




Release Notes

InterBase® 7.1
June 2003



VERSION 7.1

Borland®
InterBase®

Borland Software Corporation
100 Enterprise Way, Scotts Valley, CA 95066-3249
www.borland.com
June 11, 2003

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 © 2003 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 and updates	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 and updates

New in InterBase 7.1	2-1
New cross-platform installer.	2-1
New registration	2-1
Precision of exact numerics	2-2
The Borland Data Provider for ADO.NET	2-2
Installing the InterBase BDP	2-2
The BDP namespace.	2-2
Using the BDP with C#Builder.	2-2
Using the BDP with Microsoft Visual Studio .NET 2003	2-2
Deploying a BDP application	2-4
Older drivers	2-4
New ODS	2-4
Savepoints	2-4
Savepoints in SQL.	2-4
Savepoints in the InterBase API	2-4
Savepoints in triggers and stored procedures	2-5
A SAVEPOINT example	2-5
New keywords	2-5
Performance monitoring now accessible in IBConsole	2-5
New character sets	2-5

Enhanced SMP support	2-6
Hyperthreading support on Intel processors	2-6
Change in gbak functionality	2-6
Hyperthreading support for Intel processors	2-7
New SQL command: DROP GENERATOR.	2-7
Enhanced garbage collection and index handling.	2-7
IBConsole displays additional object dependencies.	2-7
Uninstalling InterBase	2-7
Using the InterBase Install API	2-7
Documentation fixes and changes.	2-8
UDF library documentation has been moved	2-8
Declaring Blob UDFs	2-8
Calling convention for UDFs.	2-8
Portable UDFs	2-8
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.	2-9
InterClient connection pooling	2-10
InterClient scrollability.	2-11
The Connection class	2-11
The ResultSet class	2-12
New InterClient methods	2-12
Methods for the Statement and PreparedStatement classes	2-12
The BatchUpdateException class	2-13
The DatabaseMetaData.supportsBatchUpdates function	2-14
Additional functions	2-14
Code examples.	2-14
InterClient and the Borland Enterprise Server.	2-15
Other InterClient enhancements	2-15
New in InterBase 7.0	2-15
New in InterBase 6.5	2-19

Chapter 3

Bugfix list

Bugs fixed in InterBase 7.1.	3-1
Bugs fixed in InterBase 7.0.	3-2
Bugs fixed in InterBase 6.5.	3-4
Bugs fixed in InterBase 6.0 and 6.0.1	3-5

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 and updates

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.

New in InterBase 6.5 A review of features introduced in InterBase 6.5

Chapter 3: Bugfix list

Bugs fixed in InterBase 7.1

Bugs fixed in InterBase 7.0

Bugs fixed in InterBase 6.5

Bugs fixed in InterBase 6.0 and 6.0.1

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

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

Migration issues

Migrating databases from InterBase 6, 6.5, or 7.0 to InterBase 7.1 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**.

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

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 and updates

This chapter describes what's new in InterBase 7.1 and InterClient 4.0, and reviews features introduced in InterBase 7.0 and 6.5.

New in InterBase 7.1

- New cross-platform installer
- New registration
- Precision of exact numerics
- The Borland Data Provider for ADO.NET
- Older drivers
- New ODS
- Savepoints
- New keywords
- Performance monitoring now accessible in IBConsole
- New character sets
- Enhanced SMP support
- Hyperthreading support on Intel processors
- Change in gbak functionality
- Hyperthreading support for Intel processors

New SQL command: DROP GENERATOR

- Enhanced garbage collection and index handling
- IBConsole displays additional object dependencies
- Uninstalling InterBase
- Using the InterBase Install API
- Documentation fixes and changes

New in InterClient 4.0

- DataSource properties for InterBase
- InterClient connection pooling
- InterClient scrollability
- New InterClient methods
- InterClient and the Borland Enterprise Server
- Other InterClient enhancements

New in InterBase 7.0

New in InterBase 6.5

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.

The Borland Data Provider for ADO.NET

For Windows platforms, InterBase 7.1 includes a Borland Data Provider (BDP) for ADO.NET programming. The BDP that ships with InterBase connects only to InterBase databases version 7.0 or later. It can be used with Borland's C#Builder, with Microsoft's Visual Studio .NET 2003, and as a standalone library.

For more information about working with Microsoft's .NET Framework, see the following links:

<http://msdn.microsoft.com/netframework/>

http://msdn.microsoft.com/library/default.asp?url=/library/en-us/netstart/html/cpframeworkref_start.asp

<http://msdn.microsoft.com/library/default.asp?url=/library/en-us/cpguide/html/cpconprogrammingwithnetframework.asp>

Installing the InterBase BDP

The following steps install the Borland Data Provider for InterBase, which connects only to InterBase databases. To use the BDP to connect to other databases, use Borland's C#Builder.

- 1 Run the Windows installer, *install_windows.exe*.
- 2 Click the Connectivity Drivers button to display the Select a Connectivity Driver window.
- 3 Click the ADO.NET Data Provider button; click Next.
- 4 Specify a location for the BDP install and click Install.

The BDP namespace

The Borland Data Provider packages are all under the *Borland.Data* namespace.

Using the BDP with C#Builder

The documentation that ships with C#Builder includes instructions on using the BDP with C#Builder and provides several handy tutorials.

Using the BDP with Microsoft Visual Studio .NET 2003

This section provides an outline of how to register the BDP with VS 2003, start a project, create an ADO.NET connection, and get data from the connection.

Registering BDP with Microsoft Visual Studio .NET 2003

- 1 Copy *BdpConnections.xml* and *BdpDataSources.xml* from the directory where you installed BDP to *<VisualStudio_home>\Common7\IDE*.
- 2 Start Microsoft Visual Studio .NET 2003.

3 Add a new tab to the Toolbox:

- a** If the Toolbox is not visible, select View | Toolbox to display it.
- b** Right-click in the Toolbox and choose Add Tab. This opens a field at the bottom of the Toolbox. In this field, provide a descriptive name for the new tab, such as “InterBase BDP” and press Enter.

4 Add the BDP Components to the new Toolbox tab:

- a** In the Toolbox, right-click on the new tab and select Add/Remove items to display the Customize Toolbox dialog box.
- b** In the .NET Framework Components tab, click the Browse button and browse to the folder where you installed the BDP.
- c** Select the Borland.Data.Provider.dll file and click Open.
- d** In the Customize Toolbar dialog, you should see four components highlighted. Click OK.

You should now see the new components listed in your new tab. If no project is open, they will be grayed out.

Creating an ADO.NET connection

- 1** In Microsoft Visual Studio .NET 2003, start a new project or open an existing project.
- 2** If necessary, display the Toolbox. To add a *BdpConnection* component to the Form Designer, find the *BdpConnection* component under your newly added BDP tab in the Toolbox. Double-click (or drag) it, to add a *bdpConnection1* component to your form.
- 3** Display the Connection Editor for this component either by right-clicking and choosing Connection Editor from the context menu or by selecting the component and clicking Connection Editor in the Properties Editor at the right of the screen.
- 4** In the Connection Editor, click the Add button to display the Add New Connection dialog. Select InterBase for the Provider Name and type a descriptive name—for example, “InterBase”—for the Connection Name. Click OK.
- 5** Under Connections, select the connection that you just created. Under Connection Settings, select Database and browse to the database you want to connect to. If you are exploring and need an example database, try *employee.gdb* under `<interbase_home>\examples\database`.
- 6** Click the Test button; should see a dialog box displaying a “Connection Successful” message.

Getting data from a connection

- 1** Open a project in which you have completed the steps described in [Creating an ADO.NET connection](#).
- 2** Find the *BdpDataAdapter* object on the Toolbox BDP tab and drop it on the Form Designer.
- 3** Select the new *bdpDataAdapter1* object and choose Configure Data Adapter from the context menu or from the Properties Editor to display the Data Adapter Configuration dialog box. This dialog box lists all the tables in the selected database in the Tables window. When you select a table, its columns are listed in the Columns window.
- 4** Select a table from the Tables window, make any desired settings, and click the Generate SQL button.
- 5** To test the generated SQL, select the Preview Data tab and click the Refresh button. This displays a grid view of the columns and data in your selected database. Click the Command tab and click OK. You have now added a new object, *dataSet1*, to your project.
- 6** To set the data live for visual design, expand the Live Data option on the Property Grid and set the Active property to True.
- 7** Drop a *DataGrid* onto the Form Designer and adjust its size and location as desired.
- 8** Set the *DataSource* property to *dataSet1* and the *DataMember* property to *Table1*.

9 To launch the application click Run or Build | Debug | Start.

Deploying a BDP application

The following files must be deployed with any application that uses the Borland Data Providers:

File name	Data Provider	Location
<i>Borland.Data.Common.dll</i>	All	GAC
<i>Borland.Data.Provider.dll</i>	All	GAC
<i>Borland.Data.Interbase.dll</i>	InterBase	GAC
<i>bdpint.dll</i>	InterBase	search path

Older 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
    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 and 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.

Uninstalling InterBase

To uninstall InterBase, you must have the Java VM version 1.3.1 or higher installed on your machine. You can download this from <http://www.java.sun.com/downloads>.

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

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

Name	Type	Description	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
<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*PageSize 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>
<property>
  <prop-name>dialect</prop-name>
  <prop-type>Enumerated</prop-type>
  <prop-value>interbase</prop-value>
</property>
</visitransact-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 boolean 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 sql 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 <i>PreparedStatement</i> 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
<code>public BatchUpdateException(String reason, String SQLState, int vendorCode, int [] updateCounts)</code>	Constructs a <code>BatchUpdateException</code> object where: <ul style="list-style-type: none"> • <code>reason</code> is a string describing the exception, • <code>SQLState</code> is an object containing Open Group code identification, • <code>vendorCode</code> identifies the vendor-specific database error code • <code>updateCounts</code> 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.
<code>public BatchUpdateException(String reason, String SQLState, int [] updateCounts)</code>	Constructs a <code>BatchUpdateException</code> object where: <ul style="list-style-type: none"> • <code>reason</code> is a string describing the exception, • <code>SQLState</code> is an object containing the InterBase error code • <code>updateCounts</code> 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. • The vendor code is implicitly set to zero.
<code>public BatchUpdateException(String reason, int [] updateCounts)</code>	Constructs a <code>BatchUpdateException</code> object where: <ul style="list-style-type: none"> • <code>reason</code> is a string describing the exception, • <code>updateCounts</code> 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. • The following values are implicitly set: the <code>vendorCode</code> is set to zero and the Open Group code identification is set to null.
<code>public BatchUpdateException(int [] updateCounts)</code>	<ul style="list-style-type: none"> • Constructs a <code>BatchUpdateException</code> object where <code>updateCounts</code> 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. • The following values are implicitly set: <code>reason</code> is set to null, <code>vendorCode</code> is set to zero, and the Open Group code identification is set to null.
<code>public BatchUpdateException()</code>	The following values are implicitly set: <ul style="list-style-type: none"> • <code>updateCounts</code> is set to a zero-length integer array, • <code>reason</code> is set to null, • <code>vendorCode</code> is set to zero, • the Open Group code identification is set to null.
<code>public int [] getUpdateCounts()</code>	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.

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

- 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:

```
isc_array_gen_sdl2()          isc_array_get_slice2()
isc_array_lookup_bounds2()   isc_array_lookup_desc2()
```

<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 6-31 to 6-42 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
TMP\$ATTACHMENTS	One row for each connection to a database
TMP\$DATABASE	One row for each database you are attached to
TMP\$POOL_BLOCKS	One row for each block of memory in each pool
TMP\$POOLS	One row for each current memory pool
TMP\$PROCEDURES	One row for each procedure executed since the current connection began
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.

New in InterBase 6.5

- Metadata security

This feature protects metadata from modification by unauthorized users. It adds the SQL GRANT/REVOKE security framework to InterBase's system tables. The default access privileges for PUBLIC have been changed to allow only SELECT access; this prevents ordinary users from corrupting the database by modifying the system metadata. The database owner, SYSDBA and operating system administrator (root on UNIX and Administrator on Windows) continue to have all privileges to the metadata as well as the privilege to grant access to the metadata to anyone else.

The backup/restore **gbak** utility has been modified to restore security privileges on system metadata.

NOTE: **gbak** has always backed up privileges to all tables, it just never restored them to the system tables. **gbak** is also being modified to record the ODS major and minor version attributes of the database that was backed up in the backup file. Knowing that information helps the engine know what system tables need to have default privileges added when the database is restored.

- Scripts for changing database security

Three SQL scripts are included in `<interbase_home>/examples/security: readmeta.sql, writemeta.sql` and `blindmeta.sql`. These scripts can be run against databases with ISQL to make wholesale changes to a database's metadata PUBLIC privileges.

Readmeta.sql applies the default PUBLIC access privileges of the IBv6.5 engine. It can be used to return a database with customized metadata privileges back to the default.

Writemeta.sql grants all metadata privileges to PUBLIC. This is the metadata access profile that existed in InterBase 6.0 and earlier.

Blindmeta.sql revokes all metadata privileges from PUBLIC. This prevents any PUBLIC user from querying the system tables.

For more information, see the "Database Security" chapter in the *Operations Guide*.

- 64 bit I/O

InterBase 6.5 and later supports the creation of database files greater than 4 GB. In previous versions of InterBase, databases greater than 4 GB had to be multi-file databases.

- Asynchronous cancel

An new option, *DSQL_cancel* has been added to DSQL API *isc_dsql_free_statement()*. The *DSQL_cancel* option allows for the asynchronous cancellation of an executing statement. The client that was executing the statement receives a status code of *isc_cancelled*. Once a statement has been cancelled, any subsequent execution restarts the statement, rather than resuming it.

For more information, see *isc_dsql_free_statement()* in the API Function Reference.

- New server configuration parameters

CPU_AFFINITY

This parameter specifies the processors to be used by the InterBase server on a SMP machine. This parameter is only available on Windows. The default value is 1. The InterBase server uses the first CPU on the system by default. A process affinity mask is a bit vector in which each bit represents the processor on which the threads of the process are allowed to run.

For processor 1: 1 (1st bit in bit vector)
 For processor 2: 2 (2nd bit in bit vector)
 For processors 1 & 2: 3 (11, 1st and 2nd bits in bit vector)
 For processor 3: 4 (3rd bit in bit vector)
 For processors 1,2 and 3: 7 (111 in bit vector)
 For processors 2 and 3: 6 (110 in bit vector)

And so on.

SWEEP_QUANTUM

This parameter specifies the maximum number of records that a garbage collector thread or a sweeper thread is allowed to work before yielding control back to the worker threads. The default value is 10.

USER_QUANTUM

This parameter specifies the maximum number of records that a worker thread (thread running an user query) is allowed to work before yielding control back to other threads. The default value is 100.

SWEEP_YIELD_TIME

This parameter specifies the number of milliseconds the sweeper or the garbage collector thread sleeps. The default value is 1 millisecond. Note that this does not affect worker threads.

For more information on these and other server configuration options, see the “Server Configuration” chapter of the *Operations Guide*.

- New keywords

InterBase 6.5 introduces the following new keywords:

PERCENT ROWS TIES

- The ROWS clause

InterBase now supports a ROWS clause for SQL SELECT, UPDATE and DELETE statements. It has a general syntax of ROWS <lower_value> [TO <upper_value>] [BY <step_value>] [PERCENT] [WITH TIES]. The ROWS clause can be used in isolation or (most often) in conjunction with the SQL ORDER BY clause. The WITH TIES clause must be used in conjunction with ORDER BY.

The ROWS clause introduces three new InterBase SQL keywords: ROWS, TIES, and PERCENT.

For more information, see the “Working with Data” chapter in the *Embedded SQL Guide*.

- Support for ROWS clause in gpre

GPRES now supports this additional functionality of the ROWS clause. Currently this feature is only supported with the C / C++ programming language.

- XML Generation

InterBase provides three API calls for generating XML documents directly from InterBase tables: *isc_dsqli_xml_fetch()*, *isc_dsqli_xml_fetch_all()*, and *isc_dsqli_xml_buffer_fetch()*. These functions are a part of a new client side library called *ibxml.dll* on Windows and *ibxml.so* on Solaris and Linux.

The structures defined for these functions are located in a new header file called *ibxml.h*. The prototype definitions are included in the file *ibxml_proto.h*. This header file also internally includes *ibxml.h*.

For more information, see the “Exporting XML” chapter in the *API Guide*. For code examples see *xml_api_buffer.c* and *xml_api_file.c* in the `<interbase_home>/examples/API` directory.

- Deprecated JRE API calls removed from InterClient 2.5

Some deprecated calls have been removed from the JDBC driver, InterClient 2.5 to be in compliance with JRE 1.3. This requires the presence of Java 2 on the platform.

- New JDBC methods

Added quoted identifier support for the following methods of the *DatabaseMetaData* class:
supportsMixedCaseQuotedIdentifiers(), *storesMixedCaseQuotedIdentifiers()*, and *getIdentifierQuoteString()*

The *getBigDecimal()* method of the *ResultSet* class no longer takes a scale argument. The new syntax is *getBigDecimal(int)* and *getBigDecimal(String)*.

Bugfix list

This chapter lists bugs that have been fixed for the last few InterBase releases: InterBase 7.1, 7.0, 6.5, 6.0.1, and 6.0.

Bugs fixed in InterBase 7.1

For InterBase 7.1:

Bug Number	Description
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.
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.
125652	Declaration for external function ascii_char has been fixed in code and docs.
125988	Queries with ORDER BY and GROUP BY now work as expected when a table is joined with the result set from a stored procedure.
127852	The Easysoft ODBC driver no finds the <i>ib.hlp</i> file.
135054	current_date, current_time, and current_timestamp 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.

For InterBase 7.1:

Bug Number	Description
141235	Using a trigger to delete a row prior to an update no longer corrupts the database.
146833	gbak 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.
149819	SELECT from TMP\$RELATIONS no longer gives consistency check.
151283	Fixed problem with subselects in stored procedures.
151517	CREATE VIEW now allows large metadata names.
152420	Databases can now grow larger than 4GB on Linux.
152515	Fixed problem with using COMPUTED BY with procedural views.
152584	The "Working with Generators" chapter of the Data Definition Guide now describes using GEN_ID() with SELECT.
152760	Outdated references to database file size limits have been removed from the Operations Guide.
152891	Description of the tanh function in the InterBase UDF library has been corrected.
153053	[duplicate bugs] Large merge sorts no longer cause server crash.
153075	RDB\$FIELD_LENGTH now reports correct value for VARCHAR fields after ALTER TABLE.
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 NULL in a date column could sometimes crash the server; fixed.
156443	Fixed problem with CREATE TABLE 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 <i>ibconfig</i> , the InterBase configuration file.
165415	Description of third argument to <i>substr</i> 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.

For InterBase 7.0:

Bug Number	Description
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.
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.

For InterBase 7.0:

Bug Number	Description
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.

Bugs fixed in InterBase 6.5**For InterBase 6.5:**

Bug Number	Description
58979	Creating and dropping foreign keys no longer requires exclusive access.
59034	No more continuous sweeps when sweep interval hit and OIT stuck.
59043	InterClient: DDL Statements no longer invalidate live preparedStatement handles.
60099	Fixed minor errors in UDF exception messages.
60237	#include <stdio.h> is in all 'c' code examples.
60470	File length is correctly calculated when creating a multifile database.
60507	Database cache no longer hinders performance.
67648	InterClient: Bug Code 10010 no longer occurs when reusing one statement after another statement has issued DDL.
69510	CAST (<val> AS DATE) no longer causes errors in SQL dialect 1.
71192	Fixed typo in error messages.
73555	Trigger no longer fires twice for each insertion into the subject table when trigger_source is set to NULL on an after-insert trigger.
102222	Guardian no longer launches on boot even though disable automatic launch on boot is set in Win2k control panel.
105406	Fixed views with quoted identifiers in dialect 3.
106760	Default location of InterBase Replication Directory in Replication Server setup fixed.
107296	ISQL PLAN output buffer length increased

For InterBase 6.5:

Bug Number	Description
108567	Results are correct when computing rounding values with dialect 1.
109320	varchar is no longer truncated from server to client.
109609	Multi-threaded application problems with GDS client library on Linux have been fixed.
109914	Deleting sources from system tables with triggers no longer causes some triggers to execute twice.
110816	Fixed segmentation fault for gds_inet_server.
114677	TIP pages sequence is now a LONG.
116274	Description for InterBase Server and InterBase Guardian registry keys is now written.
117152	PROCEDURE handling numerics(x,x) has been fixed.
117890	Multithreaded applications no longer give SigSegv on Linux.
118206	Sort files in temp directory when running ESQL.
119520	Fixed information for gstat -r and -t options in documentation.
120005	Multithreaded application using InterBase on Windows Local Access no longer hangs.
120548	After using gbak to backup and restore, all authorized users can access the database.
122004	Alter Domain command has been fixed.
122135	<i>libncurses.so.4</i> now found in Red Hat 7.1, 7.2 for InterBase install.
122495	Allow setting of CPU affinity on Windows for ibserver.
122723	isql extract operation delimits all table and field names with for SQL_DIALECT 3.
122907	<i>gdsint12.dll</i> module now loads correctly.
123281	Server no longer crashes when you try to make a unique index.
123635	No database corruption on Unix databases when allowing the server to attach to it as separate database instances.

Bugs fixed in InterBase 6.0 and 6.0.1**For InterBase 6.0 and 6.0.1:**

Bug Number	Description
58983	GRANT statement to stored procedures and triggers now work correctly.
58985	NOT NULL constraints are evaluated prior to triggers.
58988	Altering a procedure that is being run by another procedure no longer fails.
59012	Changes are undone when exception is raised before an insert trigger.
59014	Incorrect use of index on duplicate values no longer causes crash.
59022	No SQL error when specifying and using alternate character set.

For InterBase 6.0 and 6.0.1:

Bug Number	Description
59035	gbak permission error for db owner when other user creates a table has been fixed.
59050	Error when valid user revokes select privilege has been fixed.
59080	Doing aggregate on aggregate view no longer crashes the server.
60124	Conversion problem: Cast now produces correct result.
60166	gbak now correctly restores databases that were successfully backed up.
60188	Stored procedures no longer become undeleteable when they reference a deleted UDF.
60220	Location of <i>ib_udf.dll</i> is now documented.
60221	Data Definition Guide section on declaring external files has been rewritten.
60255	Set options in ISQL are now parsed before connecting to a database.
60278	Adding multiple licenses concurrently no longer causes crash.
60288	Attempting to remove a license using a badly formatted key/id no longer causes a crash.
60310	UNIX Install: /usr/interbase link is now correct if directory exists.
60314	UNIX Install: links are pointing to the correct directories.
60316	ibserver does now recognizes an invalid write error and responds correctly.
60320	UNIX Install: Install scripts now use portable ps command.
60325	Restore of <i>isc4.gdb</i> no longer generates an error.
60360	Altering a domain type does now rebuild any indexes which were defined.
60365	Icon no longer appears in task bar when IB Server is started only as a service.
60403	GPRES no longer crashes with examples/ <i>stat3.e</i> .
60425	GPRES give error for DROP INDEX 'ind1' (you must use double quotes).
60428	GPRES now handles delimited cursor names properly.
60430	ISQL shows correct datatype for UDFs.
60451	Problems with EXIT used with WHEN SQLCODE have been fixed.
60496	"select -999.99..." in GPRES gives correct results.
61481	Creating db with stored procedure results in Parameters mismatch for procedure
61504	Install not recognizing non-administrative user
62001	GFIX with no parameters does not generate an error
62174	Triggers can created on non-existing fields
64315	Command line help for ibmgr lists passwd instead of password
64318	If lock table is out of room the server crashes
66143	Stored Procedure produces different results the second time it is called
66339	Cannot transliterate characters between character sets error when concatenating

For InterBase 6.0 and 6.0.1:

Bug Number	Description
67046	Dates with year greater than 10000 and less than 1 can be inserted successfully
68063	Adding id and key in wrong order when adding a license causes server to crash
68159	A UDF can point to any entry point to any system call
69057	Incorrect results for timestamp calculations
70109	All work done by triggers is not undone when an exception is raised
70443	-pa is incorrectly documented as command line option for gbak
70628	The guardian continually tries to start the server if a license cannot be found
70788	Procedure mismatch during alter procedure causes procedure to become useless
71191	Killing a client during a read crashes the server
71495	Queries with LIKE against Korean data set crash the server
71953	Create procedure statement which references a non-existent generator causes hang
72112	It is possible to enter 0000 as a valid year
72458	Dynamically loading and unloading of <i>gds32.dll</i> causes resource leaks
73766	SELECT...WHERE...BETWEEN crashes server
74091	Subqueries are not supported with the between predicate
75310	When Guardian restarts the server a thread handle is not closed
75637	Error 997;Overlapped I/O operation is in progress when stopping service on WIN2K
75902	Granting permission to update a column doesn't work for roles
76353	GBAK truncates metadata names after the first space, causes damaged backup
76500	Selecting the rdb\$db_key with a group by will crash the server
87659	Stored Procedure returns incorrect result set
100331	SELECT from View with grouping causes server crash
101628	Creating too many generators can corrupt your database
102882	Select count(e.rdb\$db_key) from employee e causes connection lost to database
100627	Superserver on Windows NT does not service more than 256 users
100649	Internal UDF enables database corruption
100651	Access to InterBase databases insecure on some versions
57972	Inappropriate bug check when out of disk space
60130	Operations on corrupted databases crash IBServer.exe
60131	Restore on multiple tapes does not work

