the *interclient.jar* file to enhance code page flexibility on non-Windows machines in Russian character sets.

## New in InterBase 7.0

---

As a reminder, or for those of you who may have missed the InterBase 7.0 release, the following is a list of features that were new in InterBase 7.0, with a brief description of each.

- Database naming

InterBase no longer recommends using “.gdb” as the extension for database files, since on Windows ME and Windows XP, any file that has this extension is automatically backed up by the System Restore facility. InterBase now recommends using “.ib” as the extension for database names.

Our security database, formerly named *isc4.gdb* is now named *admin.ib*. For the present, the InterBase example databases still have the “.gdb” name. In the future we will phase out that name and use new names.

- ODS11

InterBase 7.0 introduces ODS11. This new On-Disk Structure is required by the presence of the new BOOLEAN datatype and 68-byte metadata names. To upgrade your databases, back them up with an ODS10 **gbak** and then restore them with the ODS11 **gbak** that comes with the InterBase 7.

- New name for the security database

In InterBase 7, InterBase’s security database is named *admin.ib* on all platforms. InterBase’s internal tools have all been updated to use this name. If you wish to continue using your existing security database, you must back it up and restore it using the latest **gbak**. To change the name, specify *admin.ib* as the new name during the restore. If you have existing clients that expect to find *isc4.gdb*, you must update them to use the new name.

You can specify a name of your choice for the security database by setting the ADMIN\_DB parameter in the InterBase configuration file, *ibconfig*.

- New name for Unix configuration file

On Linux and Solaris platforms, the InterBase configuration file was previously called *isc\_config*. It is now called *ibconfig*.

- New keywords

InterBase 7.0 adds the following new keywords:

```
BOOLEAN      TRUE          FALSE          UNKNOWN
```

The following keywords were added to InterBase 6.5:

```
ROWS         TIES          PERCENT
```

- New datatype: BOOLEAN

InterBase now supports a BOOLEAN datatype, implemented to the SQL 99 standard.

Examples:

```
CREATE TABLE AWARDS_1 (isEligible BOOLEAN, name VARCHAR(20));
INSERT INTO AWARDS_1 VALUES(TRUE, 'Jim Smith');
INSERT INTO AWARDS_1 VALUES(FALSE, 'John Butler');

SELECT * FROM AWARDS_1 WHERE isEligible = TRUE;
```

ISQL and IBConsole return TRUE, FALSE, and UNKNOWN. Queries created with APIs return 1, 0, and NULL, respectively. For ESQL and DSQL programmers, we define the following type in *ibase.h*:

```
define SQL_BOOLEAN 590
```

**Note** BOOLEAN is not supported in GPRE.

- No more SET TERM

When you write SQL, there is no longer any need to use SET TERM to define a temporary terminator when defining stored procedures and triggers. InterBase now parses these statements correctly without the use of SET TERM.

The document set states that IBConsole and IBX still require the use of SET TERM. InterBase believes that this was corrected after the document set was written and that SET TERM is no longer required in **isql**, IBConsole, or IBX. The old SET TERM functionality remains available in **isql**, IBConsole, and IBX, so that old scripts can still function.

- 68-byte metadata names and XSQLDA

Metadata names can now be 68 bytes long (67 bytes plus a null terminator). These names are available through all InterBase clients and are implemented in the new type 4 InterClient. They are being implemented in DBX and IBX and may be available by the time you read this.

The XSQLDA structure has been updated to support these long metadata names. Set the version field of this structure to SQLDA\_CURRENT\_VERSION to access long metadata names.

- New APIs for blobs and arrays

Ten API calls that relate to blobs and arrays have been updated to support these longer metadata names. In these new APIs, the *desc* field points to an updated descriptor structure that accommodates long metadata names.

The new API calls are:

<i>isc_array_gen_sdl2()</i>	<i>isc_array_get_slice2()</i>
<i>isc_array_lookup_bounds2()</i>	<i>isc_array_lookup_desc2()</i>
<i>isc_array_set_desc2()</i>	<i>isc_array_put_slice2()</i>
<i>isc_blob_default_desc2()</i>	<i>isc_blob_gen_bpb2()</i>
<i>isc_blob_lookup_desc2()</i>	<i>isc_blob_set_desc2()</i>

The associated structure for arrays is `ISC_ARRAY_DESC_V2`. For blobs it is `ISC_BLOB_DESC_V2`. The associated defines are:

```
#define BLB_DESC_VERSION2 2
#define BLB_DESC_CURRENT_VERSION BLB_DESC_VERSION2
#define ARR_DESC_VERSION2 2
#define ARR_DESC_CURRENT_VERSION ARR_DESC_VERSION2
```

These new API calls and their structs are documented in the *API Guide*. See Chapter 7, Chapter 8, and the new API calls in the API Reference chapter.

- Client version detection

Some clients—notably drivers, but others as well—need to query the InterBase client library for the version numbers. Three new APIs provide this capability: *isc\_get\_client\_version()*, *isc\_get\_client\_major\_version()*, and *isc\_get\_client\_minor\_version()*. They are described in detail in the API Function Reference chapter of the *API Guide*.

- New type 4 InterClient

InterBase 7.0 introduces InterClient 3.0. This new version of InterClient is a type 4 JDBC driver, which means that it can communicate directly with the InterBase server. InterServer is no longer needed in environments where all the clients have been upgraded to this new type 4 InterClient.

To upgrade a client, place the new *interclient.jar* file on each client machine and ensure that it is the first instance on the CLASSPATH. Note that although the filename is the same as it was in earlier versions, the file being distributed with InterBase 7 is very different. It is a type 4 JDBC driver. Earlier versions were type 3. Be sure that you are installing the latest driver on your client machines.

Legacy InterServer: If you are not able to upgrade all of your clients immediately, install InterServer on the InterBase Server platform. InterServer will do no harm, and its presence allows a mixture of type 3 and type 4 clients to attach to the InterBase server. The *interserver.exe* file that distributes with InterBase 7.0 has not changed from previous versions.

- SMP support

InterBase now provides symmetric multiprocessor (SMP) support for both clients and servers. Previous versions of InterBase ran on SMP systems safely by allowing only a single processor at a time to execute within the InterBase components. This release exploits SMP hardware by running InterBase threads on all processors simultaneously for increased throughput and performance.

When you purchase a single server license, you acquire the right to use a single processor. You must purchase an additional license for each additional processor that you wish to use.

- Server configuration parameter: `MAX_THREADS`

Setting the `MAX_THREADS` parameter in the *ibconfig* configuration file controls the maximum number of threads that can be active at one time within the InterBase engine. The default setting is 100:

The ideal setting for this number depends partly on the nature of the work being performed by your clients. If you have many clients performing very similar tasks, you may want to lower the `MAX_THREADS` setting to reduce contention. On the other hand, if simultaneous activity is highly diverse, setting this to a higher value may increase throughput. This setting does not affect the maximum possible threads that can be created by the InterBase server but only the number that can be active in the engine at one time.

- Expanded processor control: `CPU_AFFINITY`

On Windows multiprocessor platforms, you can specify which processors InterBase should use by adding the `CPU_AFFINITY` parameter to the `ibconfig` file. This setting is useful whenever the number of licensed processors is less than the number of actual processors present. When you purchase a single server license, you acquire the right to use a single processor. You must purchase one additional license for each additional processor that you wish to use.

`CPU_AFFINITY` is discussed in the “Server Configuration” chapter of the *Operations Guide*.

- Increased security for external tables

Under some conditions, external tables could pose a security hazard. To counter this, InterBase has added the new requirements for external tables. External tables must meet one of the following conditions:

- The table is located in `<ib_home>/ext`. InterBase can always find external files that you place here.
- The location of the table is specified in the `ibconfig` configuration file by setting the `EXTERNAL_FILE_DIRECTORY` parameter to the location of the external file.

- New HTML reference docs

InterBase now includes two references in HTML form. Both of these can be accessed from the Help menu of IBConsole or directly from the `<ib_home>/HtmlRef` directory.

- **The SQL Reference** The `SqlRef.html` file replaces the older `SqlRef.hlp` file. It contains all the SQL statement information from the “SQL Statement and Function Reference” chapter of the *Language Reference*.
- **The API Function Reference** The `APIFunctionRef.html` file is an HTML version of the “API Function Reference” chapter of the *API Guide*.

- Monitoring database attachments with system temporary tables

The InterBase Server has always kept a lot of statistics about what was going on, but it has not been easy, or in some cases possible, to surface that information. InterBase now captures that information and makes it available in a set of global system temporary tables. These tables describe the runtime behavior of a database. They also provide a level of control. The temporary table metadata is listed on pages 6-32 to 6-45 of the *Language Reference*.

It is also possible to exercise a certain amount of control over the state of a database by performing updates to these tables.

Table name	Description
<code>TMP\$ATTACHMENTS</code>	One row for each connection to a database
<code>TMP\$DATABASE</code>	One row for each database you are attached to
<code>TMP\$POOL_BLOCKS</code>	One row for each block of memory in each pool
<code>TMP\$POOLS</code>	One row for each current memory pool
<code>TMP\$PROCEDURES</code>	One row for each procedure executed since the current connection began

Table name	Description
TMP\$RELATIONS	One row for each relation referenced since the current connection began
TMP\$STATEMENTS	One row for each statement currently executing for any current connection
TMP\$TRANSACTIONS	One row for each transaction that is active or in limbo

**Querying system temporary tables** Clients can query these tables using SELECT statements, just as they would query any other table. For frequent monitoring, the best transaction control is to start the transaction as READ\_COMMITTED, READ\_ONLY. Then commit it with COMMIT\_RETAINING. This has the least impact on the system.

**Updating system temporary tables** By updating the TMP\$STATE column of certain temporary tables, you can roll back an active or limbo transaction, commit a limbo transaction, cancel an attachment's executing operation, shut down the current attachment, or make an executing statement stop running.

**System temporary table metadata** The "System Tables" chapter of the *Language Reference* lists the metadata for each of the system temporary tables.

- Thread-safe processing of database handles

InterBase 7.0 provides improved handling of InterBase database handles on behalf of client applications. Handle types include attachment, blob, BLR request, SQL statement, service and transaction objects. The client library manages the integrity of InterBase database handles in the face of concurrent application thread activity.





## Bugfix list

This chapter lists bugs that have been fixed for the last few InterBase releases: InterBase 7.5, 7.1, and 7.0.

### Bugs fixed in InterBase 7.5.

#### For InterBase 7.5:

Bug Number	Description
125851	Looking at PLAN on VIEW gives garbage output and crashes the server.
138748	GRANT SELECT,UPDATE ON TABLE TO PROCEDURE in certain cases does not work.
148790	When a Blob with sub_type 1 is created and used in a VIEW with UNION ALL the sub_type is set incorrectly.
167748	Calling stmt.executeBatch(); on a Statement with an "Empty Batch" throws a Null pointer exception. (InterClient)
167944	Registration on Linux puts negative numbers in INTERBASE.LOG. (InterBase 7.1 Service Pack 1)
168847	Too many foreign keys in a single script causes a "request depth exceeded" error.
169948	SELECT from TMP\$PROCEDURES after dropping a stored proc crashes the server. (InterBase 7.1 Service Pack 2)
171642	Optimizer is not picking the optimal indices when a select has a conjunct expression. (InterBase 7.1 Service Pack 2)
172453	Closing a database on a ReadWrite medium where the device has been made Read Only causes a hang when closing the database. (InterBase 7.1 Service Pack 2)
173750	IBINSTALL.DLL has wrong version in version resource. (InterBase 7.1 Service Pack 1)
174698	Statement.cancel() does not cancel long running statement. (InterClient)
177999	Restore of database generates error.
178624	GPRESQL program causes crash on SMP.
178971	ODBC driver install on Linux does not install ODBC drivers. (InterBase 7.1 Service Pack 2)
179926	PLAN syntax don't work with InterBase 7.1 Service Pack 1 version.
180322	Insert into large database causes corruption of indexes as shown by gfix -v -full. (InterBase 7.1 Service Pack 2)

**For InterBase 7.5:**

<b>Bug Number</b>	<b>Description</b>
180812	Each time a client tries to connect to a server, if the server name is wrong, or the system cannot resolve the host name, small amounts of memory can be leaked by the application attempting to make the connection.
183158	license.txt updated.
186832	Hyperthreaded CPUs not recognized. (InterBase 7.1 Service Pack 2)
188209	ODBC Drivers install via console option on Solaris installs all files to /tmp. (InterBase 7.1 Service Pack 2)
188211	GUI install of ODBC on Solaris halts near the end of the install.
190062	Cascading update causes server to crash.
191437	java.sql.DatabaseMetadata.getImportedKeys() returns space padded values for the FK_NAME column. (InterClient)
191438	java.sql.DatabaseMetaData.getTypeInfo() returns REAL for FLOAT fields. (InterClient)
191537	The java.sql.ResultSetMetaData will return 1111 (OTHER) for an array column. (InterClient)
191852	Wrong index reported on constraint violation discovery. (InterBase 7.1 Service Pack 2)
191853	Client disconnect while running stored procedure causes server crash. (InterBase 7.1 Service Pack 2)
192483	Index keys lost due to UPDATE rollback and garbage collection. (InterBase 7.1 Service Pack 2)
192584	Wrong error code reported on unique index key violation. (InterBase 7.1 Service Pack 2)
192632	FETCH operation from large resultset is 3 to 30 times slower in IB 6-7.1 clients when compared to IB5.6 client. (InterBase 7.1 Service Pack 2)
194424	DatabaseMetaData.getIdentifierQuoteString() returns "even SQLDialect is not 3. (InterClient)
194728	Custom install does not install ib_license.dat, ibconfig and ibmgr.cpl.
196811	Events not recognized with Windows 2000 Service Pack 4.
198047	Canceling attachment while a query is running no longer causes the server to be hung. (InterBase 6.5, 7.1 Security Update)
198313	0 values stored in Numeric field which have been multiplied by -1 are searched incorrectly when an index is used.
198572	SQL statement with many, many operands causes server to crash.
199674	Hop protocol fixed. (InterBase 6.5, 7.1 Security Update)
200754	InterClient connections with UTF8 character set fail. (InterClient)
201357	Operations which return I/O error for file with InterBase 6, result in a hang in the process when doing the same with InterBase 7.1.
202681	If you install the trial, uninstall the trial, then install Desktop the trial time period will be enforced.
203521	Corruption of non-unique indexes (SP2 applied).
203524	Descending indexes fixed.
204094	Repeatedly fetching blobs with InterClient no longer causes a server memory to grow.
205187	Interbase SuperServer 7.1 SP2 allows passwordless connections on behalf of SYSDBA.
205778	Memory leak in statementhandle. (InterClient)
206449	Make output in log file use long file name.
206633	When doing a long transaction, gfix -shut doesn't work properly.

**For InterBase 7.5:**

<b>Bug Number</b>	<b>Description</b>
206858	Server crashes with complicated query using multi-level views.
206876	Procedure memory leak.
207481	Index corruption issue resolved.
208845	Make output to log file for corrupted index indicate the index name instead of number.
209008	Using a UNION view in a sub-select crashes server.
209390	UpdateConflictException and LockConflictException were documented as being exceptions that could be raised. (InterClient)
209772	interbase.log contains a bunch of 10093 errors when the server is shutdown.
216189	Error codes different for type 4 driver than for type 3 driver. (InterClient)
216785	Group on view causes internal gds software consistency check (attempt to release free block).
216810	Passing null to rows clause causes server crash.
216836	Restoring a database with a bad plan causes server to crash.
216838	Stored Procedures can have invalid plans.
217288	Some International characters don't work in SQL statements. (InterClient)
217539	Heavy index inserts causes internal gds consistency check.

**Bugs fixed in InterBase 7.1****For InterBase 7.1:**

<b>Bug Number</b>	<b>Description</b>
58838	Self joins sometimes produced incorrect results.
58982	UDFs can no longer be dropped if a stored procedure depends on them.
59009	There is now a DROP GENERATOR command.
60175	Command for setting password was given as SET_PASSWORD rather than SET ISC_PASSWORD.
60370	Examples of constructing tpb parameters are now correct in Transactions chapter of API Guide.
60532	Developer's Guide now include section on how to declare a Blob UDF.
64535	Abbreviated forms of user and password switches for gbak are now correct.
76360	gbak can now restore databases that have non 7-bit ASCII characters as the default, or in CHECK values
114436	SHOW GENERATORS now works correctly in isql.
116325	Docs now give correct name and location of the InterBase UDF library.
122246	Continuous alternate execution of two non-SELECT stored procedures now runs without error.
123504	gbak -r -no_validity now works with NOT NULL constraints.

**For InterBase 7.1:**

<b>Bug Number</b>	<b>Description</b>
125652	Declaration for external function <code>ascii_char</code> has been fixed in code and docs.
125988	Queries with <code>ORDER BY</code> and <code>GROUP BY</code> now work as expected when a table is joined with the result set from a stored procedure.
127852	The Easysoft ODBC driver now finds the <code>ib.hlp</code> file.
135054	<code>current_date</code> , <code>current_time</code> , and <code>current_timestamp</code> now work in stored procedures.
139709	Rollback of an update after creating a unique index no longer has the potential to create duplicate keys in the index.
141235	Using a trigger to delete a row prior to an update no longer corrupts the database.
146833	<code>gbak</code> now does not validate a database when restoring unless specifically requested to do so. This makes it possible to restore a database that was corrupt when backed up.
148790	When a Blob with <code>sub_type 1</code> is created and used in a VIEW with <code>UNION ALL</code> the <code>sub_type</code> is set wrong.
149819	<code>SELECT</code> from <code>TMP\$RELATIONS</code> no longer gives consistency check.
151283	Fixed problem with subselects in stored procedures.
151517	<code>CREATE VIEW</code> now allows large metadata names.
152420	Databases can now grow larger than 4GB on Linux.
152515	Fixed problem with using <code>COMPUTED BY</code> with procedural views.
152584	The “Working with Generators” chapter of the Data Definition Guide now describes using <code>GEN_ID()</code> with <code>SELECT</code> .
152760	Outdated references to database file size limits have been removed from the Operations Guide.
152891	Description of the <code>tanh</code> function in the InterBase UDF library has been corrected.
153053	[duplicate bugs] Large merge sorts no longer cause server crash.
153075	<code>RDB\$FIELD_LENGTH</code> now reports correct value for <code>VARCHAR</code> fields after <code>ALTER TABLE</code> .
153369	InterBase client event registration across a firewall now works.
155890	Generators now restore with correct values after backup even if there are spaces in the generator name.
155995	A <code>NULL</code> in a date column could sometimes crash the server; fixed.
156443	Fixed problem with <code>CREATE TABLE</code> statements using arrays of more than ten dimensions.
157405	InterBase connection speed has been improved on SMP boxes running Windows Server 2003.
157554	Maximum number of tables per database corrected to 32,640 in the “Limits” appendix of the Operations Guide.
163883	Enable Intel hyperthreading processors for the InterBase server using a parameter in <code>ibconfig</code> , the InterBase configuration file.
165415	Description of third argument to <code>substr</code> UDF has been corrected in the <i>Developer's Guide</i> .

## Bugs fixed in InterBase 7.0

---

### For InterBase 7.0:

Bug Number	Description
58806	The Blob is read correctly in the example stat7.e.
58842	Fixed situation where multi-user applications on NT machines could start processes so quickly that the server listener thread was overwhelmed and refused attachments.
58862	UDF function ltrim now returns correct data.
58883	Data selected from an external file is now returned correctly.
58932	IB numbers have been assigned to gbak error messages.
58963	IB server no longer fails on NT when a trigger is dropped or altered to be inactive.
58967	Crash of UDF function no longer causes IB server to crash.
58981	External file locks are now released when the connection is closed.
58987	isql script no longer crashes server.
60059	Specifying a default value in a column defined with a domain no longer causes the default value to propagate to subsequent columns defined with that domain.
60109	Repeatedly executing a stored procedure no longer crashes IB server.
60151	gfix -sweep no longer causes IB server to crash.
60232	Two of the examples, <i>api13.c</i> and <i>api16.c</i> , have been fixed.
60313	Queries no longer continue running after client disconnects.
60353	Certain TPB parameters that caused the IB server to crash have been fixed.
60494	UDFs release memory properly.
60547	Install process recognizes when an IB server is running and gives appropriate message.
69521	DATE math (addition and subtraction) returns correct results.
76342	Install API no longer overwrites the security database.
85863	Rollback performs correctly on an AFTER UPDATE trigger.
97500	Maximum index key size for single field keys has been corrected to 252 bytes with additional explanatory material added.
100443	DROP VIEW no longer drops tables, DROP TABLE no longer drops views.
105375	Change the data type of the IB password saved in the registry by IBReplicator from REG_SZ to REG_BINARY.
111916	Licensing chapter that was removed from the Operations Guide has been reinstated.
111973	isc_license_* functions were reinstated in the API Guide.
112969	The linux connect example had an incorrect colon after "localhost".
116273	Fixed typo in server activation license as displayed in IBConsole 334.
122284	ComponentStartApplicationError installing IB under Windows 98.

**For InterBase 7.0:**

<b>Bug Number</b>	<b>Description</b>
122385	ROWS keyword now takes its result set after aggregate functions are executed.
123280	Calling substr() UDF with 'ACHAR' IB server crash fixed.
123396	Uninstalling InterBase now removes Start   Programs   InterBase.
123397	Uninstalling IB removes the correct files.
123464	Uninstalling IB removes InterBase Server Manager.
123563	Redundant install confirmation dialog box has been removed.
123572	Windows color settings now used for tooltip colors.
124196	Wording and miscellaneous typos corrected in uninstall screen.
124288	Do not set timestamps on <i>setupapi.dll</i> and <i>msvcrt.dll</i> .
130454	PLAN syntax now works for quoted identifiers.
130826	IBConsole no longer requires user to log into server as SYSDBA.
140544	Internal software consistency check no longer thrown when shutting down IB server with active connections.
148244	UDF substr() no longer hangs IB server.
148520	Operations Guide no longer references to Novell Netware support.