Release Notes

InterBase 5.1.1

February 1998

About this document

This Release Notes file provides the following:

- Contacting InterBase
- General information
  - The install directory
  - System requirements
  - Documentaion and Adobe Acrobat Reader
- New features in InterBase 5
- Using and configuring the InterBase 5 Guardian and server
- Migration issues
- Compiling your applications
- Known bugs in InterBase 5 with workarounds

Contacting InterBase Software Corporation

Mailing address

InterBase Software Corporation
1800 Green Hills Road
Scotts Valley, CA 95066
Fax (408) 430-1512

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

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 now provides all five books in the document set plus the Release Notes in PDF format. These documents have been indexed for full-text searching, so you need Adobe Acrobat Reader *With Search* to perform searches across the document set.

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 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.

The InterBase Document Set

You have the options 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.

System requirements

- 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: 10.20 release of HP Fortran/9000

  Hardware Model: HP/9000 Series 7xx or 8xx

  Required patches: PHKL_8376 (s700), PHKL_8377 (s800), PHSS_10565 (s700_800), PHNE_13265 (s700), PHNE_13266 (s800), PHNE_13469 (s700), PHNE_13468 (s800), PHKL_13611 (s700), PHKL_13612 (s800), PHCO_13626 (s700_800)

- 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 SC 4.2 C compiler

  C++ compiler: SPARCWorks SC3.0.1 C++ compiler

  Fortran compiler: SPARCWorks SC4.0 Fortran compiler

  COBOL compiler: MicroFocus Cobol 4.0

  Ada compiler: SPARCWorks SC4.0 Ada compiler

- Windows

  Operating system: Windows NT 4.0 or Windows 95

  Memory: 16 megabytes minimum; 64 or more recommended

  Processor/Hardware model: 486 minimum; Pentium recommended

  Compilers: Microsoft Visual C++ 4.2 or Borland C++ 5.0

New InterBase 5 features

This section describes new features in InterBase 5. The following section, "The InterBase SuperServer" on page 12, describes the functionality and operation of InterBase's SuperServer, which is new on Unix platforms, and which has been implemented on Windows only since Version 4.1.

- New On-Disk Structure (ODS)
- Cascade declarative referential integrity
- Changed UDF functionality
- New UDF library
- Index garbage collection
- New international character sets
- New security check for reference privileges (REFERENCES)
- Granting group privileges and SQL roles (CREATE ROLE, GRANT, CONNECT)
- New licensing structure (License Manager)
- Improvements in **gbak** and multifile backup (**gsplit**)
- New temporary file management
- Cache configuration
- New user-management API calls (**isc_add_user( )**, **isc_delete_user( )**, **isc_modify_user( )**)
- Improved query optimization
- Windows ISQL interface enhancements
- Performance monitoring
- New error codes

New ODS

The On-Disk Structure (ODS) for InterBase 5 has been updated to version 9.0, 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.

Cascade 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.

Changes in UDF functionality

The change to InterBase's single multi-threaded process requires some modification 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, multithread architecture, memory allocated dynamically is not released, since the process does not end. In addition, users running UDFs concurrently will be using 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.

UDFs written in Delphi must use **sysalloc** rather than **new**. Only **sysalloc** allocates dynamic memory that can be freed.

Refer to the *Language Reference*, Chapter 5, "User-Defined Functions" for more documentation on declaring and using UDFs.

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 indices. Index garbage collection dynamically decreases the size of an index when an index page becomes empty as records are deleted. As a result, less tuning is required. InterBase retains the recovered disk space for its own use; the space is not returned to the operating system.

NOTE 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.

Support for new international character sets

Additional character sets supported in InterBase 5 include:

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

New security check for reference privileges

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 the REFERENCES privilege.

Using SQL to grant and revoke group privileges

InterBase 5 implements features for assigning SQL privileges to groups of users. These features replace the Security Classes feature in past versions of InterBase. In the past, group privileges could be granted only through the InterBase-proprietary GDML language. In InterBase 5, new SQL features have been added to assist in migrating InterBase users from GDML to SQL.

Refer to the entries for the GRANT, REVOKE, and CREATE ROLE statements in the *Language Reference* for details on group privileges.

gbak improvements

- **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.

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

gsplit

gsplit is a new 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**.

Temporary file management

InterBase5 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."

Configuring the database cache

You can set the size of the default cache for a specific database or server-wide. You can also 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.

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. when a DISTINCT is specified on a JOIN, the DISTINCT is applied to one of the base tables before the JOIN is performed. The net result is a much more efficient use of resources. For example:

```
SELECT DISTINCT E.DEPT_NO FROM EMPLOYEE E, DEPARTMENT D
WHERE E.EMP_NO = D.EMP_NO;
```

returns the following plan:

```
PLAN JOIN (D ORDER RDB$PRIMARY5,E INDEX (RDB$FOREIGN8))
```

- Ordering of multi-table joins has been improved with more accurate cost estimation techniques.g
- SQL '92 JOIN syntax has been optimized for INNER JOINs: The optimizer now detects when more than one INNER JOIN has been specified and creates a plan based on the best order available. The INNER JOIN syntax for SQL '92 is now functionally equivalent to the old style comma list syntax. For example:

```
SELECT * FROM PROJECT P JOIN EMPLOYEE E
ON P.TEAM_LEADER = E.EMP_NO
JOIN DEPARTMENT D ON E.DEPT_NO = D.DEPT_NO;
```

now has the same performance and generates the same access plan as:

```
SELECT * FROM PROJECT P, EMPLOYEE E, DEPARTMENT D
WHERE P.TEAM_LEADER = E.EMP_NO
AND E.DEPT_NO = D.DEPT_NO;
```

- The optimizer removes redundant sorting: When DISTINCT, GROUP BY, and ORDER BY operators are used against the same fields. the optimizer removes redundant sorting. In particular, when these three clauses are used together against an indexed column(s), all three clauses will be mapped to the index and no sort will be necessary. For example:

```
SELECT DISTINCT LAST_NAME FROM EMPLOYEE
GROUP BY LAST_NAME ORDER BY LAST_NAME;
```

returns the following plan:

```
PLAN (EMPLOYEE ORDER NAMEX)
```

indicating that all three clauses will be handled in one pass via the index NAMEX. No sorts will be performed.

- The optimizer chooses the minimal index set: In previous versions, if there were multiple indices that could improve a query, all of them were used, even if they were duplicate indices. InterBase 5 uses only one index in these cases.
- The optimizer no longer performs the extra sort when a DISTINCT or an ORDER BY is applied to a stream for which a GROUP BY has already been done on a superset of the same fields. For example, this allows the following query to be performed without sorting if an index is available:

```
SELECT DISTINCT f1 FROM t1 GROUP BY f1 ORDER BY f1
```

- The PLAN clause now optimizes statements that include an OR specified as part of the WHERE clause.
- The plan report now includes SORT information: The SET PLAN feature reports when a sort is being performed. For example, with the query:

```
SET PLAN;
SELECT LAST_NAME FROM EMPLOYEE ORDER BY FIRST_NAME;
```

the following plan is reported:

```
PLAN SORT ((EMPLOYEE NATURAL));
```

indicating that the table will be fetched in natural order and the results will be sorted.

The SORT modifier is ignored when specified as part of the PLAN clause to a query. For example:

```
SELECT LAST_NAME FROM EMPLOYEE PLAN
SORT ((EMPLOYEE NATURAL));
```

is accepted, but does not force a sort of the stream when it is not called for by the query.

The SORT keyword is ignored in input and is not needed in order to induce a sort.

Windows ISQL interface enhancements (Windows only)

Several changes have been made to the Windows ISQL interface to make it easier to use. The major changes are as follows:

- The 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 three items: Execute executes the query that is currently in the buffer; Previous displays the last query that was executed; and Next loads the next query in the query buffer
- The Windows ISQL window has been updated to allow for the display of ROLE information
- You can now choose to extract metadata to a .sql file rather than a .ddl file
- The Open File dialog shows both .sql and .ddl files by default

See the *Operations Guide*, Chapter 10, "Interactive Query" for full documentation and reference of Windows ISQL and **isql**.

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:

- `isc_udf_exception` (335544740L). Windows NT & Windows 95 only. When an exception is generated by a user-defined function, this error is returned to the client. Text: `Exception` *`number`* `detected in blob filter or user defined function.`

- `isc_lost_db_connection` (335544741L). Windows NT & Windows 95 only. If a local client detects that the server has lost a connection to the database, the client will receive this error and it is the application's responsibility to attempt to reconnect to the database.
  Text: `Connection lost to database.`
- `isc_no_write_user_priv` (335544742L). For security reasons, clients are no longer allowed to perform write operations (INSERT, UPDATE, DELETE) on the system table RDB$USER_PRIVILEGES. Privileges must be granted or revoked using SQL GRANT and REVOKE statements.
  Text: `User cannot write to RDB$USER_PRIVILEGES.`
- `isc_token_too_long` (335544743L). Any single token in a DSQL statement may not exceed 255 characters.
  Text: `Token size exceeds limit.`
- `isc_max_att_exceeded` (335544744L). This is generated on an attempted database attachment if the number of connections time three exceeds the value in the USERS clause of the license file.
  Text: `Maximum user count exceeded. Contact your database administrator.`
- `isc_login_same_as_role_name` (335544745L).
  Text: `Your login` *`login`* `is the same as one of the SQL role names. Ask your database administrator to set up a valid InterBase login.`

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 last full document 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.

Windows NT and Win95

C and C++ Borland C++ 5.0

```
bcc32 prog.c <interbase_home>\lib\gds32.lib
```

C and C++ Microsoft Visual C++ 4.2

```
c1 prog.c
```

HP-UX

C HP C/HP-UX Version A.10.32

```
cc -w -c proc.c
cc -o prog prog.o -lgds -ldld
```

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

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

Fortran 10.20 release of HP Fortran/9000

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

Solaris

C SPARCWorks SC4.2

```
cc -w -c prog.c
cc prog.o -o prog -lgdsmt -lsocket -lthread -lnsl -ldl
```

C++ SPARCWorks SC3.0.1

```
CC -w -c prog.c
CC prog.o -o prog -lgdsmt -lsocket -lthread -lnsl -ldl
```

Known bugs with workarounds

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.

WORKAROUND

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 will start even though **ibmgr** reports an error. But **ibmgr** will not shut down the server when requested if it has no access to verify the sysdba password.

WORKAROUND

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

When compiling application with more than one **gpre**'d file it will not 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 will always produce 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 will output 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**.

WORKAROUND

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. Anyone 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.”

WORKAROUND

The InterBase 5 client correctly detects a disconnected server and will not 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 #8640

PROBLEM

Executing a Stored Procedure using one table to INSERT or UPDATE records in another table crashes the server. Symptoms include a stack overflow error.

```
create procedure test1 as
    declare variable v1 integer;
    declare variable  v2 integer;
begin
    for select f1, f2 from r1 into :v1, :v2
    do begin
        insert into r2 (f1, f2) values (:v1, :v2);
        when sqlcode -803 do
            update r2 set f2 = :v2
            where f1 = :v1;
    end
end
```

The procedure loops through several thousand rows of an input table and attempts to insert those values into an output table. If the INSERT fails because of a duplicate key error (-803), the procedure instead UPDATEs the existing row in the output table.

The observation about the "stack overflow" is appropriate. The server crashes due to the overflow, depending on the number of records which return a "duplicate error." This leads to recursion in the server (which is a bug) thereby exhausting stack space. This is not restricted to this error alone. It can happen with any error handler.

WORKAROUND

The user also reported that he got the procedure to run by breaking it into two procedures—one which did the outer select and one which did the insert or update. Following is the sample of procedures that work correctly:

```
create procedure add_or_update (v1 integer, v2 integer) as
begin
    insert into r2 (f1, f2) values (:v1, :v2);
    when sqlcode -803 do
        update r2 set f2 = :v2
        where f1 = :v1;
end

create procedure test2 as
    declare variable v1 integer;
    declare variable v2 integer;
begin
    for select f1, f2 from r1 into :v1, :v2
    do begin
        execute procedure add_or_update :v1, :v2;
    end
end
```

The recursion in the server is avoided by calling this new procedure **add_or_update** internally in procedure **test2**.

A "stack overflow" error, with disastrous consequences for the server, would occur if error handling code is invoked a large number of times (thousands), within one invocation of the original procedure.