

Release Notes

InterBase 6.0 Field Test
November 1999



Borland/INPRISE

100 Enterprise Way, Scotts Valley, CA 95066 <http://www.interbase.com>

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

Copyright 1999 Inprise/Borland. All rights reserved. All InterBase products are trademarks or registered trademarks of Inprise/Borland. All Borland products are trademarks or registered trademarks of Inprise/Borland. Other brand and product names are trademarks or registered trademarks of their respective holders.

Table of Contents

CHAPTER 1	General Information	
	General Information	6
	Contacting Inprise/Borland	6
	The InterBase install directory	6
	Installation	7
	Uninstalling InterBase 6	7
	InterBase documentation	7
	Documentation in PDF form	7
	Full-text searching	8
	Links	8
	Installing Acrobat Reader With Search	8
	System requirements	9
CHAPTER 2	New InterBase 6 Features	
	Feature overview	11
	Feature description	13
	IBConsole	13
	Dialects	14
	SQL delimited identifiers	18
	InterBase Express for Delphi users	19
	Large exact numerics	19
	SQL DATE, TIME, and TIMESTAMP	22
	Changing column and domain definitions	28
	Read-only databases	30
	New keywords	31
	Status vector and warning messages	31
	New gbak functionality	32
	Services API	33
	Install API	33
	Licensing API	33
	InterBase Client for UNIX	34

New On-Disk Structure (ODS)	34
New error messages	34
Running InterBase 6 as an application	34
Compiling and linking	35
Known bugs and workarounds	36

General Information

This chapter covers the following topics:

- Contacting Inprise/Borland
- The install directory
- System requirements
- Documentation and Adobe Acrobat Reader
- System requirements

IMPORTANT See *Getting Started* for installation instructions and detailed information about migrating existing databases and applications.

General Information

This section includes information on contacting Inprise/Borland, as well as notes about installing and uninstalling InterBase.

Contacting Inprise/Borland

▶ *Mailing address*

Inprise/Borland
 100 Enterprise Way
 Scotts Valley, CA 95066-3249
 Phone: 1-888-345-2015 or (831) 431-6500
 Fax: (831) 431-6510

▶ *World-wide web site*

Inprise/Borland maintains an Internet site on the world-wide web. The URL of this site is:

<http://www.borland.com/>

This site has product information, tools for download, technical knowledge base, and hosts a community of InterBase partners and VARs. As with any Internet site, it grows continuously, so visit it often.

To purchase InterBase or to ask questions prior to purchase, go to:

<http://www.interbase.com/cgi/contact.cgi>

▶ *Email addresses*

- For questions about product information, product release schedule, features, feature requests, and VAR partnerships, send email to:

prodinfo@interbase.com

This email address is also helpful if you are a non-U.S. customer and you need to learn the best source of technical support or sales assistance in your region.

- For issues pertaining to content and presentation on our web site, send email to:

webmaster@interbase.com

- 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/>

The InterBase install directory

Throughout this document, “interbase_home” refers to the InterBase install directory. By default, this is `C:\Program Files\Borland\InterBase` on Windows NT and Windows 95/98, and `/usr/interbase` on UNIX platforms. You can use the INTERBASE environment variable to change this location. See the *Operations Guide*, Chapter 3, “Server Configuration” for more information on environment variables.

Installation

Complete installation instructions are in the installation chapter of *Getting Started*.

Installation requires approximately 35MB of disk space for a full install that includes InterBase Server and Client, Adobe Acrobat Reader, and the full document set. Only 17MB is needed to install the InterBase server and client without the documents, examples, or Acrobat Reader.

Uninstalling InterBase 6

The Windows InterBase installation allows you to choose between performing a complete install or selecting individual components. If you install a subset of the components, you can run the install again later to add more components. However, when you uninstall, only the components that you installed the last time are removed.

InterBase documentation

InterBase supplies an extensive document set. The entire set is available online in Adobe Acrobat PDF format.

If you buy InterBase as a standalone product, you receive the printed version of the document set. If you acquire InterBase as part of another Inprise product distribution such as Delphi, you will receive only a subset of the document set in printed form. You can order the entire document set in printed form for a nominal cost from Inprise sales.

Documentation in PDF form

InterBase provides the books in the document set plus the Release Notes in PDF format. You have the option to install the complete document set in PDF form on your hard drive when you are installing the InterBase 6 product. This requires about 24MB of disk space. You also have the option of installing shortcuts to the documentation, which requires very little disk space, but which requires that you insert the CD-ROM every time you want to refer to the documentation.

The InterBase document set consists of the *API Guide*, the *Data Definition Guide*, the *Developer's Guide*, the *Embedded SQL Guide* (formerly the *Programmer's Guide*), *Getting Started*, the *Language Reference*, the *Operations Guide*, and these *Release Notes*.

To access all of the books from a single document, open [2AllBooks.pdf](#). It contains links to the rest of the document set.

IMPORTANT The beta document set is incomplete. New material is marked with change bars. Some of the new information is incomplete or missing, and some outdated information may remain.

You need Adobe Acrobat Reader With Search to view and search these documents. See "Installing Acrobat Reader With Search" on page 8 of this document for information on how to acquire and install Acrobat Reader.

Full-text searching

The five-book document set has been indexed for full-text searching. If you are viewing the documents using Acrobat Reader With Search, you can enter a query and receive a list of hits from all five books in the InterBase document set.

To use full-text searching, click the  button and search for a word or phrase. Acrobat Reader returns a list of books that contain the phrase. Choose the book you want to start looking in to display the first instance. You then use the  and  buttons to step forward and back through instances of your search target. Reader moves from one book to the next. To go to a different book at will, click the  button to display the “found” list.

Note that full-text searching is not the same as Find () , which searches only the current document.

Tip On UNIX and Linux, always open documents using an absolute pathname to the PDF file, to make Acrobat Reader With Search associate the index with the PDF document correctly.

Links

The PDF documentation set contains many hypertext links that take you to referenced points in the document with a single click. In addition, the Table of Contents and Index entries are hypertext links and therefore are clickable. Throughout the document set, clickable links appear **bold and green**.

Installing Acrobat Reader With Search

On Windows 95/98 and NT

You can install Acrobat Reader 3.01 With Search by choosing Install Adobe Acrobat Reader 3.0 from the InterBase Launcher. You can also install it directly by running **setup.exe** from the /Adobe directory of the InterBase CD-ROM.

On Solaris

The InterBase CD-ROM includes Acrobat Reader With Search for both the UNIX platform and for Windows. The files are in subdirectories of the /Adobe directory on the InterBase CD-ROM. Read instguid.txt in the /Adobe/*UNIX platform* directory for UNIX installation instructions. To install on a Windows 95/98 or Windows NT platform, run setup.exe in the /Adobe/Windows directory.

From the Adobe website

If you don't have the InterBase CD-ROM handy, you can get Acrobat Reader for free from the Adobe website. It's at <http://www.adobe.com/prodindex/acrobat/readstep.html>. Check the box next to “Include option for searching PDF files...” to download Acrobat Reader With Search, not the plain Acrobat Reader.

System requirements

Microsoft Windows NT and Windows 95/98

Operating system: Windows NT 4.0 with Service Pack 5 or Windows 95/98

Memory: 16 megabytes minimum; 64 or more recommended

Processor/Hardware model: 486 minimum; Pentium II or higher recommended

C/C++ compiler: Microsoft Visual C++ 4.2 or Borland C++ Builder 3

Sun Solaris

Operating system: Solaris 2.6.x or 7

Memory: 32 megabytes minimum; 64 or more recommended

Processor/Hardware model: SPARC or UltraSPARC



New InterBase 6 Features

This chapter lists features introduced in InterBase 6 and provides an overview and summary of each.

Bold green type indicates clickable links.

Feature overview

- **IBConsole**

InterBase now provides a single, intuitive graphical user interface. From within this environment, you can perform all the tasks necessary to configure and maintain an InterBase server, to create and maintain databases on that server, and to execute interactive SQL. IBConsole replaces the earlier Server Manager and InterBase Windows ISQL GUI environments. See **“IBConsole” on page 13**.

- **Dialects**

You now specify a dialect for both clients and databases to specify whether double quotes, large exact numerics, and the DATE datatype are interpreted as InterBase 5 or new InterBase 6 features. See **“Dialects” on page 14**.

- **Delimited identifiers**

In dialect 3, object identifiers can now include keywords, spaces, case-sensitivity, and non-ASCII characters. Such identifiers must be delimited with double quotes. See **“SQL delimited identifiers” on page 18**.

- **InterBase Express™ (IBX) for Delphi 5**

IBX is a set of Delphi 5 components that access the InterBase API directly, allowing greater speed and flexibility in writing applications for InterBase databases. IBX for InterBase 5 ships with the Delphi 5 product and Field Tests. These components address only InterBase 5 and earlier features and are found on the InterBase tab in Delphi 5. InterBase 6 includes a version of IBX that addresses InterBase 6 features, including the new Install API and Licensing API. This IBX version is a superset of the version included with Delphi 5. See **“InterBase Express for Delphi users” on page 19**.

- Large exact numerics

DECIMAL and INTEGER datatypes with a precision greater than 9 are now stored as 64-bit integers with a precision of up to 18 in columns created in InterBase dialect 3. See **“Large exact numerics” on page 19**.
- SQL DATE, TIME, and TIMESTAMP datatypes

InterBase 6 dialect 3 provides three datatypes for handling date and time: TIMESTAMP, DATE, and TIME. In dialect 3, the DATE datatype contains only date information. The TIMESTAMP datatype contains both date and time information in both dialects and is the equivalent of DATE in InterBase version 5 and earlier.

InterBase 6 also provides the new EXTRACT() function for extracting date and time information, as well as well as CURRENT_DATE, CURRENT_TIME, and CURRENT_TIMESTAMP functional operators, and three new date/time literals. See **“SQL DATE, TIME, and TIMESTAMP” on page 22**.
- ALTER COLUMN and ALTER DOMAIN

You can now use the new ALTER COLUMN clause of ALTER TABLE to change the name, position, and datatype of columns in tables. Additions to the ALTER DOMAIN syntax allow you to change the name or datatype of a domain. See **“Changing column and domain definitions” on page 28**.
- Read-only databases

Databases can now be set to read-only mode for security purposes or for distribution on read-only media. See **“Read-only databases” on page 30**.
- New SQL keywords

There are 17 new keywords associated with the new dialects, delimited identifiers, and datatype features. See **“New keywords” on page 31**.
- Updated status vector and SQL warnings

The status vector now provides the error code, type of message, and where the message was generated, and allows the use of SQL warnings and informational messages. See **“Status vector and warning messages” on page 31**.
- New **gbak** functionality

The **gbak** utility now incorporates the old **gsplit** functionality, allowing you to back up to and restore from multiple files. In addition, it implements the **-service** switch, which allows you to perform server-side backups using InterBase’s new Service Manager for improved performance. This functionality is also available from within IBConsole. See **“New gbak functionality” on page 32**
- New **gfix** functionality

The **gfix** utility now allows you to set the database dialect. This functionality is also available from within IBConsole. See **“Setting dialects using gfix” on page 17**.
- New **isql** functionality

You can now specify the dialect for **isql** at the command line when you invoke it. In addition, you can use the SET SQL DIALECT *n* statement to change the dialect of the **isql** client during a session. If you do not state a dialect explicitly, **isql** acquires the dialect of the database to which it is attached.

You can run **isql** sessions from within IBConsole.
- Services API

The Services API library provides functions that allow you to programmatically monitor and control InterBase servers and databases. See **“Services API” on page 33**.

- Install API

The Install API library provides functions that allow you to integrate the installation of your own product with the deployment of an embedded copy of InterBase. The InterBase portion of the install can be *silent*: it does not display billboards and need not require intervention from the end user. See **“Install API” on page 33**.

- Licensing API

The Licensing API library provides five functions that allow you to check, add, remove, and view certificate ID and key pairs (authorization codes). The fifth function retrieves and displays messages associated with the return values from the other four functions. See **“Licensing API” on page 33**.

- InterBase Client for UNIX

You can perform a client-only install on HP-UX and Solaris. See **“InterBase Client for UNIX” on page 34**.

- Replication

IBReplicator is a component of the InterBase 6 product that facilitates replication and synchronization between multiple InterBase databases that have similar structure. IBReplicator synchronizes databases by copying changes of the entire database or a subset of the database, as you specify. Replicant databases can reside on different servers or on the same server. This product component includes a replication server executable for each respective InterBase server platform, and also a Windows graphical tool for configuring and invoking data replication. You can find documentation for configuring and using IBReplicator in the “Data Replication” chapter of the *InterBase 6 Operations Guide*.

Feature description

This section provides a more detailed description of each of the new InterBase 6 features. These features are also discussed at greater length in the InterBase document set.

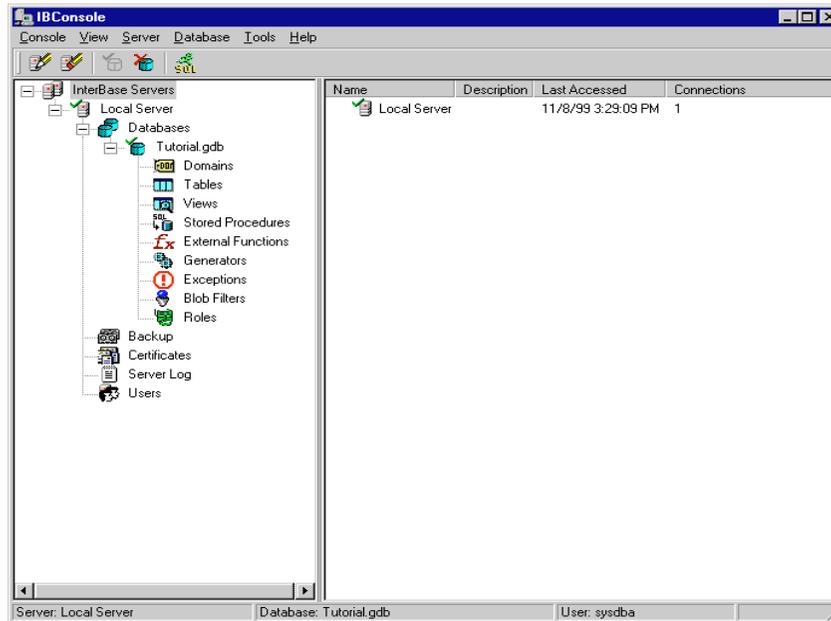
IBConsole

InterBase provides an integrated graphical user interface called IBConsole. Using IBConsole, you can configure and maintain an InterBase server, create and administer databases on the server, and execute interactive SQL, manage users, and administer security.

IBConsole is a GUI front end for InterBase’s command-line tools. It replaces the Server Manager and InterBase Windows ISQL interfaces found in earlier versions of InterBase.

IBConsole runs on Windows, but can manage databases on any InterBase server on the local network, and on UNIX, Linux, and NetWare.

FIGURE 2.1 IBConsole



You can use IBConsole to:

- Perform data entry and manipulation
- Configure and maintain a server
- Enter and execute interactive SQL
- Manage server security
- Backup and restore a database
- View database and server statistics
- Validate the integrity of a database
- Sweep a database
- Recover “in limbo” transactions

For details on using IBConsole, see the *Operations Guide*.

Dialects

InterBase 6 introduces the concept of dialects to allow users to move ahead with new features that are incompatible with older versions of InterBase: delimited identifiers, large exact numerics, and the SQL DATE, TIME, and TIMESTAMP datatypes.

► Database and client dialects

As InterBase moves forward in complying with SQL standards, some new features and usages are incompatible with older usage. Dialects assist in this transition. Dialect 1 guarantees compatibility with older databases and clients. Dialect 3 allows full access to new features. There is also a dialect 2 available for clients. It is a diagnostic mode only.

Transition features Features that behave differently in dialect 1 and dialect 3 are called *transition features*. The transition features are:

- Anything delimited by double quotes
- Date/time datatypes
- Large exact numerics (DECIMAL and NUMERIC datatypes with precision greater than 9)

Both clients and databases must be assigned a dialect. Databases are created in dialect 3 by default. The **isql** client takes on the dialect of the database to which it is attached unless you specify a different dialect.

IMPORTANT To display the dialect of both client and attached database, use ISQL from either IBConsole or the command line to issue the new SHOW SQL DIALECT statement.

The following section describes the differences between dialects.

FEATURES THAT ARE THE SAME IN ALL DIALECTS

The following InterBase 6 features are available in both dialect 1 and dialect 3:

- IBConsole
- The ALTER COLUMN clause of the ALTER TABLE statement
- The TIMESTAMP datatype, which is the equivalent of the DATE datatype in earlier versions of InterBase
- The EXTRACT() function and CURRENT_TIMESTAMP
- Read-only databases
- SQL warnings
- The Services API, Install API, and Licensing API
- InterBase Express (IBX) components in Delphi 5

DIALECT 1

In version 6 dialect 1, the transition features behave in the InterBase 5 manner:

- String constants can be delimited by either single or double quotes. Dialect 1 does not recognize delimited identifiers.
- The DATE datatype is not available, but is replaced by the TIMESTAMP datatype, which contains both date and time information. When an InterBase version 5 database is restored as version 6, datatypes that were formerly DATE are restored as TIMESTAMP.
- DECIMAL and NUMERIC datatypes with precision larger than 9 are stored as floating point numbers.

DIALECT 2

Clients can be assigned dialect 2. In this mode, they issue errors whenever they encounter double quotes, DATE datatypes, or NUMERIC/DECIMAL datatypes with precision greater than 9. This behavior is intended to alert the developer to potential problem areas during migration and is not useful for production purposes. To detect problem areas in the metadata of a database that you are migrating, extract the metadata and run it through a version 2 client, which will report all instances of transition features. For example:

```
isql -i v5metadata.sql
```

Do not assign dialect 2 to databases.

DIALECT 3

The following features are unique to dialect 3 and are incompatible with dialect 1 and all older InterBase databases and clients:

- String constants must be delimited by single quotes. Double quotes are used only for delimited identifiers. (See **“SQL delimited identifiers” on page 18.**)
- The DATE datatype holds only date information. Two new datatypes are available: a TIME datatype that holds only time information, and a TIMESTAMP datatype that holds the whole timestamp. TIMESTAMP replaces the functionality of the DATE datatype in earlier versions of InterBase. In addition, dialect 3 provides the CURRENT_DATE, CURRENT_TIME, and CURRENT_TIMESTAMP functional operators.
- DECIMAL and NUMERIC datatypes with precision larger than 9 are stored as 64-bit exact numerics if they are created in dialect 3. Note that columns that have such datatypes are still stored as floating point if they are migrated from an earlier version. See the migration chapter in *Getting Started* for more information on migrating columns to INT64 exact numerics.

► *Setting dialects from ISQL*

To use **isql** to create a database in a particular dialect, first set **isql** to the desired dialect and then create the database. There are two ways to set the dialect of **isql**:

- At the command line, you can invoke **isql** with the **-sql_dialect *n***, where *n* is 1, 2, or 3.
- Within a s SQL session, use SET SQL DIALECT *n*, where *n* is 1, 2, or 3. This overrides the dialect set in the command line when **isql** is invoked.

When you create a database with a client such as command-line **isql**, the database initially has the dialect of the client. For example, you can create a dialect 3 database in either of the following ways:

Specifying the dialect on the command line:

```
C:\> isql -dialect 3
CREATE DATABASE C:\databases\work.gdb;
```

Specifying the dialect during a session:

```
C:\> isql
SET SQL DIALECT 3;
CREATE DATABASE C:\databases\work.gdb;
```

If you create a database with **isql** and have not specified an **isql** dialect, **isql** creates a dialect 3 database.

Note If you do not specify a dialect in one of these two ways, **isql** acquires the dialect of the database to which it is attached.

► *Setting dialects using gpre*

- The command line option **-sql_dialect *n***, where *n* is 1, 2, or 3, sets the dialect of a database or client.
- EXEC SQL SET SQL DIALECT *n*, where *n* is 1, 2, or 3, sets the dialect of a database or client. SET SQL DIALECT overrides any value set at the command line.

► *Setting dialects using gfix*

The command line option `-sql_dialect n`, where *n* is 1 or 3, sets the dialect of an ODS 10 version database. The following statement sets `mydb.gdb` to dialect 3:

```
gfix -sql_dialect 3 mydb.gdb
```

► *Setting dialects with the API*

`isc_dpb_set_db_SQL_dialect`

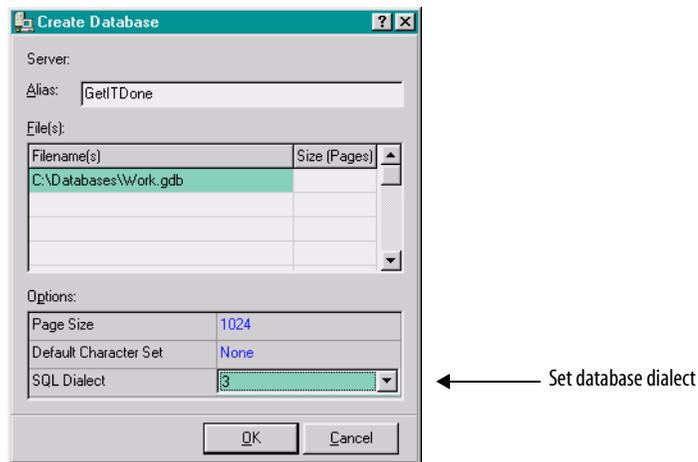
Use with the `isc_attach_database()` API function to overwrite the database's dialect.

■ `isc_info_db_SQL_dialect`

Use with the `isc_database_info()` API function to query the database's dialect.

SETTING THE DATABASE DIALECT IN IBCONSOLE

- In IBConsole, you set the dialect for a database you are creating in the Create Database dialog:



CHANGING THE DATABASE DIALECT IN IBCONSOLE

In IBConsole, select the database in the Tree pane and choose Properties to display the Database Properties dialog. Click the General tab and change the SQL dialect in the Options field.

Tip To suppress the display of system tables in IBConsole, deselect System Data from the View menu.

► *New #define constructs in ibase.h:*

```
#define SQL_DIALECT_V5          1
#define SQL_DIALECT_V6          3
#define SQL_DIALECT_CURRENT     SQL_DIALECT_V6
#define SQL_DIALECT_V6_TRANSITION
#define isc_dpb_SQL_dialect      63
#define isc_info_db_SQL_dialect  62
#define isc_dpb_set_db_SQL_dialect 65
#define isc_spb_pop_set_sql_dialect 14
```

► *Migration*

Migration issues are discussed in detail in the Migration chapter of *Getting Started*.

► *Error messages*

The following new error messages are associated with the use of dialects:

Error Code	Number	SQL Code	Message Text
isc_ddl_not_allowed_by_db_sql_dial	335544793	-817	Metadata update statement is not supported by the current database SQL dialect %d
isc_sql_dialect_datatype_unsupport	335544796	-804	Client SQL dialect %d does not support reference to %s datatype

TABLE 2.1 Dialect error message

SQL delimited identifiers

InterBase now supports SQL delimited identifiers. Delimited identifiers are database object names that are delimited by double quotes. SQL delimited identifiers are permitted only in InterBase 6 clients and databases using dialect 3.

In InterBase 6 clients and databases using dialect 3, a string constant is delimited by single quotes, and an SQL delimited identifier is delimited by double quotes. Because the quotes delimit the boundaries of the name, the possibilities for object names are greatly expanded from previous versions of InterBase.

InterBase object names can now:

- Be a keyword
- Use spaces, except for trailing spaces
- Use non-ASCII characters
- Be case sensitive

► *Migration*

Previous versions of InterBase allowed the use of both single and double quotes around string literals. In InterBase 6, double quotes are used only for delimited identifiers, and strings must be enclosed in single quotes. If you are thinking of migrating to dialect 3 or to any future versions of InterBase at any time in the future, you should always use single quotes to delimit strings. Refer to the Migration chapter of *Getting Started* for more information.

► *Error messages*

The following new error messages are associated with delimited identifiers:

Error Code	Number	SQL Code	Message Text
isc_ddl_not_allowed_by_db_sql_dialect	335544793	-817	Metadata statement is not allowed by the current database SQL dialect %d
isc_gfix_opt_SQL_dialect	335741039	None	Set database dialect <i>n</i>
isc_gfix_sql_dialect	112	None	SQL dialect must be one of %s
isc_invalid_string_constant	335544763	-104	Feature not supported

TABLE 2.2 Delimited ID error messages

InterBase Express for Delphi users

Borland Delphi 5 users can now use the InterBase Express™ (IBX) components to build InterBase database applications without the overhead of using the Borland Database Engine (BDE). IBX accesses the InterBase API directly, allowing increased speed and control within InterBase applications.

The version of IBX that comes with Delphi 5 addresses only InterBase 5 features. The IBX version that is included with InterBase 6 addresses all InterBase 6 features, using calls to the new Service API, Install API, and Licensing API, as well as the newest InterBase API.

The InterBase 5 version of IBX provides one additional tab in Delphi, labelled *InterBase*, that contains the IBX components for InterBase 5. The InterBase 6 version of IBX provides two tabs in Delphi: the *InterBase* tab is the same as in version 5 IBX; in addition there is an *InterBase Admin* tab. The *InterBase Admin* tab contains components that address the Services API, Install API, and Licensing API. It contains configuration, backup, restore, licensing, statistics, logging, and install, and uninstall components. The *InterBase Admin* tab is found at the extreme right of the tabs in Delphi 5. You will have to scroll to find it.

To install the InterBase 6 version of IBX, ensure that Delphi 5 is installed, Then display the *InterBase Launcher* and choose *InterBase Express*.

For more information on using IBX, refer to the *InterBase 6 Developers Guide* and the Delphi online help.

Large exact numerics

In dialect 3, InterBase 6 conforms with the SQL92 standard by storing NUMERIC and DECIMAL datatypes with 10 to 18 digits of precision as 64-bit integers (INT64 datatype). InterBase has always implemented NUMERIC and DECIMAL datatypes with precision less than 10 as exact numerics, but those with precision of 10 through 15 were implemented as DOUBLE PRECISION. Now, NUMERIC and DECIMAL datatypes are all stored as exact numerics. They are 16, 32, or 64 bit, depending on the precision. NUMERIC and DECIMAL datatypes with precision greater than 9 are referred to as “large exact numerics” in this discussion.

- These new 64-bit integer types are available in all contexts where datatypes are defined or used.

- NUMERIC and DECIMAL datatypes with a precision of 9 and scale *S* that caused arithmetic error messages in InterBase 5 return correct 64-bit results in InterBase 6.
- When an arithmetic operation on exact numeric types overflows, InterBase 6 reports an overflow error, rather than returning an incorrect value.
- If one operand is an approximate numeric, then the result of any dyadic operation (addition, subtraction, multiplication, division) is DOUBLE PRECISION.
- Any value that can be stored in a DECIMAL(18,*S*) can also be specified as the default value for a column or a domain.

► *Arithmetic operations*

All arithmetic operations in dialect 3 produce the results called for by the SQL92 standard, which in some cases are not the same results produced by previous versions of InterBase. The most notable problems when migrating to InterBase 6 involve the division operator and the AVG() function, which also implies division, with exact numeric operands.

SUM AND AVG

SUM() and AVG() return an exact numeric type if the subject row has an exact numeric type and the scaled sum fits in 64 bits; otherwise an exception is raised.

SUM() and AVG() operations on an exact numeric column return type NUMERIC(18,*S*) or DECIMAL(18,*S*), where *S* is the scale of the column.

If the datatype of the column is an approximate numeric (FLOAT, REAL, or DOUBLE PRECISION), SUM() and AVG() are computed using floating point arithmetic.

MIN AND MAX

MIN() and MAX() operations on an exact numeric column return an exact numeric result having the same precision and scale as the column.

ADDITION AND SUBTRACTION

If both operands are exact numeric, then adding or subtracting the operands produces an exact numeric with a precision of 18 and a scale equal to the larger of the two. For example:

```
CREATE TABLE t1 (n1 NUMERIC(16,2), n2 NUMERIC(16,3));
INSERT INTO t1 VALUES (12.12, 123.123);
COMMIT;
```

The following query returns the integer 135.243. The largest scale of the two operands is 3; therefore, the scale of the sum is 3.

```
SELECT n1 + n2 FROM t1;
```

Similarly, the following query returns the integer -111.003:

```
SELECT n1 - n2 FROM t1;
```

If either of the operands is approximate numeric (FLOAT, REAL, or DOUBLE PRECISION), then the result is DOUBLE PRECISION.

MULTIPLICATION

If both operands are exact numeric, then multiplying or dividing the operands produces an exact numeric with a precision of 18 and a scale equal to the sum of the scales of the operands. For example:

```
CREATE TABLE t1 (n1 NUMERIC(16,2), n2 NUMERIC(16,3));
```

```
INSERT INTO t1 VALUES (12.12, 123.123);
COMMIT;
```

The following query returns the integer 1492.25076 because n1 has a scale of 2 and n2 has a scale of 3. The sum of the scales is 5.

```
SELECT n1*n2 FROM t1
```

If one of the operands is approximate numeric (FLOAT, REAL, or DOUBLE PRECISION), then the result is DOUBLE PRECISION.

DIVISION AND AVG

If both operands are exact numeric, then multiplying or dividing the operands produces an exact numeric with a precision of 18 and a scale equal to the sum of the scales of the operands. If at least one operand of a division operator has an approximate numeric type (FLOAT, REAL, or DOUBLE PRECISION), then the result is DOUBLE PRECISION.

For example, in the following table defined in InterBase 6, division operations produce a variety of results:

```
CREATE TABLE t1 (i1 INTEGER, i2 INTEGER, n1 NUMERIC(16,2),
  n2 NUMERIC(16,2));
INSERT INTO t1 VALUES (1, 3, 1.00, 3.00);
COMMIT;
```

The following query returns the integer 0 because each operand has a scale of 0, so the sum of the scales is 0:

```
SELECT i1/i2 FROM t1
```

The following query returns the NUMERIC(18,2) value 0.33, since the sum of the scales 0 (operand 1) and 2 (operand 2) is 2.

```
SELECT i1/n2 from t1
```

The following query returns the NUMERIC(18,4) value 0.3333, since the sum of the two operand scales is 4.

```
SELECT n1/n2 FROM t1
```

In InterBase 5 and older, any of the above division operations return the DOUBLE PRECISION value 0.3333333333333333.

To obtain an InterBase 5 result when using InterBase 6, alter your query to cast at least one of the operands into an approximate type. For example,

```
SELECT i1/cast(i2 as double precision) from t1;
```

► Generators

Any value that can be stored in a DECIMAL(18,0) can also be specified as the value on a SET GENERATOR statement.

Generators return a 64-bit value, and only wrap around after 2^{64} invocations (assuming an increment of 1) rather than 2^{32} as in InterBase 5. Use an ISC_INT64 variable to hold the value returned by a generator.

► *Numeric input and exponents*

Any numeric string in **dsql** or **isql** that can be stored as a `DECIMAL(18,S)` is evaluated exactly, without the loss of precision that might result from intermediate storage as a `DOUBLE`. A numeric string is recognized by the **dsql** parser as a floating point value only if it contains an “e” or “E” followed by an exponent, which may be zero. For example, **dsql** recognizes 4.21 as a scaled exact integer, and passes it to the engine in that form. On the other hand, **dsql** recognizes 4.21E0 as a floating point value.

► *Changes to system tables*

Both the `RDB$FIELDS` table and the `RDB$FUNCTION_ARGUMENT` table contain a new column, `RDB$FIELD_PRECISION`, to store the precision for `NUMERIC` and `DECIMAL` datatypes. The `RDB$FIELD_PRECISION` column is of type `SMALLINT`.

In the `RDB$FIELDS` and `RDB$FUNCTION_ARGUMENTS` tables, the `RDB$FIELD_SUB_TYPE` column now holds the subtype for `NUMERIC` and `DECIMAL` datatypes. If `RDB$FIELD_TYPE` is 7 (`SMALLINT`), 8 (`INTEGER`), or 16 (`INT64`), then `RDB$FIELD_SUB_TYPE` has the following possible values:

0 or NULL	<code>RDB\$FIELD_TYPE</code>
1	<code>NUMERIC</code>
2	<code>DECIMAL</code>

For example, if a column is defined as `DECIMAL(13,4)`, the row in `RDB$FIELDS` has the following values:

	<code>RDB\$FIELD_TYPE</code>	<code>RDB\$FIELD_SUB_TYPE</code>	<code>RDB\$FIELD_PRECISION</code>	<code>RDB\$FIELD_SCALE</code>
Value	16	2	13	-4
Meaning	Stored as <code>INT64</code>	Type is <code>DECIMAL</code>	Precision is 13	Scale is 4

► *Migration*

If you back up a `NUMERIC` or `DECIMAL` column with a precision greater than 9 (for example, `NUMERIC(12,2)`) in an InterBase 5 or earlier database and restore the database as InterBase 6, the column is still stored as `DOUBLE PRECISION`. Because InterBase does not allow datatype conversions that could potentially result in data loss, you cannot use the `ALTER COLUMN` statement to change the column datatype from `DOUBLE PRECISION` to `INT64`. To migrate a `DOUBLE PRECISION` column to an `INT64` column, you must create a new `INT64` column and copy the contents of the older column into it.

In InterBase 6 dialect 3, when you create a `NUMERIC` or `DECIMAL` column with a precision greater than 9, data in it is automatically stored as an `INT64` exact numeric.

For more information, refer to the Migration chapter in *Getting Started*.

SQL DATE, TIME, and TIMESTAMP

The old InterBase `DATE` datatype, which contains both date and time information, is being replaced with the SQL92 standard `TIMESTAMP`, `DATE`, and `TIME` datatypes in dialect 3.

- In dialect 1, only `TIMESTAMP` is available. `TIMESTAMP` is the equivalent of the `DATE` datatype in previous versions. When you back up an older database and restore it in version 6, all the `DATE` columns and domains are automatically restored as `TIMESTAMP`.
- In dialect 3, `TIMESTAMP` functions as in dialect 1, but two additional datatypes are available: `DATE` and `TIME`. These datatypes function as their names suggest: `DATE` holds only date information and `TIME` holds only time.

`TIMESTAMP` is a 64-bit datatype and `DATE` and `TIME` are 32-bit datatypes.

Throughout this section, the `TIMESTAMP`, `DATE`, and `TIME` datatypes are collectively referred to as the *datetime* datatypes.

► *Querying the database for current date and time information*

The `CURRENT_DATE`, `CURRENT_TIME`, and `CURRENT_TIMESTAMP` functional operators return the date and time values based on the moment of execution of an SQL statement using the server's clock and time zone. It is no longer necessary to cast `TODAY` or `NOW` as `DATE` to obtain the current date, time, or timestamp.

For a single SQL statement, the same value is used for each evaluation of `CURRENT_DATE`, `CURRENT_TIME`, and `CURRENT_TIMESTAMP` within that statement. This means that if multiple rows are updated, as in the following statement, each data row will have the same value in the `aTime` column.

```
UPDATE aTable SET aTime = CURRENT_TIME;
```

Similarly, if row buffering occurs in a fetch via the remote protocol, then the `CURRENT_TIME` is based on the time of the `OPEN` of the cursor from the database engine, and not the time of delivery to the client.

► *Extracting date and time information*

The `EXTRACT()` function extracts date and time information from databases.

`EXTRACT()` has the following syntax:

```
EXTRACT (part FROM value)
```

The value passed to the `EXTRACT()` expression must be `DATE`, `TIME`, or `TIMESTAMP`. Extracting a part that doesn't exist in a datatype results in an error. For example:

```
EXTRACT (TIME FROM aTime)
```

A statement such as `EXTRACT (YEAR FROM aTime)` would fail.

The datatype of `EXTRACT()` expressions depends on the specific part being extracted:

Extract	Resulting datatype	Representing
YEAR	SMALLINT	Year, range 0-5400
MONTH	SMALLINT	Month, range 1-12
DAY	SMALLINT	Day, range 1-31

TABLE 2.3 Extracting date and time information

Extract	Resulting datatype	Representing
HOUR	SMALLINT	Hour, range 1-23
MINUTE	SMALLINT	Minute, range 1-59
SECOND	DECIMAL(6,4)	Second, range 0-59.9999
WEEKDAY	SMALLINT	Day of the week, range 0-6 (0 for Sunday, 1 for Monday, and so on)
YEARDAY	SMALLINT	Day of the year, range 1-366

TABLE 2.3 Extracting date and time information

► *Casting to and from date/time datatypes*

You can use the CAST() function in SELECT statements to translate between date/time datatypes and various character-based datatypes. The character datatype must be at least 24 characters in length. You can, however, cast a TIMESTAMP to a DATE and then cast the DATE to a CHAR of less than 24 characters. For example:

```
SELECT CAST (CAST (timestamp_col AS DATE) AS CHAR(10)) FROM table1;
```

It is not possible to cast a date/time datatype to or from BLOB, SMALLINT, INTEGER, FLOAT, DOUBLE PRECISION, NUMERIC, or DECIMAL datatypes.

For more information, refer to “Using CAST() to convert dates and times” in the *Embedded SQL Guide*.

Table 2.4 outlines the results of casting *to* date/time datatypes:

Cast From	To		
	TIMESTAMP	DATE	TIME
VARCHAR(<i>n</i>) CHARACTER(<i>n</i>) CSTRING(<i>n</i>)	String must be in format YYYY-MM-DD HH:MM:SS.thousands	See below.	String must be in format HH:MM:SS.thousands
TIMESTAMP	Always succeeds	Date portion of timestamp	Time portion of timestamp
DATE	Always succeeds; time portion of timestamp set to 0:0:0.0000	Always succeeds	Error
TIME	Always succeeds; date portion of timestamp set to current date	Error	Always succeeds

TABLE 2.4 Results of casting to date/time datatypes

Casting DATE to string results in YYYY-MM-DD where “MM” is a two-digit month. If the result does not fit in the string variable a string truncation exception is raised. In earlier versions, this case results in DD-Mon-YYYY HH:mm:SS.hundreds where “Mon” was a 3-letter English month abbreviation. Inability to fit in the string variable resulted in a silent truncation.

Casting a string to a date now permits strings of the form:

```
'yyyy-mm-dd'      'yyyy/mm/dd'      'yyyy mm dd'
'yyyy:mm:dd'      'yyyy.mm.dd'
```

In all of the forms above, you can substitute a month name or 3-letter abbreviation in English for the 2-digit numeric month. However, the order must always be 4-digit year, then month, then day.

In previous versions of InterBase, you could enter date strings in a number of forms, including ones that had only two digits for the year. Those forms are still available in InterBase 6. If you enter a date with only two digits for the year, InterBase uses its "sliding window" algorithm to assign a century to the years.

The following forms were available in earlier versions of InterBase and are still permitted in InterBase 6:

```
'mm-dd-yy'      'mm-dd-yyyy'      'mm/dd/yy'      'mm/dd/yyyy'
'mm dd yy'      'mm dd yyyy'      'mm:dd:yy'      'mm:dd:yyyy'
'dd.mm.yy'      'dd.mm.yyyy'
```

If you write out the month name in English or use a 3-character English abbreviation, you can enter either the month or the day first. In the following examples, "xxx" stands for either a whole month name or a three-letter abbreviation. All of the following forms are acceptable:

```
'dd-xxx-yy'      'dd-xxx-yyyy'      'xxx-dd-yy'      'xxx-dd-yyyy'
'dd xxx yy'      'dd xxx yyyy'      'xxx dd yy'      'xxx dd yyyy'
'dd:xxx:yy'      'dd:xxx:yyyy'      'xxx:dd:yy'      'xxx:dd:yyyy'
```

For example, the following INSERT statements all insert the date "January 22, 1943":

```
INSERT INTO t1 VALUES ( '1943-01-22' );
INSERT INTO t1 VALUES ( '01/22/1943' );
INSERT INTO t1 VALUES ( '22.01.1943' );
INSERT INTO t1 VALUES ( 'jan 22 1943' );
```

The following statement would enter the date "January 22, 2043":

```
INSERT INTO t1 VALUES ( '01/22/43' );
```

Table 2.5 outlines the results of casting *from* date/time datatypes:

Cast From	To VARCHAR(<i>n</i>), CHARACTER (<i>n</i>), or CSTRING(<i>n</i>)
TIMESTAMP	Succeeds if <i>n</i> is 24 or more. Resulting string is in format YYYY-MM-DD HH:MM:SS.thousands.
DATE	Succeeds if <i>n</i> is 10 or more. Resulting string is in the format YYYY-MM-DD.
TIME	Succeeds if <i>n</i> is 13 or more. Resulting string is the format HH:MM:SS.thousands.

TABLE 2.5 Results of casting to date/time datatypes

► *Adding and subtracting datetime datatypes*

The following table shows the result of adding and subtracting DATE, TIME, TIMESTAMP, and numeric values. “Numeric value” refers to any value that can be cast as an exact numeric value by the database engine (for example, INTEGER, DECIMAL, or NUMERIC).

Operand1	Operator	Operand2	Result
DATE	+	DATE	Error
DATE	+	TIME	TIMESTAMP (concatenation)
DATE	+	TIMESTAMP	Error
DATE	+	Numeric value	DATE + number of days: fractional part ignored
TIME	+	DATE	TIMESTAMP (concatenation)
TIME	+	TIME	Error
TIME	+	TIMESTAMP	Error
TIME	+	Numeric value	TIME + number of seconds: 24-hour modulo arithmetic
TIMESTAMP	+	DATE	Error
TIMESTAMP	+	TIME	Error
TIMESTAMP	+	TIMESTAMP	Error
TIMESTAMP	+	Numeric value	TIMESTAMP: DATE + number of days; TIME + fraction of day converted to seconds
DATE	-	DATE	DECIMAL(9,0) representing the number of days
DATE	-	TIME	Error
DATE	-	TIMESTAMP	Error
DATE	-	Numeric value	DATE – number of days: fractional part ignored
TIME	-	DATE	Error
TIME	-	TIME	DECIMAL(9,4) representing number of seconds
TIME	-	TIMESTAMP	Error
TIME	-	Numeric value	TIME – number of seconds: 24-hour modulo arithmetic
TIMESTAMP	-	DATE	Error
TIMESTAMP	-	TIME	Error
TIMESTAMP	-	TIMESTAMP	DECIMAL(18,9) representing days and fraction of day
TIMESTAMP	-	Numeric value	TIMESTAMP: DATE – number of days; TIME – fraction of day converted to seconds

TABLE 2.6 Adding and subtracting date/time datatypes

Note Numeric value + DATE, TIME, or TIMESTAMP is symmetric to DATE, TIME, or TIMESTAMP + numeric value.

Note the following differences between date/time arithmetic in InterBase 6 dialect 3 from earlier versions:

► *Using date/time datatypes with aggregate functions*

You can use the date/time datatypes with the MIN(), MAX(), COUNT() functions, the DISTINCT argument to those functions, and the GROUP BY argument to the SELECT() function. An attempt to use SUM() or AVG() with date/time datatypes returns an error.

► *Default clauses*

CURRENT_DATE, CURRENT_TIME, and CURRENT_TIMESTAMP can be specified as the default clause for a domain or column definition.

► *Error messages*

This feature introduces the following new error messages:

Error Code	Number	SQL Code	Message Text
isc_extract_input_mismatch	335544789	-105	Specified EXTRACT part does not exist in input datatype
isc_datatype_notsup	335544801	-901	Datatype not supported for arithmetic
isc_expression_eval_err	335544606	-902	Expression evaluation not supported/old.

TABLE 2.7 Datetime error messages

► *Migration*

Columns and domains that are defined as DATE datatype in InterBase 5 DATE appear as TIMESTAMP columns when the database is restored in InterBase 6. Note however, that a TIMESTAMP has 4 decimal points of precision, while a version 5 DATE datatype has only 2 decimal points of precision.

If you migrate your database to dialect 3 and you require only date or only time information from the TIMESTAMP column, you can use ALTER COLUMN to change the datatype to DATE or TIME. These columns each take only 4 bytes; whereas TIMESTAMP and the InterBase 5 DATE columns each take 8 bytes, so this could improve your data storage. If your TIMESTAMP column holds both date and time information, then you cannot change it to an InterBase 6 DATE or TIME column using ALTER COLUMN because ALTER COLUMN does not permit data loss.

The primary migration problem will exist in the source code of application programs that use the InterBase 5 DATE datatype. In InterBase 6, the DATE keyword represents a date only datatype, while in InterBase 5 DATE represents a date and time datatype. For more information on this and other migration issues, refer to the Migration chapter of *Getting Started*.

Changing column and domain definitions

You can now alter the name, datatype, and position of a column using the ALTER COLUMN clause of the ALTER TABLE statement. Extensions to the ALTER DOMAIN statement allow you to alter the name and datatype of a domain. This functionality is available in InterBase 6 dialects 1 and 3.

► *Altering columns in tables*

The new ALTER COLUMN clause of the ALTER TABLE statement allows you to change:

- The datatype of a field
- The name of a field
- The position of a field with respect to the other fields

The SQL syntax is shown here with the changes in bold type. Only part of the original syntax is shown:

```
ALTER TABLE table <operation> [,<operation> ...];
```

```
<operation> = {
    ADD col_def
    | ADD tconstraint
    | DROP col
    | DROP CONSTRAINT constraint
    | ALTER [COLUMN] col_name <alt_col_clause> }
```

```
<alt_col_clause> = TO new_col_name
    | TYPE new_col_datatype
    | POSITION new_col_position
```

For example, to change the name of a column:

```
ALTER TABLE table1 ALTER COLUMN field1 TO field2;
```

To change the datatype of a column:

```
ALTER TABLE table1 ALTER COLUMN field1 TYPE char(20);
```

To change the position of a field:

```
ALTER TABLE table1 ALTER COLUMN field1 POSITION 4;
```

The ALTER COLUMN clause of the ALTER TABLE expression is useful for migration.

► *Altering domains*

The ALTER DOMAIN statement has new options that allow you to change the name and datatype of a domain.

The SQL syntax is shown here with the changes in bold type:

```
ALTER DOMAIN name {
    SET DEFAULT {literal | NULL | USER}
    | DROP DEFAULT
    | ADD [CONSTRAINT] CHECK ( dom_search_condition )
    | DROP CONSTRAINT
    | TO new_domain_name
    | TYPE datatype}
```

For example, to change the name of a domain:

```
ALTER DOMAIN domain1 TO domain2;
```

The ALTER COLUMN clause of ALTER TABLE and the TYPE clause of ALTER DOMAIN do not allow you to make datatype conversions that could lead to data loss. For example, they do not allow you to change the number of characters in a column to be less than the largest value in the column.

► *Constraints on altering datatypes*

InterBase does not let you alter the datatype of a column or domain in a way that might result in data loss.

Table 2.8 lists valid datatype conversions. The columns list the target datatype and rows are the source datatype.

	Blob	Char	Date	Decimal	Double	Float	Integer	Numeric	Timestamp	Time	Smallint	Varchar
Blob												
Char		X										X
Date		X	X						X			
Decimal		X		X				X				X
Double		X			X	X						X
Float		X			X	X						X
Integer		X		X	X		X	X				X
Numeric		X						X				X
Timestamp		X	X						X	X		
Time		X							X	X		
Smallint		X		X	X	X	X	X			X	X
Varchar		X										X

TABLE 2.8 Possible datatype conversions using ALTER COLUMN and ALTER DOMAIN

Converting a numeric datatype to a character type requires a minimum length for the character type as listed in table 2.9:

Datatype	Minimum length for converted character type
Decimal	20
Double	22
Float	13
Integer	11
Numeric	22
Smallint	6

TABLE 2.9 Minimum character lengths for numeric conversions

► *Error messages*

This feature introduces the following new error messages:

Error Code	Number	SQL Code	Message Text
isc_dyn_char fld_too_small	336068816	-604	New size specified for column %s must be greater than %d characters.
isc_dyn_dependency_exists	336068814	-616	Column %s from table %s is referenced in %s.
isc_dyn_domain_name_exists	336068812	-612	Cannot rename domain %s to %s. A domain with that name already exists.
isc_dyn_dtype_conv_invalid	336068818	-688	Cannot change datatype for column %s from a character type to a non-character type.
isc_dyn_dtype_invalid	336068815	-688	Cannot change datatype for column %s. Changing datatype is not supported for BLOB or ARRAY columns.
isc_dyn_field_name_exists	336068813	-612	Cannot rename column %s to %s. A column with that name already exists.
isc_dyn_invalid_dtype_conversion	336068817	-604	Cannot change datatype for column %s. Conversion from base type %s to base type %s is not permitted.

TABLE 2.10 Alter column error messages

Read-only databases

You can now change InterBase databases to *read-only mode*. This provides enhanced security for databases by protecting them from accidental or malicious updates. Databases are always in read-write mode at creation time. You can change any InterBase 6 database, regardless of dialect, to read-only mode using **gbak** or **gfix**.

Both **gbak** and **gfix** now have a **-mode** option, which has a value of either **read_only** or **read_write**. The new syntax is as follows:

```
gbak -create -mode read_only old_database.gbk new_database.gdb
gfix -mode read_only database.gdb
```

Both of these operations require exclusive access to the database.

If an InterBase 6 client tries to set an InterBase 5 or older database to read-only, the server silently ignores the request. No error or warning is issued.

An InterBase 5 or older client can select from a read-only database. However, these older clients cannot distinguish between a read-only and read-write database. If an older client attempts to do anything other than select from an read-only database, the attempt fails with an error.

For information on creating a read-only database, see “Read-write and read-only databases” in the *Operations Guide*

► *Error messages*

This feature introduces the following new error messages:

Error Code	Number	SQL Code	Message Text
isc_read_only_database	445	817	Attempted to update read-only database
isc_gfix_opt_mode	109	901	-mode "read_only" or "read_write"
isc_gfix_mode_req	110	901	"read_only" or "read_write" required
isc_gbak_opt_mode	278	901	-mode access "read_only" or "read_write"
isc_gbak_mode_req	279	901	"read_only" or "read_write" required

TABLE 2.11 Read-only database error messages

New keywords

InterBase 6 introduces the following new keywords:

COLUMN	SECOND
CURRENT_DATE	SQL
CURRENT_TIME	TIME
CURRENT_TIMESTAMP	TIMESTAMP
DAY	TYPE
EXTRACT	WEEKDAY
HOUR	YEAR
MINUTE	YEARDAY
MONTH	

If you want to use these keywords as object names in databases using dialect 3, then you must delimit them with double quotes.

► *Migration*

You can migrate an InterBase 5 database that uses InterBase 6 keywords as identifiers to InterBase 6 dialect 1 without changing the names. These objects are then accessible to InterBase 5 clients, but not to InterBase 6 clients. To use the database in InterBase 6 clients, you must change the names to non-keywords. If you are going to migrate the database to dialect 3 immediately, you can retain the keyword identifiers, but must delimit them with double quotes.

For more information, refer to the Migration chapter in *Getting Started*.

Status vector and warning messages

The InterBase status vector is a mechanism that holds information about the current operation, where it is accessed via API calls. In previous versions of InterBase, the status vector contained only the error code, but in InterBase 6 it also contains the information about the source of the error message, and the error message type, one of error, warning, or informational. Warning and informational messages do not impede normal client/server operations, but may advise the client of a problem that needs investigation.

► *New isql parameter*

By default, **isql** displays any message returned in a status vector, even if no error occurred. To display warning messages if and only if an error occurs, specify **-nowarnings** on the command line when starting **isql**.

► *Using warnings*

Warnings can be issued for the following conditions:

- SQL statements with no effect
- SQL expressions that produce different results in InterBase 5 versus InterBase 6
- API calls which will be replaced in future versions of the product
- Pending database shutdown

New gbak functionality

The InterBase 6 **gbak** command incorporates the functionality of the version 5 **gsplit** utility, allowing the database owner or SYSDBA to back up to and restore from multiple files.

gbak now allows you to set databases to read-only with the **-mode read_only** switch. Refer to “Read-only databases” on page 30 for more information.

You can use **gbak**'s new **-service** switch to perform server-side backups. In this mode, **gbak** uses the new Services API and performs the backups on the server, incurring significantly less network traffic. Previously, backups were all performed on the client.

When using the **-service** switch, make sure that all path names to databases and backup files must be given relative to the server.

When backing up without using the Services API, backups are performed on the client platform where **gbak** is running. All pathnames to databases and backup files must be given relative to the client.

For more information on the **gbak** command, see the *Operations Guide*

► *Error messages*

This feature introduces the following new error messages:

Error Code	Number	SQL Code	Message Text
isc_gbak_page_size_missing	336330754L	-901	size specification either missing or incorrect for file <string>
isc_gbak_file_outof_sequence	336331015L	-901	file <string> out of sequence
isc_gbak_join_file_missing	336331016L	-901	can't join -- one of the files missing
isc_gbak_stdin_not_supptd	336331017L	-901	standard input is not supported when using join operation
isc_gbak_stdout_not_supptd	336331018L	-901	standard output is not supported when using split operation

TABLE 2.12 gbak error messages

Services API

The InterBase 6 Services API allows you to write applications that monitor and control InterBase servers and databases. Tasks that you can perform with this API include:

- Performing database maintenance tasks such as database backup and restore, shutdown and restart, garbage collection, and scanning for invalid data structures
- Creating, modifying, and removing user entries in the security database
- Administering software activation certificates
- Requesting information about the configuration of databases and the server

The Services API is a group of functions in the InterBase client library (**gds32.dll** on Windows, **libgds.a** on UNIX/Linux). The Services API family consists of the following four functions:

- **isc_service_attach()** initiates a connection to a specified Services Manager
- **isc_service_start()** invokes a service task
- **isc_service_query()** requests information or task results from the Services Manager
- **isc_service_detach()** disconnects from the Services Manager

The features that you can exercise with the Services API include those of the command-line tools **gbak**, **gfix**, **gsec**, **gstat**, and **iblicense**. The Services API can also perform other functions that are not provided by these tools.

For more information about the Services API, refer to “Working with Services” in the *API Guide*.

Install API

InterBase provides developers with a new group of functions that facilitate the process of silently installing InterBase as part of an application install on the Win32 platform. In addition, it allows you to interact with users if desired, to gather information from them and to report progress and messages back to them.

For more information refer to “Using the Install and Licensing APIs” in the “Developer’s Guide.”

Licensing API

The InterBase server functionality must be activated by installing *authorization codes* that are provided on software activation certificates obtained from InterBase. Each authorization code consists of a Certificate ID and Certificate key. You can activate the server as part of your install by using functions provided in the InterBase License API. If you do not activate the server as part of the install, it will be inactive until the end user provides authorization codes using IBConsole.

For more information refer to “Using the Install and Licensing APIs” in the “Developer’s Guide.”

InterBase Client for UNIX

InterBase 6 comes with a UNIX-based client package called “InterBase Client Only” (not to be confused with the InterBase 5.x driver called InterClient). The package contains a version for Solaris, which can be installed from the CD-ROM using the `setup.ksh` script. The `setup.ksh` script can also be used to install the InterBase documentation and the Adobe Acrobat Reader from the CD-ROM. For more information on installing and setting up the InterBase Client for UNIX, refer to the Installation chapter in *Getting Started*.

► Solaris

InterBase on Solaris uses a OS-specific install utility called `pkgadd`. Install the Solaris software using the `setup.ksh` script. For more information, refer to the Installation chapter in *Getting Started*.

New On-Disk Structure (ODS)

InterBase 6 databases stores data using the On-Disk Structure (ODS) version 10, which is required to use delimited IDs, exact numerics, and SQL DATE, TIME, and TIMESTAMP datatypes.

IMPORTANT InterBase 6 servers can connect only to ODS version 10 databases, and InterBase 5 servers can connect only to databases with an ODS version of 9 or 8. See the section on Migration in *Getting Started* for more information and for how to make existing tables compatible with this new ODS.

New error messages

InterBase 6 has the following new error messages, in addition to those listed in sections above.

Error Code	Number	SQL Code	Error message text
isc_ext_readonly_err	335544651L	-816	I/O error for file <string> -Error while trying to write to file -Cannot insert because the file is read-only or is on a read only medium.
isc_ext_file_delete	335544786L	-901	Cannot delete rows from external files.
isc_ext_file_modify	335544787L	-901	Cannot update rows in external files.

TABLE 2.13 InterBase 6 error messages

Running InterBase 6 as an application

In Windows NT, InterBase 6 runs as a service; that is, the InterBase Server and Guardian start automatically when you boot up your system. To run InterBase 6 as an application that you can start from the desktop:

1. In the Control Panel, choose Services.
2. In the Services window, select InterBase Guardian and click Stop.

3. Click Startup and select Disabled as the Startup type.
4. Create a shortcut to *ib_install_dir*\bin\ibguard.exe.
5. Open the Properties for the shortcut.
6. Click the Shortcut tab.
7. At the end of the Target field, add **-a** to the target statement, after the quotation mark. For example:

```
"C:\Program Files\Borland\Interbase\bin\ibguard.exe" -a
```

Once you have completed these steps, you can double-click the shortcut icon to start the InterBase Guardian and Server.

Compiling and linking

This section explains the compiling and linking options that are available for creating InterBase applications. These steps apply whether the code is output from the **gpre** embedded SQL preprocessor or you're using the InterBase API to code your application. You must link your applications only with the shared GDS library. The pipe server functionality is made obsolete by the SuperServer architecture.

Note It is highly recommended to use an IDE to create a Project for your applications on Windows, instead of using command-line compilation.

► *Windows NT and Win95/98*

Borland C++ Builder 3.0

```
bcc32 -a4 -tWM -tWC -I<interbase_home>\include
  application.c -eapplication.exe <interbase_home>\lib\gds32.lib
```

C and C++ Microsoft Visual C++ 5.0

```
cl -W3 -G4 -Gd -MD -I<interbase_home>\include application.c
  <interbase_home>\lib\gds32_ms.lib /Feapplication.exe
```

► *Solaris*

C SPARCWorks 4.2

```
cc -mt -w -I/usr/interbase/include application.c
cc -mt application.o -o application -lgdsmt
  -lsocket -lthread -lnsl -ldl
```

C++ SPARCWorks 4.2

```
CC -mt -w -I/usr/interbase/include -c application.C
CC -mt application.o -o application -lgdsmt
  -lsocket -lthread -lnsl -ldl
```

Known bugs and workarounds

Here are brief descriptions of the known bugs in this release. For more information, or possible workarounds, go to the InterBase Knowledge Base at <http://www.interbase.com/cgi/kbase.cgi>.

Bug Number	Description
57825	Invalid SDL can crash engine
57972	Inappropriate bug check when out of disk space
58277	Using Events with TCP/IP eventually hangs InterBase
60104	The control word for the numeric coprocessor is altered by the client library (gds32.dll) for certain calculations and then not changed back when it is passed back to the application. The result is that incorrect floating point numbers are sometimes returned.
60130	Operations on corrupted databases crash IBServer.exe
60131	Restore on multiple tapes does not work
60152	ISC_COMMIT_RETAINING updates are very slow over time
60188	Stored Procedures can be set to be undeletable
60338	Competing garbage collection processes corrupt the database
60435	Dynamically loading and unloading of gds32.dll causes resource leaks
60549	If you add a user with the Security API, and you specify a username flag, but no value, the server crashes
61482	A dialect 1 client attaching to a dialect 3 database can select and insert from columns with the new V6 datatypes, such as TIME, DATE, and TIMESTAMP.

TABLE 2.14 Known bugs