InterBase XE Update 2 Readme

Updated: May, 2011

This file contains important information that may not appear in the online Help. Please read this file in its entirety.

The readme has five main sections:

System Requirements

For system requirements to install and run InterBase XE, please refer to the Installation, Registration, and Licensing Information document.

New Features

There are several new features available for this release:

OpenSSL Updated from 1.0.0a to 1.0.0d

InterBase XE Update 2 has upgraded from 1.0.0a to OpenSSL 1.0.0d.

For more information on using OpenSSL in InterBase XE, please reference *Network Configuration* in the Operations Guide, as well as *Encrypting your Data* in the Data Definition Guide.

For additional information on OpenSSL please refer to:http://openssl.org.

InterBase Direct I/O for DataBase Files

The direct Input/Output (I/O) capability makes InterBase more scalable with very large databases on systems where there are memory resource limitations. With the Update 2 release, InterBase enables this "direct I/O" functionality on Windows OS. This functionality circumvents the issue observed by many on Windows 2008 R2 and Windows 7 64-bit OS editions where System File Cache uses up too much physical memory leading to sluggish system performance.

Issue: InterBase uses buffered file I/O on all platforms to perform I/O on database pages for the file on disk. The pages are delivered via the System File Cache, which acts as a duplicate store of the pages on RAM. Subsequent loads of the same page(s) will be quickly served by the OS kernel if the page exists in the System File Cache. On systems where there is high contention with other files for the System File Cache (a shared pool used by all processes for buffered file I/O) the performance of InterBase may not be optimal. If available System File Cache is limited due to RAM resource limitations, the kernel must spend time cleaning up unused blocks of memory from other processes as well as provide for servicing a new block I/O request.

Solution: The performance problem is alleviated by using "direct I/O" (also known as non-buffered I/O) so blocks of pages are directly read from the disk into the process space and do not need to use the System File Cache.

Implementation/Usability

- The **gfix** command line tool has been modified to allow setting a database to be in "direct" I/O write mode.

```
# gfix {-write {async, sync, direct}] . . .
For example:

# gfix -write direct foo.ib -user sysdba -password masterkey
```

- The **gbak** command line tool now has a new restore option (optional) setting to override a database write mode. The "write" mode will be preserved during a backup/restore lifecycle.

```
# gbak [-write {async, sync, direct}] . . .

For example:

# gbak -write direct -r foo.ibk foo.ib -user sysdba -password masterkey
```

- **Services API** support for the new and updated *gfix* and *gbak* options
  - You can find the various new agruments and respectives values in ibase.h
  - **API Guide** Table 12.5: Service API database restore arguments

    Argument: `isc_spb_res_write_mode`

    Purpose: Set the write mode of the database: the next byte must be one of
    `isc_spb_res_wm_async`
    `isc_spb_res_wm_sync`
    `isc_spb_res_wm_direct`
    Corresponds to **gbak -write**

    Argument Length: 1 byte

    Argument Value: byte

  - **API Guide** Table 12.6: Service API database properties arguments

    Add `isc_spb_prp_wm_direct` to the following argument:
    `isc_spb_prp_write_mode`

- The **gstat** command line tool will exhibit the following setting, direct, in its "Attributes" header line output.

```
# gstat -h foo.ib -user sysdba -password masterkey
. . .

Database header page information:

     Flags 0

     Checksum 12345
     Write timestamp Mar 3, 2011 13:36:31
     Page size 8192
     ODS version 15.0
     . . .
     Creation date Feb 23, 2011 14:58:27
     Attributes force write, direct, no reserve

     Variable header data:
     Sweep interval: 20000
     *END*
    . . .
```

Requirements and Constraints

- Supported on **Windows OS only.**
- This setting is **not supported on non-Windows** platform databases; you will see the following error.

  ```
       feature is not supported
       -direct I/O operation
  ```

- If a database enabled with "direct" I/O is then copied to an older version of InterBase, the setting will not be used by the older InterBase server. The older server will employ the "sync" write mode in this case.

  Limitations:

  - direct" I/O setting on a database is only possible if the database page size is an exact multiple of the file's underlying disk sector size. The standard for so many decades has been 512 bytes per sector on hard disks. Newer hard disks however are trying to adopt the more Advanced Format of 4096 bytes per sector.

InterBase supports the following database page sizes: 1024, 2048, 4096, 8192 and 16384 bytes per page. Databases that have a page size of 1024 or 2048 bytes cannot be set to "direct" I/O on hard disks that only support the 4096 bytes per sector standard; you need to restore your database to a larger page size on such disks before enabling "direct" I/O on them.

If you try to enable "direct" I/O on an incompatible device, the following error message is returned stating the minimum required database page size. The following example shows an error message where the disk sector size is 4096 bytes.

```
   Error: Must backup and restore to DB page size >= 4096 bytes to support
direct I/O on this device.
```

Migration Issues

- A database needs to be set with "gfix -write direct" option and reloaded by the database engine for this to take effect.
- Since the System File Cache will not be used when "direct" I/O is set, it is recommended that the database cache setting and database linger interval be set suitably. This allows the most frequently used pages to be in memory, the InterBase database cache, when new connections are serviced.

InterBase 32-Bit and 64-Bit Native Binary Applications

For both the 32-bit and 64-bit editions, if you are still using dialect 1, you must migrate to dialect 3. Performance monitor tables will not work in the counters and will generate errors. For more information please reference *Understanding SQL dialects* in the Operations Guide.

InterBase 32-bit Native Binary Application

The ib_install.exe is delivered with the 32-bit Edition issue. You will need to run this installer if you want to to use the 32-bit Edition on Windows.

InterBase 64-Bit Native Binary Application

- With the 64-bit edition, you can continue to use 32-bit InterBase applications.
- Please note that the 64-bit kit cannot be installed on a 32-bit operating system.
- Because of InterBase�s focus on backward compatibility, it is easier to migrate to new editions.

The topics below cover the critical information you need to implement the 64-bit application.

Compatability Issues

- The local and remote connections from older clients are expected to work with newer 32-bit/64-bit servers; and vice-versa.
- IBMgr.exe and IBConsole.exe continue to be 32-bit applications. But they are expected to work with 32-bit and 64-bit servers.
- This version supports existing ODS 12.x and ODS 13.x databases that were introduced in previous releases of InterBase.
- You can expect to move databases between 32-bit and 64-bit kits of InterBase (with this version) as long as they support compatible ODS versions.

Client Library Name Changes for 64-Bit DLL's

The following table displays the new client library names for the 64-bit DLL's.

Library Name	Location	Notes
ibclient64.dll	<interbase>/bin	This is a new InterBase client DLL for native 64-bit applications. It is being used by 64-bit InterBase command-line tools currently, and will need to be deployed with 64-bit applications built by customers (in place of gds32.dll which is for the 32-bit target).
ibxml64.dll	<interbase>/bin	This is a new InterBase XML DLL for native 64-bit applications. It will need to be deployed with 64-bit applications built by customers (in place of ibxml.dll which is for the 32-bit target) if they are using the InterBase XML api.
ib_util64.dll	<interbase>/bin	This is a new InterBase UTILS DLL for native 64-bit applications. It will need to be deployed with 64-bit applications built by customers (in place of ib_util.dll which is for the 32-bit target).
ibclient64_ms.lib	<interbase>/SDK/lib_ms	Import library for building 64-bit applications targeting the ibclient64.dll.
ibxml64_ms.lib	<interbase>/SDK/lib_ms	Import library for building 64-bit

		applications targeting the ibxml64.dll.
ib_util64_ms.lib	<interbase>/SDK/lib_ms	Import library for building 64-bit applications targeting the ib_util64.dll.

JDBC Driver Updates for Blob/Clob Support

New interfaces have been implemented which support the following:

JDBC Name	InterClient Name	Exceptions/Comments
java.sql.Blob	interbase.interclient.Blob	For all API the parameter pos is ignored and assumed to be 1, hence the complete BLOB is returned. For example in the method OutputStream setBinaryStream(**long** pos) the parameter pos is ignored. The same is true for all other methods which take the "pos" parameter. **public long** position(byte[] pattern, **long** start) and **public long** position (java.sql.Blob blob, **long** start) are not supported.
java.sql.Clob	interbase.interclient.Clob	
java.io.inputStream	interbase.interclient IBBlobInputStream	Special implementation of the java.io.inputStream for InterBase Blob (and Clobs). Use the read() methods from this stream to access the underlying data.

The following methods have been implemented in this release:

In the java.sql.PreparedStatement class

- public void setObject (int parameterIndex, Object x) now works when parameter x is of type java.io.Inputstream.
- All variations of the setCharacterStream () method are implemented
- All variations of the setAsciiStream() and setBinaryStream() methods are implemented
- all variations of method setBlob() and setClob() are implemented.
- Method isClosed() is implemented.

In the java.sql.Result class

- All variations of the getCharacterStream () method are implemented
- All variations of method getBlob () and getClob() are implemented

Stronger Password Protection

This release implements stronger password protection on InterBase databases to comply with password requirements from the Payment Card Industry - Data Security Standard (PCI DSS). This additional functionality supports a longer effective password length, resulting in stronger password protection.

Requirements/Constraints

- This design supports server-wide user authentication as manifested by the USERS table of the security database, configured with the IBCONFIG.ADMIN_IB property parameter, which defaults to the file admin.ib.

- The design also supports EUA databases. As with the non-EUA databases, it also has to be explicitly enabled by the owner/administrator. Please note that the USERS table in admin.ib has RDB$USERS as the counterpart in EUA databases; so the earlier references have to be compatible with EUA database references.

- For EUA databases, the database metadata upgrade to 13.2 is automatic (for ODS 13.x databases); the database owner does not have to create the PASSWORD_DIGEST domain or modify the RDB$USERS table explicitly to set 'DES-CRYPT' for existing user accounts. The explicit actions are required only for admin.ib since the user account table on that database is a user table and not a system table.

- A plaintext password length of 32 bytes is supported in this release, up from 8 bytes in earlier versions of InterBase.

- An updated version of IBConsole is present in the kit. This version does not show the �Default� buttons in the database/server login screens.

- A batch script (changepassword.bat) is now provided in the /bin directory to update the SYSDBA account password post-install.

Getting Started

The DES-CRYPT password algorithm has been replaced with a modern cryptographic hash function that is more widely accepted by organizations in private industry and government. The design uses SHA-1, which generates a fixed length 160-bit hash.

1. Before starting, it is strongly recommended that you backup your old admin.ib from the current installation before installing the new InterBase. This allows you to restore it, if needed.

2. If you are upgrading to a newer version you may want to continue using your admin.ib from an earlier InterBase. If so, enable stronger password protection on that admin.ib by using the following SQL commands after IB has been installed on the server:

```
isql admin.ib -user SYSDBA -pass xxxxxxx
sql> ALTER DATABASE SET PASSWORD DIGEST 'SHA-1';
sql> CREATE DOMAIN PASSWORD_DIGEST AS VARCHAR(16)
CHARACTER SET ASCII;
sql> ALTER TABLE USERS ADD PASSWORD_DIGEST
PASSWORD_DIGEST;
sql> UPDATE USERS SET PASSWORD_DIGEST = 'DES-
CRYPT';
sql> COMMIT;
```

NOTE: If it is a completely new installation of InterBase you do not need to run the SQL commands.

The ALTER DATABASE command can only be run by the database owner or SYSDBA. This command modifies RDB$DATABASE.RDB$PASSWORD_DIGEST to the string value "SHA-1". This means that all new password hash generation for new or existing user accounts in the USERS table will use the SHA-1 hash function.

The password hash function can be reset to DES-CRYPT using the same DDL:

```
ALTER DATABASE SET PASSWORD DIGEST 'DES-CRYPT';
```

The admin database is now prepped so that new user accounts or modifying the password of existing accounts will generate SHA-1 password hashes against plaintext passwords up to an untruncated length of 32 significant bytes.

GSEC [add | modify], IBConsole, and the IB Services API support the SHA-1 password hash algorithm. Any of these tools can be used to maintain the passwords of server-wide user accounts. If an existing user account has had its password changed then that user must log in to the server using the new IB client library.

> Caution: There will be backward compatibility problems if the converted admin.ib database is backed up and restored by an older IB engine after the password hashes have been converted to SHA-1. Older IB engines will not understand the different password hashes and will cause unrecoverable login errors.

Larger Database Cache Settings for 64-bit InterBase

There is now a larger database cache setting for 64-bit InterBase. The limit for the 64-bit engine is 75 million pages, as compared to 750K pages for 32-bit engines.

EXECUTE STATEMENT for Stored Procedures

InterBase XE now provides support for the EXECUTE STATEMENT functionality.

User Interface/Usability

This feature enhances the InterBase Stored Procedure language. Once this is implemented, Stored Procedure developers can embed three variations of EXECUTE STATEMENT within their Stored Procedures. The variations depend on the number of rows returned from the EXECUTE STATEMENT command. The three cases are: No rows or data returned; One row of data returned; and Variable number of rows returned.

No Rows or Data Returned

EXECUTE STATEMENT <statement>

<statement> ::= a SQL statement returning no rows of data

Example:

```
CREATE PROCEDURE EXEC_STMT_NO_RET (proc_name varchar(20))
AS
DECLARE VARIABLE EMPNO INT;
DECLARE VARIABLE EXECSTMT;
BEGIN
  SELECT MAX(EMP_NO) from EMPLOYEE into EMPNO;

  EXECSTMT = 'EXECUTE PROCEDURE' || proc_name || '( ' || cast (EMPNO as varchar(10)) || ')';

  EXECUTE STATEMENT EXECSTMT;

END
```

One row of data returned

EXECUTE STATEMENT <select-statement> INTO :<var>[, :<var>..]

<select-statement>::= SQL statement returning one or no rows of data

<var> ::= valid procedure variable, the ":" is optional.

Example:

```
CREATE PROCEDURE EXEC_STMT_SINGLETON (TABLE_NAME VARCHAR(50))
AS
DECLARE VARIABLE MAXEMPNO

BEGIN
EXECUTE STATEMENT 'SELECT MAX(EMP_NO) FROM' || TABLE_NAME INTO :MAXEMPNO;


SUSPEND

END
```

Any number of data rows returned

FOR EXECUTE STATEMENT <select-statement> INTO :<var> [, :<var> ..]

DO <compound-statement>

<select-statement>::=   SQL statement returning one or no rows of data

<var>   ::= valid procedure variable, the ":" is optional.

Example:

```
CREATE PROCEDURE EXEC_STMT_ANY (TABLE_NAME VARCHAR(50), INT_FIELD INTEGER)
RETURNS
(INT_RETVAR INTEGER)
AS
DECLARE VARIABLE IFIELD INTEGER
BEGIN

FOR EXECUTE STATEMENT

'SELECT' || INT_FIELD || 'FROM' || TABLE_NAME INTO :IFIELD
  DO

    IF (IFIELD = 0) THEN
      INT_RETVAR=0;
    ELSE
      INT_RETVAR = INT_RETVAR+IFIELD;
  SUSPEND;

END
```

New Error Messages Added

The following table displays the new error messages added:

SQL Code	Error Number	Error Message
-204	335544850	EXECUTE STATEMENT could not prepare statement : *<string>*
-204	335544851	SQL statement invalid as it returns no records. SQL : *<string>*
-204	335544852	Parameter mis-match for the statement : *<string>*

-204	335544853	Could not execute statement : *<string>*
-204	335544854	EXECUTE STATEMENT fetch error
-204	335544855	EXECUTE STATEMENT in this form must return single row, not multiple rows.
-204	335544857	Sql statement not allowed in EXECUTE STATEMENT : *<string>*

Requirements/Constraints

Please note the following when using EXECUTE STATEMENT:

- The Statement is "prepared" every time it is executed which will affect the performance of the Stored Procedure.
- No checks are done on the statement when the procedure is created; dependency checks are not done when the procedure is created, also the checks for existence of tables or column names referred to in the execute statement are not performed. All these checks will be done at execute time and will result in errors if an error condition occurs.
- The feature can be used to perform DDL operations.
- All statements are executed based on the privileges of the user executing the Stored Procedure.
- SQL statements, "COMMIT", "COMMIT RETAIN", "ROLLBACK", "ROLLBACK RETAIN" and �CREATE DATABASE� are not supported with EXECUTE STATEMENT. These statements return the error code isc_exec_stmt_disallow.

Migration Issues

This feature is only available in InterBase XE and is not backward compatible.

Database Fast Sweep

Sweeping a database is a systematic way of removing outdated records. Periodic sweeping prevents a database from growing too large. In the past sweeping slowed system performance and users disabled the automatic database sweep function because of the impact on product operations.

InterBase databases periodically need to be swept. Otherwise the main memory allocated for each transaction's bitmap increases to the point where performance becomes unacceptable. The longer sweep takes to complete, the more main memory requirements increase for starting new transactions.

With the implementation of the fast sweep optimization in InterBase XE, the memory allocation issue has been mitigated. The user has the option to configure their databases for automatic sweep. In cases where large databases have large archival or infrequently modified tables, a database sweep will have minimal impact on the performance of running transactional operations.

User Interface/Usability

There is no new user interface or action required by the user to enable this functionality. Manual sweep initiated by the GFIX command line tool, IBConsole, or programmatically, as well as automatic sweep configuration on a database, use the fast sweep mechanism.

Requirements and Constraints

- Only ODS 15 and later databases can perform fast database sweeps.
- The effectiveness of a fast sweep is directly proportional to the fraction of database data pages that have been modified since the last sweep. If every data page has been changed, fast sweep is no faster than the former methodology. If very few pages are changed, fast sweep is nearly instantaneous. If half the pages were updated, fast sweep is then half the former sweep time.

Migration Issues

A database needs to be created or backed up and then restored to ODS15.

Table-Specific Blocking Factor

The term blocking factor is used to denote the number of records stored in a block. InterBase employs a single database-wide blocking factor that maximizes the number of rows that can be stored on a data page.

In InterBase XE, ODS 15 databases introduce table-specific blocking factors that optimize how many rows can be stored on a data page and minimizes the row number assigned to a row. In addition, the engine minimizes the size of a number of run-time data structures. This is especially important as tables storing a very large number of rows and accessed by a large number of database connections can cause excessive memory consumption.

User Benefit

Databases with tables containing a large number of rows or is expected to grow to a large number of rows benefit from table-specific blocking factors. It allows those tables to store more rows in the same record number space, while using less memory for run-time retrieval data structures.

User Interface/Usability

There is no user interface or any action required by the user to enable this functionality. The table-specific blocking factors are set automatically after a database restore.

The blocking factor values for individual tables can be observed in the system columns:

- RDB$RELATIONS.RDB$DATA_BLOCKING_FACTOR
- RDB$RELATIONS.RDB$BLOB_BLOCKING_FACTOR.

If a table does not have a table- specific blocking factor, this system column will query as NULL. Each time a database restore is performed, these blocking factors are retuned for the actual row data restored. Therefore, it is possible that these values will change over time as the database is modified and is backed up and restored.

Requirements and Constraints

- Only ODS 15 and later databases have table-specific blocking factors.
- Not all tables are capable of having a table-specific blocking factor.
- If a table has Blob columns and no indexes defined, then that table uses the database-wide blocking factor as its Blob blocking factor.
- System tables, views, external tables, and temporary tables do not use table-specific blocking factors.
- A table-specific blocking factor cannot be modified in-place once it has been set by database restore.

Migration Issues

If an ODS 15 database has been created and loaded with data, taking a backup and restoring will set the table-specific blocking factors.

Larger Index Key Segment Size

With the ODS 15 databases the maximum index key size limit is increased. Now larger column data can use this for both single-byte character sets and multi-byte (such as UTF8) columns.

Because InterBase XE supports UTF8 and multiple other multi-byte character sets, the limit has been increased. For example, a single-column key using 4-byte UTF8 character would calculate to 1020/4 = 254 UTF8 characters with a 4KB page size.

ODS 15 databases automatically allow index definitions where the underlying key size is now a factor of the database page size.

  - An index key can now be up to 4 bytes less than 1/4th the page size.
  - By default, InterBase databases are created with a 4KB page size. This can be overridden up to

16KB page size by the database developer.
  - The 4KB page size database would allow indexes that can accommodate 1020 bytes per key.
  - A 16KB page size can accommodate 4092 bytes per key and so on.

Requirements and Constraints

Databases created with engines enabled with this functionality cannot be moved back to older versions of InterBase.

Also a database restore to a smaller page size will fail if indexes with a large key size cannot fit within limit specified above.

No user interface or actions are required by the user to enable this functionality. Each time a database restore is performed, the indices are recreated.

Migration Issues

Only ODS 15 and later databases have support for larger index keys. If you want to use this facility, restore your database to ODS 15. Other indices that use a smaller size than 252 bytes continue to have the same on-disk storage without any penalty.

Migration Issues for InterBase XE

- Be sure to back-up all databases, including the security database, before uninstalling the previous version.

- We recommend that you upgrade your InterBase clients to 10.0.0.292 or above if you are targeting a SUSE 11 SP1 (or above) server. There are known problems with older clients waiting for events from a SUSE 11 SP1 server.

- Back-up the ibconfig file if it has been customized.

- This version creates new databases with ODS version 15.

- Please only provide 64-bit UDF libraries for a 64-bit edition, if any, of InterBase. Note that the 64-bit server will not be able to load any 32-bit UDF libraries that you may have. The UDF library that is provided (OOTB, ib_udf) is already built for 64-bit and is installed with the product.

- InterBase XE increased the number of records that can be stored in a table. The current count has 32-bit limitations. If you want to get a count of the number of record values beyond 32-bit, you need to take the following steps:

  - If a table does have more than 2G rows then the developer must code a CAST(COUNT(*) AS NUMERIC(18,0)) to get a 64-bit count.

- InterBase XE implemented stronger password protection. If you have older InterBase clients (locally or from remote systems) communicating with this installation, please note the following while upgrading your system to InterBase XE:

  - If you have an existing user authentication database (from previous InterBase versions), use that database file (copied to the InterBase XE folder) and authentication for local/remote clients will work correctly.

  - If you use the new admin.ib from InterBase XE, note that this database has "SHA-1" strength passwords (by default). If you want to continue using the (weaker) DES password algorithm (previous InterBase releases), use the ALTER DATABASE command referred to in the Stronger Password Protection topic.

  - If you want to use "stronger password length" (provided in InterBase XE), you will need to recreate your user accounts AND install the new InterBase clients in remote machines connecting to this InterBase XE server. This is because the "old" InterBase clients are not capable of computing the "SHA-1" passwords and thus pass in the "DES" strength passwords which in turn don't match what the InterBase XE server expects. This gives you the error "Your user name and password are not defined..."

- InterBase XE updated SSL parameter names. The old OTW client properties have been replaced by new OTW properties. Please refer to the OperationsGuide.pdf "Setting up OTW Encryption" and Table 5.2 located in Chapter 5.

- **Restoring database results in the error "unassigned code".**
  When restoring a database with InterBase XE, get the error unassigned code. With databases with a long lineage or databases backed up with InterBase 2009 and restored with InterBase XE, each case has different metadata security settings. So when selecting a system table (for example: RDB$RELATIONS) you get the error message: no permission for read/select access to table RDB$RELATIONS by user SYSDBA.

  This error occurs with databases which have these criteria:

  - Originally created with a version of InterBase prior to InterBase 6.5.
  - Backed up with a version of InterBase prior to InterBase 2009.
  - Readmeta.sql has not been previously applied.

  This behavior is exhibited due to stricter enforcement of meta data rights in InterBase XE during the restore of a database.

  Solution:
  To resolve this problem execute readmeta.sql against your database before you back it up. readmeta.sql may be found in \examples\security. You can execute readmeta.sql against your database using isql or IBConsole.

  Explanation:
  The problem occurs with databases that have a long lineage. The two cases are (1) a database has a restore history of IB6->IB7->IB2007->IB2009->IBXE; and (2) a database backed up and restored as IB2009->IBXE. Each case has different metadata security settings. The first instance never had metadata security because it originated from IB6. However with the second instance, it was created (not restored) by IB2009 with a full complement of security privileges for all system tables.

  With the first case, the database was backed up and restored and with each succeeding release, the new release would install privileges for the new system tables in that release (think RDB$USERS, RDB$ENCRYPTIONS, RDB$ROLES, etc.). But it couldn't alter the original system tables because it had no way of knowing if the database owner had already altered their security privileges. For example, a user might have revoked all privileges to RDB$TRIGGERS and RDB$PROCEDURES to conceal their trigger and stored procedure code.

  Also, in the first case, a SYSDBA may have run readmeta.sql years ago and refined the metadata from that baseline to a custom security profile. InterBase cannot override that customization by automatically resetting it after the XE restore. InterBase XE can't assume that every database it restores should unconditionally install the default metadata privileges because it doesn't know the history of individual databases.

  So it is recommended to run readmeta.sql, which sets the default or starting point for configuring it the way you want it. This advice is independent of whether you are migrating to XE.

  Example using isql:

  ```
  isql "path to database" -user sysdba -password masterkey -i readmeta.sql
  ```

  Executing readmeta.sql with IBConsole

  1. Connect to your database with IBConsole.
  2. Tools | Interactive SQL ...
  3. Query | Load Script, select readmeta.sql, click OK.

- **Error: No Permission for read/select access to table RDB$XXXX by user SYSDBA**
  Databases originally created before InterBase 6.5 may have the error: *no permission for read/select access to table RDB$XXXX by user SYSDBA with InterBase XE.*

Solution:

InterBase XE enforces tighter meta data security and this error may result from doing meta data operations on databases originally created with versions of InterBase prior to version 6.5. Meta data operations involve requesting information about system objects such as listing system objects or updating them and using the Performance Monitor.

To resolve this error you need to perform two similar operations. The first operation grants rights to system tables. To do so, execute readmeta.sql from the examples\security folder in your InterBase install directory.You can execute readmeta.sql against your database using isql or IBConsole.

Example using isql:

```
isql "path to database" -user sysdba -password masterkey -i readmeta.sql
```

Executing readmeta.sql with IBConsole

1. Connect to your database with IBConsole.
2. Tools | Interactive SQL ...
3. Query | Load Script, select readmeta.sql, click OK.

Second, you need to grant rights to system temorary tables if you are going to do performance monitoring. Due to potential security concerns, most installations will want to grant rights for system temporary tables only to sysdba and the database owner, which is what is presented below. If you wish for all users to be able to view system temporary tables, modify this example to GRANT TO PUBLIC. Some installations will want only specific users to have rights, in which case a more customized script may be needed.

To grant rights for system temporary tables, save the following as a text file, then execute it the same as readmeta.sql above.

```
 create procedure granttmp as
  declare variable stmt varchar(1024);
  declare variable ownername varchar(66);
  declare variable tablename varchar(66);
 begin
   select rdb$owner_name from rdb$relations where rdb$relation_name = 'RDB$RELATIONS'
   into :ownername;
 for select rdb$relation_name from rdb$relations where rdb$system_flag>0   and rdb$relation_name starts
with 'TMP$' into :tablename do
 begin
  stmt = 'grant all on ' || tablename || ' to sysdba';
  execute statement stmt;
  stmt = 'grant all on ' || tablename || ' to ' || ownername;
  execute statement stmt;
 end
end;
execute procedure granttmp;
drop procedure granttemp;
commit;
exit;
```

Known Issues

UNICODE Character Sets

The 16-bit UNICODE character sets UNICODE_LE and UNICODE_BE only work for Server character sets. These character sets cannot be used as a client character set. If your client needs full UNICODE character support, please use UTF8 instead of UNICODE_LE and UNICODE_BE for the client character set (aka LC_CSET). A client can use the UTF8 (or other native) client character set to connect with a UNICODE database.

InterBase XE supports no defined UNICODE collations in this release. The default collation is binary sort order for UNICODE.

Windows Error Reporting

Windows Error Reporting (WER) dialog pops up intermittently if and when an InterBase server crashes.

Resolution: We are working on fixing any crashes that we are aware of. In the meantime you can disable the Windows Error Reporting dialog from popping up by modifying the Windows Registry thus. Set the registry attribute HKEY_CURRENT_USERS\Software\Microsoft\Windows\Windows Error Reporting\DontShowUI value to 1 to disable. This is as per recommendation of MSDN article http://msdn.microsoft.com/en-us/library/bb513638(VS.85).aspx We may address this configurable option from within ibserver.exe in future builds by informing WER to disable this only for InterBase server binary.

Resolved Defects

The following are resolved defects for InterBase XE. Additional bug fixes are listed in the Release Notes.pdf.

Internal Defect #	External Defect # (QC)	Description
InterBase XE Update 2: April-2011, Bugs Fixed in 10.0.2 Version		
INTB-1054		XE client compatibility issue (with encryption and service manager).
INTB-1052	90926	ISQL crashes when "Show Procedures" command is issued if the database contains a stored procedure using UDFs.
INTB-1051	92451	This issue has been fixed so that the server no longer crashes. An error is also returned to disallow GROUP BY operation for Blob or array columns since that behavior is not well-defined.
INTB-1035		Unable to restore AppWave database backup.
INTB-1033		Count of a query involving a join and a left join returns zero.
INTB-1031		Inconsistent EVENT behavior.
INTB-1015		Server vulnerability with malformed packets reported by Zero Day Initiative of Tipping Point resolved. This vulnerability was processed through the Zero Day Initiative (ZDI), an initiative launched by TippingPoint, a division of 3Com. The ZDI is designed to reward security researchers for responsibly disclosing discovered vulnerabilities. Further information regarding the ZDI is available at: http://www.zerodayinitiative.com
INTB-1008		A parameterized query based on a union fails to return rows second time it is executed.
INTB-1000	90879	InterBase server crashes during validation phase, after database restore.
INTB-998	90582	OUTER JOIN query returns wrong result in InterBase XE (and works well in InterBase 2009).
INTB-985		Cannot establish a secondary connection for event processing for server behind a NAT router.
INTB-954	89559	Some queries with lots of ANDs and ORs cause server crash.
INTB-912		TCS: EX_NUM_ARRY_01 crashes ibserver on MacOS 10.6.
InterBase XE Update 1: December-2010, Bugs Fixed in 10.0.1 Version		
INTB-890		Sysdba has no metadata rights to databases created with InterBase 6.0 or earlier which have not had readmeta.sql applied. Raid 280556 was this same problem.
INTB-860		InterBase crashes with a specific database when a foreign key is added.
INTB-853		SELECT count() returns BIGINT - compatibility issue with existing driver/applications.
INTB-839	88501	Sweep while inserts are happening on a database with journaling causes the

		server to hang and a corrupted database.
INTB-824		Investigate InterBase XE 64-bit server crash when querying for statistics using InterBase service manager.
INTB-822		RAID 280541: Server crash on ODS 13.1 database with complicated metadata script.
INTB-818	88241	Internal gds software consistency check (can't find shared latch (300)) error with multiple connections.
InterBase XE November-2010, Bugs Fixed in 10.0.0.304 Version (UNIX builds)		
INTB-791		When JAVA application connect/disconnect/operate to OTW port, IBSever claims "SSL_READ: SSL_ERROR_ZERO_RETURN" most of the times. This is a harmless issue, and does not cause any functionality issues.
INTB-756		IBConsole: crashes when pressing the F5 key.
INTB-661	62444	RAID 260610: Examples are built on UNIX but do not run successfully.
INTB-673	57839	RAID 279453: LEFT JOIN operation returns wrong result.
INTB-546	74055	Parameter's incorrect descriptor lengths showing up inside UDFs.
InterBase XE: September-2010, Windows		
INTB-772	37729	RAID 279996: An error occurs when trying to create or change a password that begins with a question mark.
INTB-675		NullPointerException (instead of an sql error message) thrown when losing the network connection and trying to run a query.
INTB-671	79851	RAID 273825: getMetaData(). getPrecision() returning wrong value.
INTB-670	48152	RAID 251980: JOIN of 2 stored procedures give "arithmetic exception" error.
INTB-669	81786	RAID 278498: Increase request impure space size from the current 256K limit to a larger value.
INTB-659	62255	RAID 260612: QLI crashes as soon as "SHOW FUNCTIONS" command is given.
INTB-654	50375	RAID 260671: OnlineDump with multiple files.
INTB-653	55499	RAID 260629: Inserting from VARCHAR field in one table to BLOB in another table inserts additional junk in the target table.
INTB-650		no T privilege with grant option on table/view TableName.
INTB-597	2311 77931	RAID 146832: ORDER BY using an index on a DATE column gives wrong order.
INTB-547	365	RAID 279571: Non-SYSDBA db owner SELECT fails even with GRANTED rights on the table.
INTB-538		RAID 273806: Optimizer regression in JOINS since InterBase 2007 Service Pack changes.
INTB-523	77315	COALESCE and aggregate functions.
INTB-513		RAID 191536: The java.sql.DatabaseMetaData.getColumns() method returns incorrect data type for arrays and blobs.
INTB-493		IndexOutOfBoundsException thrown when calling PreparedStatement executeBatch().
INTB-402		Unexpected behavior of database linger.
INTB-355		Parameterized query params order in SQLDA/SQLVAR not the same as visual order of question marks.
INTB-319		Possible Optimizer regression with fixes in the InterBase 2007 SP3.
INTB-267		Create Database from JDBC application does not respect character set requested.
INTB-264		RAID 272543: RDB$FILTERS does not have a unique index on filter name leading to duplicate filters.
INTB-253		RAID 270024: InterBase cannot use 2GB+ database cache even if Large Address Aware flag was set.
INTB-236		RAID 271600: "Request Synchronization error" reported on 8-core system with heavy activity.
INTB-235		RAID 270959: Accessing ODS 10.x database leads to server crash.
INTB-151		RAID 269627: Unique superkey license files belonging to the same "addon" package are all not loaded by the licensing engine.
	3424	When you open the Properties form for a table or procedure and select the

		Permissions tab, the column headers rarely appear. If you select another tab and then come back to the Permissions tab, the column headers will be there.
	4989	I add a UDF with a param of type DATE, the UDF is created, but when I try to show the property of the UDF, IBConsole shows the Message: 'unsupported Datatype' The UDF cannot be dropped in IBConsole. The UDF works fine in ISQL, and can be dropped in ISQL This happend also with TIMESTAMP and TIME.
	6714	Create a table in an InterBase database and fill it with data. Then export the table to a Comma Seperated Value file using IBConsole and the export works fine. Check the exported file using MS Excel and it appears to be fine. Then empty the table. But when I try and import data from the .csv file that I exported the table to, I get the following message: Access Violation at address xxxxxxxx in module 'IBConsole.exe' Read of address 000000000. This is the case with any .CSV file that I try to import data from.
	63968	Database Restore Fails: "No current record for fetch operation".
	68637	DB Restore can't be done on IBconsole.
	74598	The name of the directory that contains the IBConsole.xml file is misspelled.
	75028	When trying to change the data type for a domain, an error appears.
	76342	Metadata names longer than 22 characters are truncated.
	76616	When trying to connect as a different user from the InteractiveSQL, the connection window opens but it doesn't connect as a different user and an error message appears.
	76622	When disconnecting from a database using the InteractiveSQL then trying to reconnect to the database, an error message appears.
	79418	When using the IBConsole for creating a journal, a journal archive, and to archive the database, the Archive Database option returns an error message (this fails for all types of connections.
	79420	When creating a journal, a journal archive and a database archive for a database, using Interactive SQL, the 'Archive Database' option doesn't refresh after disconnecting from the database and then reconnecting to the database.
	80580	IBConsole Open Table, Error.
	81324	Interactive SQL in IBConsole: truncated RDB$ names.
	82503	When creating a database then a generator using Interactive SQL, an error appears. You then have to use Task Manager to terminate the process.
	82706	When using a field that can be null in the Primary Key constraint definition, then removing it from there, an error appears.
	82776	When creating an external function with an entry point larger than 67 chars, an unclear error message appears.
	82829	When altering a view to add a second column, the table name is added incorrectly.
	82859	When altering a trigger by adding a local variable with the same name as another variable, then changing it back to the way it was, the changes cannot be made.
	82904	When choosing a value larger than 10 digits for the input type of a blob filter, an error message appears and the Blob Filter Editor window closes.
	83369	When choosing the 'New Connection' option from the Tools menu to connect to a different database then what is running a query, the IBConsole shuts down.
	83370	When trying to see the License Manager from the Tools menu of the IBConsole, an error message appears.
	83820	When trying to create a blob filter with a duplicate name, the Cancel button doesn�t work.
	83881	When creating a database with a name that already exists, the Cancel button doesn't work.
	84783	Text cannot be copied inside the name area of procedures and triggers, even if the focus is inside the name area; the text is added inside the body area.

	84934	When using the IBConsole to encrypt a database, then deleting the encryption from the system table of encryptions and changing the value of EUA, an AV error message appears.
	85054	When performing a database sweep using the IBConsole, the text of the confirmation message isn't correctly aligned.
	85104	For InterBase XE, License Manager from the Tools menu of the IBConsole doesn't open, and an AV error message is shown instead.
	85514	When using the IBConsole to run a couple of queries without committing, disconnecting the database creates an AV error message.
	86294	When connecting to a database as sysdso using IBConsole and trying to view data from a table that the user has select permission on, an error message appears.
	86737	EUA_USER and EUA_PASSWORD options must be added to Database Restore.
	86739	IBConsole should be using the new camel-cased and case-sensitive OTW parameter names (for IB OpenSSL clients).
	86749	SELECT COUNT(*) and SELECT AVG should return a SQL_INT64 type. If there is any code in IBConsole that uses 32-bit integer type local variable to receive this value, it should be changed to a 64-bit integer type.
	86924	IBConsole requires full path to backup file.
Documentation Updates: Sept-2010, Windows		
INTB-773		Language Reference Guide: Chapter 3: WHEN...DO. Updated GDSCODE errcode. Also updated Table 5.5 with correct GDSCODE usage.
INTB-769		Operation Guide: Appendix B: Table B.1 *InterBase Specifications.* Updated maximum size of external table file.
INTB-744		Developer�s Guide: Chapter 14 - *Applying Cached Updates with a Dataset Component Method.* Replaced old procedure example with a new IBTransaction1 example.
INTB-739		Language Reference Guide: Chapter 6 - Table 6.34. Added table RDB$USERS.
INTB-736		Embedded SQL Guide: Chapter 13- Topic: *SQL Datatype Macro Constants.* Updated information on sqlsubtype settings.
INTB-735		Developer�s Guide: Chapter 6 - Table 6.7. Updated integer information.
INTB-734		API Guide: Chapter 3 - *Environmental Variables Used by Interbase.* Added two new variables: IB_PROTOCOL and INTERBASE.
INTB-733		Language Reference Guide: Chapter 6 - Topic: *RDB$DATABASE.* Added missing fields to table 6.1.
INTB-731		Language Reference Guide: Chapter 2 - *ALTER TABLE, ALTER DATABASE, CREATE ENCRYPTION, CREATE TABLE.* Added note to each topic: For detailed information on encryption and decryption, see the topics �Encrypting Data�(page 13-9) and �Decrypting Data� (page 13-11) in the Data Definition Guide.
INTB-728	22384	Operation Guide: Chapter 4 - Topic *Using ibmgr to Start and Stop the Server.* Instructions for starting InterBase under the interbase user account have been updated.
INTB-726		API Guide: Chapter 5. Added new topic: *Use of Commit/Rollback with Multidatabase Transaction.*
INTB-725		Developers Guide: Chapter 6: *Explanation of dsc_flags.* Added DSC-systems information to Table 6.5.
INTB-723		Updated installation information. Desktop and ToGo setups are only available for Windows O.S.
INTB-722		Embedded SQL Guide: Chapter 6 - Topic *Using Comparison Operators in Expressions.* Added Table 6.4 listing comparison operators in expressions.
INTB-720	5253	RAID Issue 171696: Developer�s Guide: *Accessing InterClient extensions to the JDBC.* Added a suitable example for how to cast the JDBC driver object.
INTB-719		RAID Issue 259389: isc_dpb_archive_journals and

		isc_dpb_archive_database not listed in documentation.
INTB-718		RAID Issue: 217986: Operation Guide: Appendix A - Topic: New InterBase Keywords. Need the new Keywords introduced in InterBase 7.5 and above.
INTB-515		The Operations Guide PDF (page 5-10 to 5-12) details the various OTW parameters that can be used by InterBase native and JDBC applications. With InterBase XE, the older named parameters can be removed from the table and ONLY the new ones mentioned. Note, the new OTW parameters are case-sensitive and hence the documentation in the above pages need to be cleaned up with exact camel-casing of OTW parameters.
INTB-43		Small omission in Language Reference Guide. Field rdb$depended_on_type and rdb$dependent_type do not list types 11 and 14. 14 is External Functions. 11 is a generator.
		RAID Issue 112111. Operation Guide: Chapter 11 - Topic *Edit.* On Windows platforms, EDIT calls the text editor specified by the EDITOR environment variable. If this environment variable is not defined, then EDIT uses the Microsoft ~~mep~~ Notepad editor.
		RAID Issue 251988.Operation Guide: Chapter 9 - *Journaling Tips and Best Practices.* Updated CHECKPOINT LENGTH information.
		RAID Issue 218084. Language Reference Guide: Chapter 6. TMP$STATE field not documented in description for TMP$DATABASE.
		RAID Issue 263959. Data Definition Guide: Chapter 13. gbak examples on page 13-17 have invalid syntax.
	23253	RAID Issue 238178. Language Reference Guide. Chapter 2. The SQL example for the CASE statement won't work because it is truncated.
		RAID Issue 194123. Language Reference Guide. Chapter 2. Yearday for extract is listed as returning values 1-366. It should say 0-365. Hour and minute are similarly wrong. They are listed as going from 1-23 and 1-59. They should list 0-23 and 0-59.
		RAID Issue 256520. API Guide. Chapter 15. Signature for "isc_dsql_batch_execute()" incorrectly includes "isc_db_handle *db_handle". Signature for "isc_dsql_batch_execute()" incorrectly includes declaration using "ULONT Signature for "isc_dsql_batch_execute()" incorrectly includes declaration "int dialect" when actually the argument is implemented as an "unsigned short" in ibase.h
		RAID Issue 240455. API Guide. Chapter 15. isc_string_too_large error is generated when SQL is greater than 64K. 64K should be changed to 2 GB.
		RAID Issue 218053. Language Reference Guide. Chapter 2. ALTER DATABASE describes it as "Adds secondary files to the current database." This does not consider other new actions that can now be taken. Changed to "Changes the characteristics of a database".
		RAID Issue 120748. API Guide. Chapter 3. Remove ISC_DATABASE from Table 3.1. It is not implemented any longer.
		RAID Issue 106998.Operations Guide. Chapter 7. Error �OBJECT database_name IS IN USE� is incorrect.
	5742	RAID Issue 192587. Data Definition Guide. Chapter 8. Documentation says you can�t use UNION when creating a VIEW in DSQL. This hasn�t been true since IB 5.6.
	84435	RAID Issue 276427. Data Definition Guide. Chapter 14. UTF_8 Maximum character size is changed from 1 byte to VARCHAR(63).