

InterBase 6

Release Notes



Borland®

SOFTWARE CORPORATION

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

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

Copyright 2001 BorlandSoftware Corporation. All rights reserved. All InterBase and Borland products are trademarks or registered trademarks of BorlandSoftware Corporation. Other brand and product names are trademarks or registered trademarks of their respective holders.

1INT0050WW21003 5E4R0898

9899000102-9 8 7 6 5 4 3 2 1

D4

Release Notes 1

CHAPTER 1

General Information

Contacting Borland	6
Installation and Migration	7
InterBase Client for UNIX	7
Migration to InterBase 6	7
Running InterBase 6 as an application	7
Uninstalling InterBase 6	8
System requirements	8
InterBase documentation	8
Documentation installation	9
Full-text searching	10
Links	10

CHAPTER 2

InterBase 6 features

Feature overview	11
IBConsole	13
SQL Dialects	15
Read-only databases	17
Language elements	17
New keywords	17
SQL delimited identifiers	18
New On-Disk Structure (ODS)	18
SQL DATE, TIME, and TIMESTAMP	19
EXTRACT() function	20
CAST() function	20
Changing column and domain definitions	21
Command-line tool enhancements	23
New gbak functionality	23
isql enhancements	23
gpre and gfix enhancements	24
API enhancements	24
Services API	24

Install API.	25
Licensing API.	25
Status vector and warning messages.	25
InterBase Express for Delphi and C++ Builder users .	26
New connectivity tools	26
Arithmetic operations	27
SUM and AVG	27
MIN and MAX	27
Addition and subtraction	27
Multiplication.	28
Division and AVG	28
Generators	29
Numeric input and exponents	29
Large exact numerics	29
Changes to system tables	30
Replication	31

CHAPTER 3

Error Information	
New error messages	33
Fixed bugs, known bugs, and workarounds	36

General Information

This chapter covers the following topics:

- Contacting Borland
- Installation and Migration
- System requirements
- Documentation and Adobe Acrobat Reader

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

Contacting Borland

► *Mailing address*

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

► *World-wide web sites*

- 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://www.community.borland.com/interbase>
- For installation and presales questions, visit:
<http://www.borland.com/devsupport>
- To discuss issues, such as features and potential bugs, with other InterBase users, visit:
<http://www.borland.com/newsgroups>

► *Email addresses*

- For questions about product information, product release schedule, features, feature requests, and VAR partnerships, send email to:
interbase@borland.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@borland.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/>

TABLE 1.1

Installation and Migration

This section contains information on installation, migration, and setting up UNIX clients. Complete installation instructions are in the installation chapter of *Getting Started*.

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*.

Migration to InterBase 6

Migrating databases from previous versions to InterBase 6 requires planning. First you must learn about:

- Transition features (features whose meanings have changed between InterBase versions)
- Dialects (indicators that specify how transition features are interpreted)
- On-disk structure (differences in architecture).

Begin by reading chapter two of these release notes to learn more about dialects and how features have changed in this release, then read the Migration chapter in *Getting Started*.

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\Interbase Corp\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.

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.

System requirements

Installation requires approximately 44MB of disk space for a full install that includes InterBase, Adobe Acrobat Reader, and the full document set. Only 17MB is needed to install the InterBase product without the documents, examples, or Acrobat Reader. For a complete listing of disk requirements, system hardware, OS and compiler requirements, refer to the installation chapter in *Getting Started*.

InterBase documentation

InterBase supplies online documentation in Adobe Acrobat PDF format. All documents except *Release Notes* and *Getting Started* are also available in soft-cover books. The document set consists of the following:

Document Title	Contents
Release Notes	Contact information, new feature descriptions, known problems and work-arounds
Getting Started	Complete installation information, how to migrate from previous versions of InterBase
Operations Guide	Configuring and operating InterBase Server, using command-line and GUI tools
Data Definition Guide	Designing and building InterBase databases, using database utilities
Developer's Guide	Developing InterBase database applications using Borland development tools
Embedded SQL Guide	Developing InterBase database applications using embedded SQL
API Guide	Developing InterBase database applications using the InterBase Application Programming Interface
Language Reference	InterBase SQL, DSQL, isql, and stored procedure and trigger language elements, syntax and semantics

TABLE 1.2 InterBase documentation set

Documentation installation





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.


To access all of the books from a single document, open [2AllBooks.pdf](#). It contains links to the rest of the document set. In addition to InterBase's own documentation, there are a number of documents for the EasySoft connectivity product set. You can access all of them through links in [2AllConnectivity.pdf](#).

IMPORTANT You need Adobe Acrobat Reader With Search to view and search these documents. See the Getting Started manual 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 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**.

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 configure and maintain an InterBase server, create and maintain databases on that server, and execute interactive SQL. IBConsole replaces the earlier Server Manager and InterBase Windows ISQL GUI environments. See **“IBConsole” on page 13**.

- SQL 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 InterBase 6 features. See **“SQL Dialects” on page 15**.

■ 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 17**.

■ Language Elements

InterBase 6 contains several new and upgraded language elements, including new keywords, delimited identifiers, large exact numerics, additions to the ALTER DOMAIN syntax, and a new ALTER COLUMN clause of ALTER TABLE to change the name, position, and datatype of columns in tables. InterBase 6 includes three datatypes for handling date and time: TIMESTAMP, DATE, and TIME. 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 **“Language elements” on page 17** for more information.

■ Command-line tool enhancements

InterBase 6 contains enhancements to the gbak, gfix, gpre, and isql utilities:

- The **gbak** utility can back up to and restore from multiple files, perform server-side backups using InterBase’s new Service Manager, and can be accessed from within IBConsole. See **“New gbak functionality” on page 23**.
- The **gfix** and **gpre** utilities now allow you to set the database dialect. This functionality is also available from within IBConsole. See **“gpre and gfix enhancements” on page 24**.
- 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. You can run **isql** sessions from within IBConsole. See **“isql enhancements” on page 23**.

■ API Enhancements

InterBase 6 contains several API enhancements, including:

- **Services API**. Services API library functions allow you to programmatically monitor and control InterBase servers and databases. See **“Services API” on page 24**.
- **Install API**. Install API library functions 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 25**.
- **Licensing API**. Licensing API library functions allow you to check, add, remove, and view certificate ID and key pairs (authorization codes). See **“Licensing API” on page 25**.

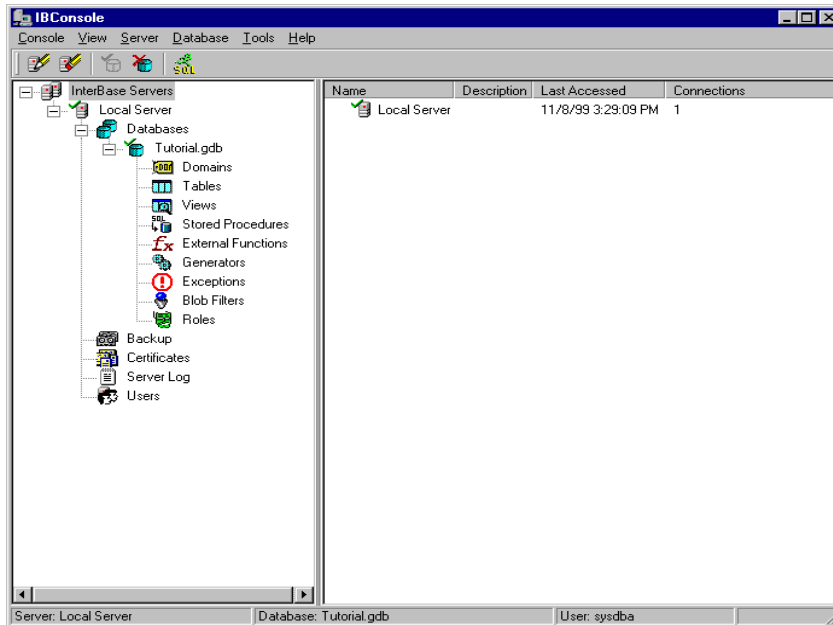
- **InterBaseExpress™ (IBX).** InterBase 6 includes a version of IBX that addresses InterBase 6 features, including the new Install API and Licensing API. This IBX version is a superset of the version included with Delphi 5. See **“InterBase Express for Delphi and C++ Builder users” on page 26.**
- **Status vector.** 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 25.**
- **New connectivity tools**
 The ODBC driver provided with earlier versions of InterBase has been replaced with EASYSOFT, an InterBase-specific server that can access databases on multiple servers using existing operating system and network infrastructures, while increasing network speed and security. See **“New connectivity tools” on page 26** for more information.
- **Arithmetic Operations**
 InterBase 6 contains several changes to arithmetic operations, SQL functions, generators, and numeric input and exponents. See **“Arithmetic operations” on page 27** for more information.
- **Changes to System Tables**
 InterBase 6 contains enhancements to the RDB\$FIELDS and RDB\$FUNCTION_ARGUMENTS tables. For a descriptions of these enhancements, see **“Changes to system tables” on page 30.**
- **Replication**
 InterBase 6 contains IBReplicator, which facilitates replication and synchronization between multiple InterBase databases that have similar structure. See **“Replication” on page 31.**

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, run 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 or sweep a database
- View database and server statistics
- Validate the integrity of a database
- Recover “in limbo” transactions

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

SQL 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.

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

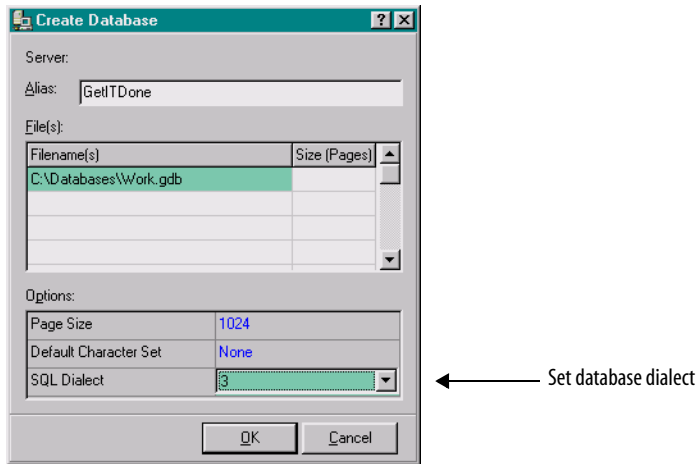
- Anything delimited by double quotes
- Date/time datatypes
- Large exact numerics (DECIMAL and NUMERIC datatypes with precision greater than 9)

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.

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.

SETTING/CHANGING THE CLIENT DIALECT IN IBCONSOLE

To set or change the dialect of a client in IBConsole:

1. Access the **Tools** menu.
2. From the **Tools** menu, select the **Interactive SQL** entry.
3. From the **Interactive SQL Edit** menu, select **Options**.
4. In the **Options** tab, click the number next to **Client Dialect** and select a dialect from the dropdown list.

For further information on using IBConsole and **isql** to set client and database dialects, see the *Operations Guide*.

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*.

Language elements

New keywords

InterBase 6 introduces the following new keywords:

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

Note To use a keyword as an object identifier (such as a column name), in dialect 3 you must enclose the identifier in double quotes (“”). These keywords are described further in the following sections.

If you are upgrading from a previous version of InterBase, see the Migration chapter in *Getting Started* for more information on how keywords are affected by the new dialects.

SQL delimited identifiers

InterBase now supports SQL delimited identifiers. This means that InterBase object names can:

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

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.

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.

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 on how to make existing tables compatible with this new ODS.

SQL DATE, TIME, and TIMESTAMP

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

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.

► *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.

► *Migration*

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.

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. 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*.

EXTRACT() function

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 a DATE, TIME, or TIMESTAMP datatype. Extracting a part that doesn't exist in a datatype results in an error. For example, the following expression would fail:

```
EXTRACT (YEAR FROM aTime)
```

For more information, refer to EXTRACT() in the *Language Reference Guide*.

CAST() function

You can use the CAST() function in SELECT statements to translate between date/time datatypes and various character-based datatypes.

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*.

Note that there are some differences from behavior in past versions of InterBase:

- Casting DATE to string results in YYYY-MM-DD where “M”M 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 DATE supports YYYY-MM-DD and other unambiguous input formats such as YYYY-Mon-DD. YYYY-DD-MM is not supported. Two-digit years are not allowed. In previous versions, input format was MM-DD-YYYY HH:mm:ss.hundreds or DD.MM.YYYY HH:mm:ss.hundreds. Two-digit years were interpreted as being within 100 years of the current year.

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 domains*

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

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

► *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

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 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.

Table **2.1** lists valid datatype conversions. The columns list the target datatype and rows are the source datatype.

	Blob	Char	SQL Date	Decimal	Double	Float	Integer	Numeric	Timestamp	Time	Smallint	Varchar
Blob												
Char		X										X
SQL 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 2.1 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 2.2:

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

TABLE 2.2 Minimum character lengths for numeric conversions

Command-line tool enhancements

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 17** 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*.

isql enhancements

- At the command line, you can invoke **isql** with the **-sql_dialect *n***, where *n* is 1, 2, or 3.
- Within a 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.
- 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.

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. You can also use API functions to set and query dialects:

- *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.

gpre and gfix enhancements

Use the gpre command line option **-sql_dialect *n***, where *n* is 1, 2, or 3, to set 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.

Use the gfix command line option **-sql_dialect *n***, where *n* is 1 or 3, to set the dialect of an ODS 10 version database.

API enhancements

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*.

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, either 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.

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

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**.

InterBase Express for Delphi and C++ Builder users

Borland Delphi and C++ builder 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.

New connectivity tools

InterBase now ships with a series of standards-based client/server middleware products that allow clients to access database on multiple servers using existing operating system and network infrastructures, while increasing network speed and security. The thin-client architecture facilitates installation and administration, and prevents client conflict with multiple DLLs, databases, and network stacks.

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.

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.

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	RDB\$FIELD_TYPE
1	NUMERIC
2	DECIMAL

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

	RDB\$FIELD_TYPE	RDB\$FIELD_SUB_TYPE	RDB\$FIELD_PRECISION	RDB\$FIELD_SCALE
Value	16	2	13	-4
Meaning	Stored as INT64	Type is DECIMAL	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*.

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 *Operations Guide*.

The version of IBReplicator for InterBase 6 contains the following new features:

- A new command, **Remove System Objects**, removes all triggers and other system objects from a source database without having to remove target databases from the schema.
- All configuration databases now have an option to change/enter the Current Schema Number. This is required in complex replication environments; see IBReplManager.hlp for details.
- The Replication Server can automatically map all tables and fields on the source database to tables and fields of the same name on the target database.

This release also fixes bugs related to the **Create System Objects** command, recovery from lost/completed connections, replicated deletes with NUMERIC and INT64 primary keys.

CHAPTER

3

Error Information

This chapter lists error messages introduced in InterBase 6 and provides descriptions and workarounds, if available, for known bugs.

New error messages

Below are tables listing new error messages for this release of InterBase. These error messages are sorted by feature, for example, dialects error messages or gbak error messages. InterBase 6 also has the following new global error messages, in addition to those listed below.

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 3.1 InterBase 6 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 3.2 Dialect error message

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 3.3 Delimited ID error messages

The datetime 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 3.4 Datetime error messages

Alter column 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 3.5 Alter column error messages

The read-only database 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 3.6 Read-only database error messages

The gbak 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 3.7 gbak error messages

Fixed bugs, known bugs, and workarounds

Here are brief descriptions of the bugs in this release of InterBase. For more information, search the Borland community site at <http://search.borland.com>.

Bug Number	Description
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

TABLE 3.8 Fixed bugs

Here are brief descriptions of the known bugs in this release. For more information, or possible workarounds, search the Borland community site at <http://search.borland.com>.

Bug Number	Description
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

TABLE 3.9 Known bugs