Release Notes

InterBase 5.5

September 1998

About this document

This Release Notes file provides the following:

- Contacting InterBase Software Corporation
- General information
  - The install directory
  - System requirements
  - Documentation and Adobe Acrobat Reader
  - System requirements
- New InterBase 5.5 features
- InterBase 5.0 and 5.1.1 features
- Migration issues
- Compiling your applications
- Known bugs in InterBase 5 with workarounds
- List of bugs fixed in InterBase 5.0, 5.1.1, and 5.5

Contacting InterBase Software Corporation

- *Mailing address*

  InterBase Software Corporation
  100 Enterprise Way, Suite B2
  Scotts Valley, CA 95066
  Fax (831) 431-6512

- *World-wide web site*

  InterBase Software Corporation maintains an Internet site on the world-wide web. The URL of this site is:

  http://www.interbase.com/

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

- *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 world wide, look at the following web page:

    http://www.inprise.com/bww/

General information

This section describes document options for InterBase 5 and lists system requirements for each platform.

The InterBase install directory

Throughout this document, *<interbase_home>* refers to the InterBase install directory. By default, this is */usr/interbase* on UNIX platforms, and *C:\Program Files\InterBase Corp\InterBase* on Windows NT and Windows 95. You can use the INTERBASE environment variable to change this location. See the *Operations Guide*, Chapter 4, "Server Configuration" for more information on environment variables.

Installation instructions

Complete installation instructions are in Chapter 3 of the *Operations Guide*.

Installation requires approximately 36MB of disk space for a full install that includes InterBase, InterClient, Adobe Acrobat Reader, and the full document set. Only 11MB is needed to install the InterBase product without InterClient, the documents, or Acrobat Reader.

Documentation in PDF form

InterBase provides all five 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 5 product. This requires about 15MB of disk space. In addition, the document set and Release Notes are available on the CD-ROM in uncompressed PDF form in the */doc* directory. You can read them directly from that location if you don't want to install them on your system.

You need Adobe Acrobat Reader With Search to view and search these documents. See "Installing Acrobat Reader With Search" on page 6 of this document for 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 TOC and Index entries are hypertext links and therefore are clickable. Throughout the document set, clickable links appear **bold and green**.

Navigating in Acrobat Reader

Acrobat Reader can display the document with a *bookmarks* window pane on the left. The bookmarks are a clickable list of the contents of the book. To display the bookmarks, click the second button on the toolbar (), press Ctrl-7, or use the menu item "Bookmarks and Page" under the View menu.

Reader also provides a rich array of navigation aids. Check the View menu for options, including keyboard shortcuts.

Installing Acrobat Reader With Search

ON WINDOWS 95 AND NT

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

ON HP-UX AND SOLARIS

InterBase 5 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 or Windows NT platform, run *setup.exe* in the /Adobe/Windows directory.

FROM THE ADOBE WEBSITE

If you don't have the InterBase CDROM handy, you can get Acrobat Reader for free from the Adobe website. It's at http://www.adobe.com/prodindex/acrobat/readstep.html. Be sure to download Acrobat Reader With Search, not the plain Acrobat Reader.

System requirements

Microsoft Windows NT and Windows 95

Operating system: Windows NT 4.0 with Service Pack 3 and the Teardrop2 hotfix or Windows 95

Memory: 16 megabytes minimum; 64 or more recommended

Processor/Hardware model: 486 minimum; Pentium II recommended

C/C++ compilers: Microsoft Visual C++ 4.2 or Borland C++ 5.0

Sun Solaris

Operating system: Solaris 2.5.x or 2.6.x

Memory: 32 megabytes minimum; 64 or more recommended

Processor/Hardware model: SPARC or UltraSPARC

C compiler: SPARCWorks 4.2, V SC 3.0.1

C++ compiler: SPARCWorks 3.0.1, V SC 3.0.1

Fortran compiler: SPARCWorks 4.2, V SC 3.0.1

COBOL compiler: MicroFocus COBOL 4.0

Ada compiler: SPARCWorks 4.2, V SC 3.0

Hewlett-Packard HP-UX

Operating system: HP-UX 10.20

HP DCE/9000 runtime support (DCE-Core) must be installed

Memory: 32 megabytes minimum; 64 or more recommended

Processor: PA-RISC

C compiler: HP C/HP-UX Version A.10.32

C++ compiler: HP C++/HP-UX Version A.10.22

Fortran compiler: HP Fortran/9000 10.20

COBOL compiler: MicroFocus COBOL 4.0

Ada compiler: ALSYS ADA 5.5.4

Hardware Model: HP/9000 Series 700 or 800

Patch number	Series 700	Series 800
PHKL_8376	✔	
PHKL_13611	✔	
PHNE_13265	✔	
PHNE_13469	✔	
PHCO_13626	✔	✔
PHSS_10565	✔	✔
PHKL_8377		✔
PHKL_13612		✔
PHNE_13266		✔
PHNE_13468		✔

TABLE 1 Required HP-UX operating system patches

New InterBase 5.5 features

This section lists features introduced in the 5.5 version of InterBase. The following section reviews features that were introduced in InterBase 5.0 and 5.1.1.

Greatly enhanced stability and more efficient memory use

To respond to the priority requests of customers, the InterBase 5.5 release has great improvements in performance, stability, and reliability. The InterBase Software Corp. engineers fixed many bugs and generally addressed all issues that affect stability in a multi-user environment.

- Memory use is significantly more efficient than in previous releases
- It is safe to update metadata in a database that is in use
- New error messages help database developers identify faults in their UDF code
- InterBase can serve a greater number of concurrent clients without loss of stability or reasonable performance

New ODBC driver

The INTERSOLV® DataDirect™ 3.11.01 ODBC driver for InterBase implements the Microsoft ODBC 3.0 standard and is faster and more reliable that the Visigenic ODBC driver that shipped with earlier versions of InterBase. It supports multithreading, and the use of character sets.

Correction The INTERSOLV ODBC driver supports **SQL Roles** as a DSN property and as a connection string option. This is a recent change to the driver, made after the print date for the InterBase 5.5 manuals. The manuals incorrectly state that the ODBC driver does not support SQL Roles. See the ODBC help file, *IBINT13.HLP*, for details on support for SQL Roles.

InterClient 1.5

InterBase 5.5 ships with InterClient 1.5. This latest release of InterClient runs at up to 30 times the speed of InterClient 1.0 and is more reliable. In addition, it supports 23 international character sets.

Transparent upgrade

The version 5.5 release of InterBase does not require any changes to client applications.

New On-Disk Structure (ODS)

The On-Disk Structure (ODS) for InterBase 5.5 has been updated to version 9.1, which supports cascading referential integrity, index garbage collection, and SQL roles. See the section on Migration in the *Operations Guide*, Chapter 1, "Introduction" for more information and for how to make existing tables compatible with this new ODS.

Changed UDF functionality and improved documentation

InterBase's multi-threaded server architecture requires some care in the way memory is allocated and released in user-defined functions (UDFs) and in the way these UDFs are declared. In the new single-process, multithreaded architecture, memory allocated dynamically is not released, since the process does not end. In addition, users running UDFs concurrently use the same static memory space, with predictably disastrous results. InterBase's new FREE_IT keyword allows InterBase users to write thread-safe UDF functions without memory leaks.

Documentation of these changes has been greatly expanded in the InterBase 5.5 document set. For information about writing and declaring UDFs and about handling memory for them, refer to the entry for DECLARE EXTERNAL FUNCTION and Chapter 5 in the *Language Reference*, to Chapter 11 of the *Data Definition Guide*, and to Chapter 10 of the *Programmer's Guide*.

Correction Previous versions of InterBase manuals instructed Delphi programmers to use **sysalloc()** to allocate memory. This is an error. Programmers writing UDFs for InterBase 5.5 should use **ib_util_malloc()**. See "Handling memory for return values" on page 221 of the *Programmer's Guide* for more information.

InterBase 5.5 introduces a mechanism for detecting exceptions in UDFs. If your UDF has a defect that raises a runtime exception, the **ibserver** process returns an ISC error code. You don't need to change your UDF programming methods to take advantage of this feature. The error codes are returned to the client application; use **isc_interprete()** to retrieve the text of the error message based on the error code. The error message text is also written to the *interbase.log*.

The list of exception error codes and associated messages is listed below:

Error token	Error code	Error message
isc_exception_access_violation	335544768L	Access violation. The code attempted to access a virtual address without privilege to do so.
isc_exception_datatype_missalignment	335544769L	Datatype misalignment. The attempted to read or write a value that was not stored on a memory boundary.
isc_exception_array_bounds_exceeded	335544770L	Array bounds exceeded. The code tried to access an array element that is out of bounds.
isc_exception_float_denormal_operand	335544771L	Float denormal operand. One of the floating-point operands is too small to represent a standard float value.
isc_exception_float_divide_by_zero	335544772L	Floating-point divide by zero. The code attempted to divide a floating-point value by zero.
isc_exception_float_inexact_result	335544773L	Floating-point inexact result. The result of a floating-point operation cannot be represented as a decimal fraction.
isc_exception_float_invalid_operand	335544774L	Floating-point invalid operand. An indeterminate error occurred during a floating-point operation.
isc_exception_float_overflow	335544775L	Floating-point overflow. The exponent of a floating-point operation is greater than the magnitude allowed.
isc_exception_float_stack_check	335544776L	Floating-point stack check. The stack overflowed or underflowed as the result of a floating-point operation.
isc_exception_float_underflow	335544777L	Floating-point underflow. The exponent of a floating-point operation is less than the magnitude allowed.
isc_exception_integer_divide_by_zero	335544778L	Integer divide by zero. The code attempted to divide an integer value by an integer divisor of zero.
isc_exception_integer_overflow	335544779L	Integer overflow. The result of an integer operation caused the most significant bit of the result to carry.
isc_exception_unknown	335544780L	An exception occurred that does not have a description. Exception number *number*.
isc_exception_stack_overflow	335544781L	Stack overflow. The resource requirements of the runtime stack have exceeded the memory available to it.

TABLE 2 UDF exception handling error code and messages

IMPORTANT Do not rely on this feature as a substitute for writing robust UDF code. Defects in UDF code that raise exceptions are likely to cause other problems in the **ibserver** process, possibly resulting in data corruption. Use this feature during development to identify and diagnose defects in your UDFs. Do not use UDFs that raise exceptions on a production server. If InterBase raises any exception errors in your production environment, you are strongly recommended to shut down **ibserver** immediately, and to debug your UDFs.

Specify database cache size on restore

The new **-buffers** switch for **gbak** lets you specify a default cache size for a database when you are restoring it with **gbak**. (The switch is disallowed when creating backups.)

Ownership retained on database restore

gbak -r and **gbak -c** now retain the original ownership of a database and its objects when a database is restored. Earlier versions of InterBase assigned ownership to the login that was performing the restore.

New error messages

InterBase 5.5 adds the following error messages, in addition to those listed under the changed UDF functionality above.

Error token	Error code	Error message
isc_sort_rec_size_err	335544758L	An operation attempted to sort records that are too big
isc_bad_default_value	335544759L	Cannot assign a NULL default value to a column with a NOT NULL constraint
isc_invalid_clause	335544760L	"invalid clause *string*" where "*string*" is the text that the user tried to enter that was determined to be invalid (for example, "NOT NULL DEFAULT NULL")"
isc_too_many_handles	335544761L	Returned if a connection tries to open more than USHORT handles against a port
isc_optimizer_blk_exc	335544762L	Optimizer implementation limits are exceeded; for example, only 256 conjuncts (ANDs and ORs) are allowed

TABLE 3 New error messages in InterBase 5.5

InterBase 5.0 and 5.1.1 new features

This section describes features that were introduced in InterBase 5.0 and 5.1.1.

Cascading declarative referential integrity

The definition for FOREIGN KEY has been extended to support the SQL 2 standard CASCADE feature for declarative referential integrity. This feature provides a mechanism for defining the actions to be taken in secondary tables when updating or deleting the primary key. You can use this new definition in CREATE TABLE and ALTER TABLE statements.

InterBase 5 enforces compliance with the SQL 92 standard. Refer to the *Data Definition Guide*, Chapter 6, "Working with Tables" for a full description of integrity constraints. Refer to the *Language Reference* for the complete syntax of the CREATE TABLE and ALTER TABLE statements.

New UDF library

InterBase now provides a number of frequently needed functions in the form of a UDF library, which is named *ib_udf.dll* on Windows platforms and *ib_udf* on UNIX platforms. These UDFs are all implemented using the standard C library. This section describes each UDF and provides its declaration.

Refer to *Language Reference*, Chapter 5, "User-Defined Functions" for documentation on declaring and using the functions found in the Interbase UDF library.

Index garbage collection

InterBase 5 performs garbage collection on indexes. Index garbage collection dynamically decreases the size of an index when an index page becomes empty as records are deleted. InterBase retains the recovered disk space for its own use; the space is not returned to the operating system. Index garbage collection is available only in newly created databases and in older databases that have been restored using the new InterBase 5 **gbak** utility.

New international character sets

InterBase 5 supports the following international character sets:

- BIG_5 (Chinese)
- KSC5601 (Korean)
- GB2312-80 (Chinese)

New security check for reference privileges (GRANT REFERENCES)

A security check for REFERENCES privileges allows the owner of a table to allow or disallow reference to its primary key from a foreign table.

Refer to the entry for the GRANT statement in the *Language Reference* for details on using GRANT REFERENCES.

SQL roles (CREATE ROLE, GRANT, CONNECT)

InterBase now supports SQL roles. This is a four-part process: you first use CREATE ROLE *rolename* to create the role. After creating the role, you use GRANT to grant privileges to the role and then use GRANT again to grant the role to users. A user must then state the role as part of a CONNECT statement to acquire the privileges granted to that role. Refer to the entries for the GRANT, REVOKE, and CREATE ROLE statements in the *Language Reference* for details on group privileges. **Note** This feature is not supported by clients using the BDE middleware. To use SQL roles with visual components in Delphi, use the third-party components Free IB Components (found on www.interbase.com) or IB Objects (found on www.ibobjects.com).

Improvements in gbak and multifile backup (gsplit)

gbak speed has been improved for both backup and restore functions. Previously the output from **gbak** had to be a file. **gbak** can now backup using *stdout* as the backup device, allowing output to be piped to other utilities or sent directly to a tape device. Previously when you ran **gbak** *create/replace*, the input had to come from a file; now **gbak** accepts *stdin* as input. **Note** This feature is supported only on UNIX and Linux.

See the *Operations Guide*, Chapter 8, "Database Backup and Restore" for full documentation on **gbak**.

gsplit is a new command-line utility that works with **gbak** to create backup files larger than the OS limit for file size. You can use **gsplit** to split the backup file created by **gbak** into multiple files. You can also use **gsplit** to restore a database from several files.

Refer to the *Operations Guide*, Chapter 8, "Database Backup and Restore" for full documentation on using **gsplit** with **gbak**.

New temporary file management

InterBase 5 includes a whole new concept of how temporary file space is managed. InterBase 5 creates two types of temporary files: *sort files* and *history list files*. For full documentation on temporary file management, see the *Operations Guide*, Chapter 4, "Server Configuration."

Cache configuration

You can set the size of the default cache for a specific database or server-wide and can modify the cache size for a specific ISQL connection. See the *Operations Guide*, Chapter 7, "Database Configuration and Maintenance" for documentation on cache specification.

New user-management API calls

Authors of InterBase applications can now add, delete, and modify users using three new API functions:

Function	Description
isc_add_user( )	Adds a user record to the password database
isc_delete_user( )	Deletes a user record from the password database
isc_modifiy_user( )	Modifies a user record in the password database

See the *API Guide* for reference documentation on use of these API functions.
Note These API functions are a temporary solution to provide programmatic user management capabilities to the product. InterBase 6.0 will introduce a more extensive collection of API functions for administering servers and databases; the three API functions above will become deprecated at that time.

Improved query optimization

Query optimization has been greatly improved in InterBase 5, reducing the need for you to formulate your own query plans. The following are some of the more notable improvements: The DISTINCT operator has been optimized to use an index where possible; ordering of multi-table joins has been improved with more accurate cost estimation techniques; SQL '92 JOIN syntax has been optimized for INNER JOINs; the optimizer now removes redundant sorting; the optimizer chooses the minimal index set; the PLAN clause now optimizes statements that include an OR specified as part of the WHERE clause; the plan report now includes SORT information;

InterBase Windows ISQL interface enhancements

The InterBase Windows ISQL window is resizable; the View menu and the Extract menu have been combined to form the Metadata menu; there is a new Query menu with entries for executing the current query and for displaying the previous or next query in the SQL Input Area; the Windows ISQL window has been updated to allow for the display of ROLE information; and you can now choose to extract metadata to a .sql file rather than a .ddl file. See the *Operations Guide*, Chapter 10, "Interactive Query" for full documentation on InterBase Windows ISQL and the **isql** command-line tool.

Performance monitoring

You can now use the **iblockpr** (**gds_lock_print** on UNIX) utility to monitor performance by checking lock requests. For details of using this utility, see the *Operations Guide*, Chapter 9, "Database and Server Statistics."

New error codes

Applications written for InterBase V3.3 or V4.0 that check for specific values of SQLCODE should be reviewed against changes in the error code list. See the *Language Reference*, Chapter 6, "Error Codes and Messages" for a complete list of error codes. The following error codes are new in InterBase 5:

Error token	Error code	Error message
isc_udf_exception	335544740L	A fatal exception occurred during the execution of a user defined function.
isc_lost_db_connection	335544741L	connection lost to database
isc_no_write_user_priv	335544742L	User cannot write to RDB$USER_PRIVILEGES
isc_token_too_long	335544743L	token size exceeds limit
isc_max_att_exceeded	335544744L	Maximum user count exceeded. Contact your database administrator.
isc_login_same_as_role_name	335544745L	Your login *login* is same as one of the SQL role name. Ask your database administrator to set up a valid InterBase login.
isc_reftable_requires_pk	335544746L	"REFERENCES table" without "(column)"; requires PRIMARY KEY on referenced table
isc_usrname_too_long	335544747L	the username entered is too long. Maximum length is 31 bytes.
isc_password_too_long	335544748L	the password specified is too long. Maximum length is 8 bytes.
isc_usrname_required	335544749L	a username is required for this operation.
isc_password_required	335544750L	a password is required for this operation
isc_bad_protocol	335544751L	the network protocol specified is invalid
isc_dup_usrname_found	335544752L	a duplicate user name was found in the security database
isc_usrname_not_found	335544753L	the user name specified was not found in the security database
isc_error_adding_sec_record	335544754L	an error occurred while attempting to add the user

TABLE 4 New error messages in InterBase 5.0

isc_error_modifying_sec_record	335544755L	an error occurred while attempting to modify the user record
isc_error_deleting_sec_record	335544756L	an error occurred while attempting to delete the user record
eisc_rror_updating_sec_db	335544757L	an error occurred while updating the security database

TABLE 4 New error messages in InterBase 5.0

The InterBase SuperServer

SuperServer is new on UNIX platforms. On Windows platforms, it was implemented in Version 4.2, but it has not been formally documented on paper, since the previous full documentation release was for InterBase 4.0. The online Help system for Windows contains some information about SuperServer. SuperServer for both platforms is documented in this section.

The InterBase SuperServer architecture

SuperServer is a multi-client, multi-threaded implementation of the InterBase server process. This implementation replaces the "Classic" implementation used for previous versions of InterBase.

Classic architecture, the design in InterBase 4.0 and earlier, was process-based. For every client connection, a separate server process was started to execute the database engine, and each server process had a dedicated database cache. The server processes contended for access to the database, so a Lock Manager subsystem was required to arbitrate and synchronize concurrent page access among the processes.

SuperServer serves many clients at the same time using threads instead of separate server processes for each client. Multiple threads share access to a single server process. The benefits of SuperServer architecture include:

- Having a single server process eliminates bottlenecks resulting from arbitration for shared database pages and reduces the overhead required for multiple process startups and database queries.
- SuperServer improves message interaction performance because a shared library call is always faster than an interprocess communication request to a server process.
- SuperServer improves database integrity because only one server process has write access to the database, rather than one process for each client. All database engine functionality is encapsulated into a unified, protected subsystem that is isolated from user application error.
- SuperServer allows for the collection of database statistics and user information that InterBase's tools can use for performance monitoring and administrative tasks.
- SuperServer is more cost-effective than the Classic architecture. All operating systems have limits on the number of OS processes that can run concurrently. SuperServer allows for a fixed number of database threads to be multiplexed over a potentially large number of concurrent database connections. Since these threads are not hard-wired to any specific database connection, SuperServer can support a larger number of users with minimum resources use.

Managing the superserver

For documentation on starting and shutting down the InterBase SuperServer process, see the *Operations Guide*, Chapter 4, "Server Configuration."

Migration issues

See the section on Migration in the *Operations Guide*, Chapter 1, "Introduction" for an overview of new, changed, and obsolete components in InterBase 5 that could have some bearing on your upgrade to this current version.

Compiling and linking

Applications

This section explains the compiling and linking options that are available for creating InterBase 5 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

C and C++ Borland C++ 5.0

```
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 -c application.C
CC -mt application.o -o application -lgdsmt
  -lsocket -lthread -lnsl -ldl
```

HP-UX

C HP C/HP-UX Version A.10.32

```
cc -w -c application.c
cc -o application application.o -lgds -ldld
```

C++ HP C++/HP-UX Version A.10.22

```
CC -w -c application.C
CC application.o -o application -lgds -ldld
```

Fortran HP Fortran/9000 10.20

```
f77 -w -c application.f
f77 application.o -o application -lgds -ldld
```

User-defined function libraries

This section details the compiling and linking commands to use to build a user-defined function (UDF) library.

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

Windows NT and Win95

C and C++ Borland C++ 5.0

```
bcc32 -a4 -tWM -tWCD -I<interbase_home>\include
  udflib.c -eudflib.dll InterBase\lib\gds32.lib
```

C and C++ Microsoft Visual C++

Use the Visual C++ IDE to create a Project for a dynamic link library (DLL).

Delphi Borland Delphi 4.0

Use the Delphi IDE to create a Project for a dynamic link library (DLL).
All exported functions in your library are available as UDFs.

Solaris

C SPARCWorks 4.2

```
cc -O -KPIC -mt -c udflib.c
ld -mt -G -Bsymbolic udflib.o -lgdsmt -lm -lc -o udflib.so
```

HP-UX

C HP C/HP-UX Version A.10.32

```
cc -O +z -c udflib.c
ld -b udflib.o -lm -o udflib.sl
```

APPENDIX

A

Bug Lists

This chapter contains a list of some known bugs with workarounds, as well as a list of all bugs fixed for the 5.0, 5.1.1, and 5.5 releases of InterBase.

Known bugs with workarounds

The following list provides details on how to solve or work around selected InterBase issues. A comprehensive list of all outstanding bugs is not available.

Bug #3297

Problem

Parentheses are disallowed around query-expressions. There is no way to clarify ambiguous queries like:

```
SELECT * FROM TABLE_A UNION
SELECT * FROM TABLE_B UNION
SELECT * FROM TABLE_C
```

It's the same problem as the classic programming issue of nesting IF...IF...ELSE. This especially affects queries that mix UNION and UNION ALL.

Workaround

Avoid this type of complex query, and use several simpler queries instead.

Bug #3446

Problem

gpre does not support BASED_ON in ANSI C-style function parameter declarations:

```
int foo( BASED_ON states.population fpop,
    BASED_ON states.state fstate )
```

Workaround
Use traditional, "K&R" C-style function parameter declarations:

```
int foo(fpop, fstate)
BASED_ON states.population fpop;
BASED_ON states.state fstate;
```

Bug #7520

Problem
CREATE TABLE which references a field of another table as its foreign key fails if an explicit unique index has been created for that field in the other table.

Workaround
Do not create the unique index on a primary key column. A primary key already implicitly creates a unique index, so creating another one is unnecessary.

Bug #8251

Problem
isc_dsql_execute() API call does not work when used to submit an EXECUTE STORED PROCEDURE statement which does not take any input parameters.

Solution
Use **isc_dsql_execute2()** for EXECUTE statements that have no input parameters.
Use **isc_dsql_execute()** for EXECUTE statements that do have input parameters.

Bug #8412

Problem
Backing up with **gbak** is extremely slow when the database contains a large number of back record versions.

Workaround
Using **gbak -b -g** inhibits the normal garbage collection task; this alleviates most of the performance problem.

Bug #8429

Problem
If you fail to install a valid license file, and attempt to start the server by using **ibmgr**, the server starts even though **ibmgr** reports an error. But **ibmgr** doesn't shut down the server when requested if it has no access to verify the sysdba password.

Solution
To shut down the server forcibly, use **kill** to halt the **ibguard** and **ibserver** processes, in that order. This is safe to do, because if there is no license, then there is no way there could be any active transactions on the database.

Bug #8542

Problem
Applications with more than one **gpre**'d file don't link because of the duplicate symbol definitions.

For each **gpre**'d file there are variables created that are given a global scope (for example, *isc_trans*, or the status vector). When two or more **gpre**'d files are compiled/linked it always produces linker errors because these variables are defined multiple times.

Workaround
Use `#define` to change the name of the global variables to something specific to each file.

Bug #8591

Problem
Running **isql -extract** against an old-style database that has GDML definitions for domains, views, triggers or COMPUTED BY columns outputs a mixture of SQL and GDML. Those metadata constructs are stored in Blobs, and are not converted by the extraction step, as is all the other metadata. **isql** cannot read GDML, so in this case the output of **isql** cannot be read by **isql**.

Solution
After you extract the metadata with **isql -extract**, you must rewrite all GDML as equivalent SQL code before using it as an SQL script.

Note This bug affects only databases that contain certain types of GDML metadata. Any database that was conceived with InterBase V4.0 or later should not have any GDML metadata.

Bug #8600

Problem
The InterBase 4.2 client does not detect a server disconnection when connected via Netbeui/Named Pipes. Therefore client applications may occasionally fail with the error message, "no process is on the other end of the pipe."

Solution
The InterBase 5 client correctly detects a disconnected server and doesn't try to use a dead connection. Old client versions still have the bug. You should upgrade both the InterBase client and server software to InterBase 5.

Bug #8813

Problem
Loading DLL's on Windows NT doesn't work if you specify the DLL without using the .DLL file suffix, and the DLL is located under a directory whose name contains a dot. For example:

- 'C:\A.B\MYLIB' does not work

- 'C:\A.B\MYLIB.DLL' works
- 'C:\A_B\MYLIB' works
- 'C:\A_B\MYLIB.DLL' works

This is a Microsoft bug. It is independent of InterBase. The same behavior exists when you programmatically load a library with the Windows API function **LoadLibrary()**.

Workarounds

The easiest solution is to always explicitly use the .DLL file suffix (as in the second case above) when specifying DLL's, for instance in a DECLARE EXTERNAL FUNCTION statement.

Another solution is to make sure that the path to your DLL contains no dots.

TIP If you have existing UDFs declared in your database, and their module names do not have the .DLL file suffix, you can update the declarations by connecting as SYSDBA and executing the following SQL statement:

```
UPDATE RDB$FUNCTIONS
    SET RDB$MODULE_NAME = RDB$MODULE_NAME || '.DLL'
    WHERE RDB$MODULE_NAME NOT LIKE '%.DLL';
```

Bug #10069

Problem

Creating a specific kind of improper stored procedure results in corruption of a system index.

It is not permitted to create a procedure that references a generator that doesn't exist. This is an appropriate restriction. For example:

```
CREATE PROCEDURE BUG10069 RETURNS (X INTEGER) AS
BEGIN
    X = GEN_ID(MISSING_GENERATOR, 1);
END
```

If `MISSING_GENERATOR` does not exist, InterBase immediately returns a number of error messages such as:

```
Statement failed, SQLCODE = -104
invalid request BLR at offset 47
-generator MISSING_GENERATOR is not defined
```

```
Statement failed, SQLCODE = -902
internal gds software consistency check (invalid SEND request (167))
```

```
Statement failed, SQLCODE = -902
internal gds software consistency check (can't continue after
bugcheck)
```

InterBase fails to undo some of the work done by create procedure. Specifically, the system indexes RDB$INDEX_18 on table RDB$PROCEDURE_PARAMETERS, and RDB$INDEX_21 and RDB$INDEX_22 on table RDB$PROCEDURES are left in an corrupted state. You can test a database for this kind of corruption by using the command: **gfix -v -f *database.gdb***. **gfix** detects but does not correct the corruption. Look for errors:

```
DBSERVER (Server)       Mon Aug 31 13:47:05 1998
  Database: D:\DATABASE.GDB
  Index 1 is corrupt (missing entries) in table RDB$PROCEDURES(26)
```

```
DBSERVER (Server)       Mon Aug 31 13:47:05 1998
  Database: D:\DATABASE.GDB
  Index 2 is corrupt (missing entries) in table RDB$PROCEDURES(26)

DBSERVER (Server)       Mon Aug 31 13:47:05 1998
  Database: D:\DATABASE.GDB
  Index 1 is corrupt (missing entries) in table
   RDB$PROCEDURE_PARAMETERS (27)
```

Data in the database is unaffected. Only the indexes named above are damaged.

Solution

The easiest solution is prevention. Always create a generator before you use it in a stored procedure.

If you find that you have referenced a non-existent generator and you have corrupted the index, you can use the following methods to repair the damaged indexes:

- Rebuild the indexes by using the following commands:

```
ALTER INDEX RDB$INDEX_18 INACTIVE;
ALTER INDEX RDB$INDEX_18 ACTIVE;
```

Repeat these steps with RDB$INDEX_21 and RDB$INDEX_22.

- Backing up and restoring the database forces InterBase to rebuild all indexes, so this is also an effective way to repair the corrupted indexes.

Note **gfix -mend** does not fix this type of index corruption.

IMPORTANT Always use the **gfix -v -f** command after attempting to repair the corrupted indexes, to verify that the indexes have in fact been repaired in the correct database.

Bug #10072

Problem

The server crashes when you execute the following or similar SELECT statement:

```
SELECT RDB$DB_KEY FROM ANY_STORED_PROCEDURE;
```

Workaround

Do not select the RDB$DB_KEY pseudocolumn in a query that invokes a Select Procedure.

Bug #10098

Problem

If you back up a database that has domains defined but no tables, restoring the backup file gives non-fatal error messages reporting "DEADLOCK." You can use the restored database, but this is not recommended because you cannot back it up again.

Workaround

Before you back up a database with domains, create at least one table.

Connection error: "REMOTE INTERFACE not licensed"

Problem

On Windows NT or Windows 95, after you uninstall InterBase 4.x and install InterBase 5.x, you are no longer able to connect to remote InterBase servers with BDE clients. You get the following error message:

```
REMOTE INTERFACE not licensed.
```

All InterBase tools (Windows ISQL, ComDiag) work normally. Local connections using BDE tools (for example, Delphi, Paradox, C++Builder) work normally. Only remote connections using BDE tools fail.

Solutions

There are several causes for this situation. Use the following troubleshooting methods:

- Find and remove old copies of *GDS32.DLL*, especially those that are in the BDE install directory. You must have only one copy of *GDS32.DLL* on a given system, and it must match the version of the InterBase client you use. Right-click on a file and check the Properties sheet to determine its version.
- If this does not fix the problem, it is possible that your InterBase client does not have a legitimate remote client software activation key. Use the License Manager tool (**iblicense.exe**) to verify that the InterBase Remote Client certificate is installed. If the certificate is not installed, add it by specifying the certificate ID `ISC-30811`, and the certificate key `ca-9-3a-0`.
- Your client workstation might not have the InterBase SQL-Links driver installed, or has an outdated version of the BDE or InterBase SQL-Links driver. Upgrade your software. Check the INPRISE web site for downloadable updates.
- If you uninstalled an old version of BDE before installing an updated version, you might get this error message. Remove all files from the old version of BDE before installing a new version.
- Finally, the entry in the Windows Registry for the BDE's DLLPATH might be invalid. Run **regedit** and find the Registry key: *HKEY_LOCAL_MACHINE/ Software/ Borland/ Database Engine*. The item *DLLPATH* should indicate the directory where BDE and SQL-Links DLL's are located. Check that the registry item lists the correct directory, and fix the entry if necessary.

Installation error: "Internal error near IBcheck"

Problem

Trying to install InterBase on Windows NT or Windows 95 from the JBuilder or Delphi CDROM produces the error message:

```
String variable is not large enough for string. Check the string
declarations. Error: 401
```

This is followed by the error:

```
Internal error near IBcheck
```

After this the installation aborts.

Solution

Run **regedit** and check the registry key *HK_CURRENT_USER/Environment*. The PATH item should be a registry item with a string type. JBuilder and Delphi use InstallShield to install the product. InstallShield has a bug such that it incorrectly creates PATH as a binary item. If the PATH item has a binary type, delete it or convert it to a string item. Then restart the installation of InterBase.

Bugs fixed for InterBase 5.0

Bug No.	Brief Description
	Bugs fixed for InterBase 5.0
1640	Nested subqueries are very expensive
2214	Optimizer uses redundant indexes.
3095	Near-identical query takes 400X cpu time
3516	Performance statistics produce negative number for elapsed time
3753	Performance problems over long-haul network connections
3870	Corrupt database
4181	Views based on equivalent queries were optimized differently
4742	Optimizer problem causing poor performance
4785	**gfix -write sync** is not saved if db is backed up & restored
5599	View performance problem
5977	Parser does not permit table update from another table
6498	Need a way to specify location of temp files and *interbas* directory.
6879	**gds** consistency check (differences record too long (182))
7039	Bad performance with table joins and indexes
7173	**gbak** names everything "volume 0"
7459	**gstat** doesn't say if db is shut down.
7565	Large PLANs are not displayed;
7953	Implement $INTERBASE for all product files
7973	Query on view fails to use index
7975	Vanishing client makes CPU grieve over new events
8001	GRANT ALL ON *table* TO *user* for 95 tables takes 5 minutes.
8039	A join of views give db corruption error
8054	No warning when a field that has a multisegment index on it is dropped

TABLE 5 Bugs fixed for InterBase 5.0

Bug No.	Brief Description
	Bugs fixed for InterBase 5.0
8055	**gbak** fails if index on integer w/scale is changed to just integer
8066	Remove "D" license requirement for GRANT/REVOKE operations
8071	No security on the RDB$USER_PRIVILEGES table
8072	Cannot restore **gbak**ed database if stored procedures use PLAN option
8073	**gbak** doesn't ABORT if stored procedures with PLAN clause depend on INACTIVE indexes
8076	SQL UDFs cannot return BLOBs
8086	*perf.c* elapse times < cpu times
8093	**isql** extract does not get the name of the RI constraint.
8097	Remote access via serial networks is unreliable
8104	**gds_inet_server** processes start up when unlicensed client fails to connect
8105	**gbak** performance is slow
8106	Duplicate indexed fields are not identified when brought from external file
8126	DSQL dumps core on a long illegal statement
8129	Correlated subquery crashes NLM, core dump on UNIX
8132	SYSDBA can't change privileges when moving v3 to v4
8134	**gpre** translates named transaction to gds_trans
8135	No error message return when connect in isql with bogus user and password.
8141	**blr_store2** core dumps
8149	Can (sometimes) delete from RDB$TRIGGERS
8150	Count of DISTINCT records returns wrong result
8158	DISTINCT clause causes ORDER BY clause to be ignored
8168	Reference privilege on tables is undocumented
8169	SQL doesn't allow GRANT to a group
8172	**isql** does not parse number of cache buffers
8180	DISTINCT causes incorrect ordering when used with ORDER BY
8183	Cannot transliterate character between char sets 3.3 >> 4.0.
8185	**gds_lock_print -i** core drops
8189	Remote prefetch loses accurate error codes
8193	Dropping procedure creates access violation

TABLE 5 Bugs fixed for InterBase 5.0

Bug No.	Brief Description
	Bugs fixed for InterBase 5.0
8201	Security class restrictions do not migrate to V4.0
8219	Trigger ABORTs but still deletes records
8238	Correlated subquery does not product right number. of columns
8249	**gds_lock_mgr**, **gds_pipe5**, **gds_pipe** closing only 20 file descriptors
8253	**gbak** fails when stored procedure inserts to non-updateable view
8254	LIBS 4.1 **gfix** is broken.
8255	**gpre** generates the wrong PIC X(8) value for COBOL programs
8261	**isql -x** does not extract domain information properly
8262	Error: Unsuccessful metadata update: depth exceeded (recursive definition).
8265	ON UPDATE SET DEFAULT feature (RI/CASCADE) fails
8269	Messages with a message length but no message are crashing the server
8270	A UDF returning a **cstring()** > 32752 causes a GPF in *ibserver.exe*
8284	Optimizer bug on same query with different positioning of fields.
8285	db file owned by root may not be the best design
8290	Solaris system stalls when running **gbak** utility
8293	Dynamic **gpre** gives segmentation violation with COBOL
8294	External table feature is a security hole
8295	External table doesn't get properly **gbaked** and restored
8309	Solaris system hangs when declaring referential integrity actions
8314	No option for *-user* or *-password* in **gsec**
8321	'C' and 'COBOL' give different error codes with same program
8322	**isql** extract of international domain produces incorrect syntax
8332	Differences record is too long (182)
8338	Column aliases do not work with UNION
8342	Character set name is included with the default for a field
8343	Stored procedure consistently crashes server
8344	No documentation for trigger updating through a view
8345	**gbak -o** fails with UDF db
8348	Calling **get_blob_segemnt**, **put_segemnt**, and **BLB_lseek** hangs SuperServer.

TABLE 5 Bugs fixed for InterBase 5.0

Bug No.	Brief Description
	Bugs fixed for InterBase 5.0
8349	SELECT query with ORDER BY crashes server
8350	Deadlock error in V4.2.1
8354	NULL XSQLDA in **isc_dsql_fetch()** causes remote server to crash
8359	4.5 Sync err, localhost err and server dies
8361	**isql -x** does not extract metadata from pre-4.5 databases
8365	**gbak** produces "deadlock" error during restore
8366	**gpre -x** does not work
8373	Problem dropping table in ISQL after stored procedures dropped
8377	Depth exceeded error from schema file in ISQL1_05 TCS test
8378	SuperServer uses "interbase", which is not an allowed username on HP10
8379	COBOL application returns sqlcode 0 when sqlcode -100 is expected
8395	Ambiguous positioned update of cursors with same name
8396	DSQL parser: cursor named after SQL keyword disables that keyword
8397	Constraint creates index, preventing DROP TABLE/COLUMN
8400	**isql** doesn't display not_null constraint
8402	Server crashes with SELECT statement involving large varchar / order by
8406	Operations do not correctly inherit privileges of procedures
8408	**gstat -x** does not display all available flags
8409	Cannot read error for **isc_guard1**.machine if start as interbase user
8414	Consistency chk error when GRANT is done before a grant to unix grp
8415	Error message with DISTINCT on subselect statements
8416	Error using SELECT DISTINCT on an indexed field with a stored procedure
8417	Server does not return memory to OS
8418	**gfix** shutdown timeout value appears not to work
8419	ALTER TABLE ... DROP CONSTRAINT command drops **ibserver**
8437	Lack of DROP GENERATOR statement should be explained in documentation
8441	**gbak** to tape does not work

TABLE 5 Bugs fixed for InterBase 5.0

Bug No.	Brief Description
	Bugs fixed for InterBase 5.0
8445	ALTER TABLE ... DROP (column) fails when it should succeed
8457	SELECT statement with singleton count crashes server
8465	**gpre** fails to add an "END - IF" in a COBOL program
8784	Indices were using character set ID instead of text type for length; usage of Portuguese collation (and others) would result in a "not found" error message.

TABLE 5 Bugs fixed for InterBase 5.0

Bugs fixed for InterBase 5.1.1

Bug No.	Brief Description
	Bugs fixed for InterBase 5.1.1
4840	Adding a column with DEFAULT clause fills column with zeroes; should be default value
7995	Cannot initialize event subsystem when the OS has inadequate semaphores
8589	License tool needs to enforce evaluation key; user should not remove eval key
8640	Updating or inserting values from one table to a second table crashes the server
8698	Expiration date of evaluation key is not Year 2000 correct
8725	Server occasionally fails to increment next transaction ID
8729	WISQL has no connection property for SQL roles
8732	SQL script hangs on GRANT statement; internal ACL data structure has static size
8734	**isql**'s flag for SQL roles doesn't work
8749	Grant ROLE to user is not generated by **isql -extract**
8752	Memory leak on server
8754	Server process size exceeds OS limit under heavy load; server hangs
8757	Server process grows in size without releasing memory
8781	Server does not release DSQL statement memory pools on database detach operation
8812	Database corruption; same cause as in bug 8732

TABLE 6 Bugs fixed for InterBase 5.1.1

Bug No.	Brief Description
	Bugs fixed for InterBase 5.1.1
8825	Temporary deadlock caused by **gfix -attach**
n/a	Server exits when lock manager memory is exceeded (UNIX only)
n/a	Miscellaneous memory leaks on client
n/a	No clear error message when memory is exhausted
n/a	Need IPX/SPX support in Win32 client in order to connect to NetWare servers

TABLE 6 Bugs fixed for InterBase 5.1.1

Bugs fixed for InterBase 5.5

Bug No.	Brief Description
	Bugs fixed for InterBase 5.5
8580	New Install script and IBLICENSE behavior not documented in the README
8676	Online help for WISQL not updated.
8732	IB server stalls when a Lock Manager contention occurs
8775	Execute stored procedure from InterClient application crashes ibserver
8780	Update from trigger doesn't get rolled back
8786	Transliteration exceptions converting Unicode_fss to DOS437
8787	Transliteration exceptions converting Unicode_fss to DOS850
8788	Transliteration exceptions and corruption converting Unicode_fss to DOS852
8789	Transliteration exceptions converting Unicode_fss to DOS857
8790	Transliteration exceptions & corruption converting Unicode_fss to DOS860
8791	Transliteration exceptions converting Unicode_fss to DOS861
8792	Transliteration exceptions converting Unicode_fss to DOS863
8793	Transliteration exceptions converting Unicode_fss to DOS865
8796	Transliteration exceptions converting Unicode_fss to NEXT
8798	Transliteration exceptions & corruption converting Unicode_fss to WIN_1250
8799	Transliteration exceptions & corruption converting Unicode_fss to WIN_1251
8801	Transliteration exceptions & corruption converting Unicode_fss to WIN_1253
8802	Transliteration exceptions converting Unicode_fss to WIN_1254
8803	Too many metadata versions corrupt a database (ALTER TRIGGER)

TABLE 7 Bugs fixed for InterBase 5.5

Bug No.	Brief Description
	Bugs fixed for InterBase 5.5
8812	A large number of GRANTs on a single table corrupts the database
8813	Cannot load gdsintl if the InterBase install path has a dot in it; document workaround
8822	Executing a view that is based on another view crashes the IB server
8825	GFIX -attach does not allow active transaction to complete
8842	Multi-user test allows only 96 NT connections.
8850	Restore of a backup fails if the database contains a stored procedure that references a view
8851	WARNING: column EM_FNAME is not defined in table RDB$PAGES
8864	**cast()** returns wrong result with NUMERIC numbers < 1
8867	Asking for better documentation on IB 5.0 license
8880	Casting numerics to CHARs fails when CHAR field is too large
8881	Casting the largest negative integer to CHAR produces error
8883	Error selecting data from an external file
8891	Too many (200+) ALTER statements in a script produce “request depth exceeded” error
8906	isc_info_base_level returns wrong version
8921	**gbak** does not preserve ownership of stored procedures
8926	**gbak** failure for some databases containing triggers for referential integrity
8927	**gbak** does not preserve ownership of tables
8935	GRANT on stored procedures produces inconsistent behavior
8936	**rtrim()** in a SELECT statement drops connection to IBServer
8937	**ltrim()**, **lower()**, and **rtrim()** return faulty length of a CHAR field
8938	**lower()** returns incorrect results
8945	UDF compiled with Borland C++ doesn't release memory
8949	Dropping a trigger causes core dump on HP-UX
8952	SYSDBA cannot revoke privileges from user when privileges have been proliferated to others
8955	ALTER TRIGGER INACTIVE causes core dump
8963	IB server fails on NT when triggers are repeatedly dropped or altered
8967	UDF functions sometimes crash the IB server
8968	**gbak** for IB5.1.1 cannot restore a .gbk file from previous version of IB
8987	**isql** script crashes server

TABLE 7 Bugs fixed for InterBase 5.5

Bug No.	Brief Description
	Bugs fixed for InterBase 5.5
8988	Altering a procedure while it is in use by another procedure results in an abortive bug check
8990	SHOW TRIGGERS is very slow due to unavailable system index
8995	Multistep triggers may not execute all steps
9015	Procedure ownership is not preserved through GBAK / RESTORE
9016	Database objects are owned by the user performing a restore
9018	Cannot alter a procedure that calls another procedure after its parameter list has changed
9019	CHECK constraints don't check trigger results
9023	IB server fails when a trigger is dropped while it is in use
9026	Possible lock manager problem
9027	IB server fails when a stored procedure is dropped immediately after execution
9033	SHOW GRANT functionality skips TRIGGER keyword
9038	Invalid check on EXECUTE permission for stored procedure
9053	Too many informational messages in interbase.log
9061	Improper pathname for **gsec** -database argument hangs NT server
10064	Rampant memory growth during restore of databases with certain combinations of dependant stored procedures
10085	Committing certain types of stored procedures crashes the server
n/a	**ibserver** crashes and/or corrupts data when more than 50 users put heavy demands on the server simultaneously and for an extended period of time

TABLE 7 Bugs fixed for InterBase 5.5