Command-line utilities

General enhancements

Utilities support for database triggers

(**v. 2.1**) A new parameter was added to gbak, nbackup and isql to suppress database triggers from running. It is available only to the database owner and SYSDBA:

```
gbak -nodbtriggers
isql -nodbtriggers
nbackup -T
```

Password hiding

Alex Peshkov

Command-line utilities that take a -password parameter are vulnerable to password sniffing, especially when the utility is run from a script. As a step towards hardening against this on POSIX platforms, the [PASSWORD] argument now displays the process list as an asterisk (*), where previously it showed in clear.

Firebird services

New command-line utility fbsvcmgr

Alex Peshkov

(**v.2.1**) The new utility fbsvcmgr provides a command-line interface to the Services API, enabling access to any service that is implemented in Firebird.

Although there are numerous database administration tools around that surface the Services API through graphical interfaces, the new tool addresses the problem for admins needing to access remote Unix servers in broad networks through a text-only connection. Previously, meeting such a requirement needed a programmer.

Using fbsvcmgr

fbsvcmgr does not emulate the switches implemented in the traditional “g*” utilities. Rather, it is just a frontend through which the Services API functions and parameters can pass. Users therefore need to be familiar with the Services API as it stands currently. The API header file - ibase.h, in the ../include directory of your Firebird installation - should be regarded as the primary source of information about what is available, backed up by the InterBase® 6.0 beta API Guide.

Parameters

Specify the Services Manager

The first required parameter for a command line call is the Services Manager you want to connect to:

- For a local connection use the simple symbol service_mgr
- To attach to a remote host, use the format hostname:service_mgr

Specify subsequent service parameter blocks (SPBs)

Subsequent SPBs, with values if required, follow. Any SPB can be optionally prefixed with a single "-" symbol. For the long command lines that are typical for fbsvcmgr, use of the "-" improves the readability of the command line. Compare, for example, the following (each a single command line despite the line breaks printed here):

```
# fbsvcmgr service_mgr user sysdba password masterke
        action_db_stats dbname employee sts_hdr_pages
```

and

```
# fbsvcmgr service_mgr -user sysdba -password masterke
        -action_db_stats -dbname employee -sts_hdr_pages
```

SPB syntax

The SPB syntax that fbsvcmgr understands closely matches with what you would encounter in the ibase.h include file or the InterBase® 6.0 API documentation, except that a slightly abbreviated form is used to reduce typing and shorten the command lines a little. Here's how it works.

All SPB parameters have one of two forms: (1) isc_spb_VALUE or (2) isc_VALUE1_svc_VALUE2. For fbsvcmgr you just need to pick out the VALUE, VALUE1 or VALUE2 part(s) when you supply your parameter.

Accordingly, for (1) you would type simply VALUE, while for (2) you would type VALUE1_VALUE2. For example:

```
isc_spb_dbname => dbname
isc_action_svc_backup => action_backup
isc_spb_sec_username => sec_username
isc_info_svc_get_env_lock => info_get_env_lock
```

and so on.

Note: An exception is isc_spb_user_name: it can be specified as either user_name or simply user.

It is not realistic to attempt to describe all of the SPB parameters in the release notes. In the InterBase® 6.0 beta documentation it takes about 40 pages! The next section highlights some known differences between the operation of fbsvcmgr and what you might otherwise infer from the old beta documentation.

back to top of page

fbsvcmgr syntax specifics

“Do's and Don'ts”

With fbsvcmgr you can perform a single action - and get its results if applicable - or you can use it to retrieve multiple information items from the Services Manager. You cannot do both in a single command. For example,

```
# fbsvcmgr service_mgr -user sysdba -password masterke
    -action_display_user
```

will list all current users on the local Firebird server:

```
SYSDBA              Sql Server Administrator            0 0
QA_USER1                                                0 0
QA_USER2                                                0 0
QA_USER3                                                0 0
QA_USER4                                                0 0
QA_USER5                                                0 0
GUEST                                                   0 0
SHUT1                                                   0 0
SHUT2                                                   0 0
QATEST                                                  0 0
```

...and...

```
# fbsvcmgr service_mgr -user sysdba -password masterke
    -info_server_version -info_implementation
```

will report both the server version and its implementation:

```
Server version: LI-T2.1.0.15740 Firebird 2.1 Alpha 1
Server implementation: Firebird/linux AMD64
```

But an attempt to mix all of this in single command line:

```
# fbsvcmgr service_mgr -user sysdba -password masterke
    -action_display_user -info_server_version -info_implementation
```

raises an error:

```
Unknown switch "-info_server_version”"
```

Undocumented items

The function isc_spb_rpr_list_limbo_trans was omitted from the InterBase® 6 beta documentation. It is supported in fbsvcmgr.

Support for new Services API items in v.2.1

Two new items that were added to the Services API in Firebird 2.1 are supported by fbsvcmgr:

- isc_spb_trusted_auth (type it as trusted_auth) applies only to Windows. It forces Firebird to use Windows trusted authentication.
- isc_spb_dbname gives the ability to set a database name parameter (type as dbname) in all service actions related to accessing the security database from a remote client, equivalent to supplying the -database switch to the gsec utility.

Note: For gsec the -database switch is mostly used to specify a remote server you want to administer. In fbsvcmgr, the name of the server is already given in the first parameter (via the service_mgr symbol) so the [isc_spb_]dbname parameter is mostly unnecessary.

Documentation bugs

The format described for some parameters in the InterBase® 6 beta documentation are buggy. When in trouble, treat ibase.h as the primary source for the correct form.

Unsupported functions

- Everything to do with licensing was removed from the original InterBase® 6 open source code and is therefore not supported either in Firebird or by fbsvcmgr.
- The old config file view/modification functions have been unsupported since Firebird 1.5 and are not implemented by fbsvcmgr.

back to top of page

Backup service misbehaviour fixed

A. Peshkov

Feature request CORE-1232

(**v.2.1**) Some misbehaviour that could occur when the Services Manager was doing backup/restore operations and some parameter items were missing or in the wrong sequence. The problem still affects lower versions, including v.2.0.x, so care should be taken to specify all required switches and supply the database name and backup file spec in the correct order when using the -se[rvice_mgr] switch.

Disable non-SYSDBA access to privileged services

A. Peshkov

Feature request CORE-787

Non-SYSDBA access to parts of the Services API that return information about users and database paths has been disabled. A non-privileged user can retrieve information about itself, however.

back to top of page

Backup tools

Firebird 2 brings plenty of enhancements to backing up databases: a new utility for running on-line incremental backups and some improvements to gbak to avoid some of the traps that sometimes befall end users.

New on-line incremental backup

N. Samofatov

Fast, on-line, page-level incremental backup facilities have been implemented.

The backup engine comprises two parts:

- nbak, the engine support module.
- nbackup, the tool that does the actual backups.

NBak

The functional responsibilities of NBak are:

1. to redirect writes to difference files when asked (ALTER DATABASE BEGIN BACKUP statement)
2. to produce a GUID for the database snapshot and write it into the database header before the ALTER DATABASE BEGIN BACKUP statement returns
3. to merge differences into the database when asked (ALTER DATABASE END BACKUP statement)
4. to mark pages written by the engine with the current SCN [page scan] counter value for the database
5. to increment SCN on each change of backup state.

The backup state cycle is:

```
nbak_state_normal -> nbak_state_stalled -> nbak_state_merge ->
nbak_state_normal
```

- In *normal* state writes go directly to the main database files.
- In *stalled* state writes go to the difference file only and the main files are read-only.
- In *merge* state new pages are not allocated from difference files. Writes go to the main database files. Reads of mapped pages compare both page versions and return the version which is fresher, because we don't know if it is merged or not.

Note: This merge state logic has one quirky part. Both Microsoft and Linux define the contents of file growth as “undefined” i.e., garbage, and both zero-initialize them.

This is why we don't read mapped pages beyond the original end of the main database file and keep them current in the difference file until the end of a merge. This is almost half of NBak fetch and write logic, tested by using modified PIO on existing files containing garbage.

nbackup

The functional responsibilities of NBackup are

1. to provide a convenient way to issue ALTER DATABASE BEGIN/END BACKUP
2. to fix up the database after file system copy (physically change nbak_state_diff to nbak_state_normal in the database header)
3. to create and restore incremental backups.

Incremental backups are multi-level. That means if you do a Level 2 backup every day and a Level 3 backup every hour, each Level 3 backup contains all pages changed from the beginning of the day till the hour when the Level 3 backup is made.

Backing up

Creating incremental backups has the following algorithm:

1. Issue ALTER DATABASE BEGIN BACKUP to redirect writes to the difference file
2. Look up the SCN and GUID of the most recent backup at the previous level
3. Stream database pages having an SCN larger than was found at step 2 to the backup file.
4. Write the GUID of the previous-level backup to the header, to enable the consistency of the backup chain to be checked during restore.
5. Issue ALTER DATABASE END BACKUP
6. Add a record of this backup operation to RDB$BACKUP_HISTORY. Record current level, SCN, snapshot GUID and some miscellaneous stuff for user consumption.

Restoring

Restore is simple: we reconstruct the physical database image for the chain of backup files, checking that the backup_guid of each file matches prev_guid of the next one, then fix it up (change its state in the header to nbak_state_normal).

Usage

```
nbackup <options>
```

Valid Options

1. L <database> Lock database for file system copy
2. N <database> Unlock previously locked database
3. F <database> Fixup database after file system copy
4. B <level> <database> [<filename>] Create incremental backup

5. R <database> [<file0> [<file1>...]] Restore incremental backup
6. U <user> User name
7. P <password> Password

Note:

1. <database> may specify a database alias.
2. incremental backup of multi-file databases is not supported yet.
3. “stdout” may be used as a value of <filename> for the -B option.

Improvement in v.2.1.3

A. Peshkov

In the Firebird 2.5 beta, an improvement was done for POSIX versions to address a problem whereby the full backup tool of nBackup would hog I/O resources when backing up large databases, bringing production work to a standstill. This improvement was backported to v.2.1.3. Now, nBackup tries to read from the operating system cache before attempting to read from disk, thus reducing the I/O load substantially.

Note: The “cost” may be a 10 to 15 percent increase in the time taken to complete the full backup under high-load conditions.

Tracker reference CORE-2316.

User manual

P. Vinkenoog

A user manual for NBak/NBackup has been prepared. It can be downloaded from the documentation area at the Firebird website: https://www.firebirdsql.org/pdfmanual/ - the file name is

Firebird-nbackup.pdf

.

back to top of page

gbak backup/porting/restore utility

A number of enhancements have been added to gbak.

Changed behaviours, new switches

V. Khorsun

The new gbak switch

1. RECREATE_DATABASE [OVERWRITE]

is a separate switch designed to make it harder for the unsuspecting to overwrite a database accidentally, as could occur easily with the shortened form of the old switch:

1. R[EPLACE_DATABASE]

In summary:

- gbak -R (or gbak -r) now applies to the new -R[ECREATE_DATABASE] switch and will never overwrite an existing database if the O[VERWRITE] argument is absent.
- The short form of the old gbak -R[EPLACE_DATABASE] is now -REP[LACE_DATABASE]. This switch does not accept the O[VERWRITE] argument.
- The -REP[LACE_DATABASE] switch should be considered as deprecated, i.e. it will become unavailable in some future Firebird release.

This change means that, if you have any legacy batch or cron scripts that rely on gbak -r or gbak -R without modification, then the operation will except if the database exists.

If you want to retain the ability of your script to overwrite your database unconditionally, you will need to modify the command to use either the new switch with the OVERWRITE argument or the new short form for the old -REPLACE_DATABASE switch.

gbak made more version-friendly

C. Valderrama

(**v.2.1**) In its latest evolution, gbak can be used to restore a database on any version of Firebird.

Hide user name & password in shell

A. Peshkov

Feature request CORE-867

(**v.2.1**) gbak now changes param0 to prevent the user name and password from being displayed in ps axf.

gbak -V and the "counter" parameter

During Firebird 1 development, an optional numeric <counter> argument was added to the -V[erbose] switch of gbak for both backup and restore. It was intended to allow you to specify a number and get a running count of rows processed as the row counter passed each interval of that number of rows. It caused undesirable side-effects and was removed before Firebird 1.0 was ever released. So, although it never happened, it was documented as “implemented” in the release notes and other places.

back to top of page

ISQL query utility

Work on isql has involved a lot of bug-fixing and the introduction of a few new, useful features.

One trick to note is that CHAR and VARCHAR types defined in character set OCTETS (alias BINARY) now display in hex format. Currently, this feature cannot be toggled off.

New switches

The following command-line switches were added:

-b[ail] "Bail out"

D. Ivanov, C. Valderrama

Command line switch -b to instruct isql to bail out on error when used in non-interactive mode, returning an error code to the operating system.

When using scripts as input in the command line, it may be totally inappropriate to let isql continue executing a batch of commands after an error has happened. Therefore, the -b[ail] option will cause script execution to stop at the first error it detects. No further statements in the input script will be executed and isql will return an error code to the operating system.

- Most cases have been covered, but if you find some error that is not recognized by isql, you should inform the project, as this is a feature in progress.
- Currently there is no differentiation by error code - any non-zero return code should be interpreted as failure. Depending on other options (like -o, -m and -m2), isql will show the error message on screen or will send it to a file.

Some features

- Even if isql is executing nested scripts, it will cease all execution and will return to the operating system when it detects an error. Nested scripts happen when a script A is used as isql input but in turn A contains an INPUT command to load script B an so on. Isql doesn't check for direct or indirect recursion, thus if the programmer makes a mistake and script A loads itself or loads script B that in turn loads script A again, isql will run until it exhausts the memory or an error is returned from the database, at whose point -bail, if activated, will stop all activity.
- DML errors will be caught when being prepared or executed, depending on the type of error.
- In many cases, isql will return the line number of a DML statement that fails during execution of a script. (More about error line numbers ...)
- DDL errors will be caught when being prepared or executed by default, since isql uses AUTODDL ON by default. However, if AUTO DLL **[AUTODDL?]** is OFF, the server only complains when the script does an explicit COMMIT and this may involve several SQL statements.
- The feature can be enabled/disabled interactively or from a script by means of the command

```
SET BAIL [ON | OFF]
```

As is the case with other SET commands, simply using SET BAIL will toggle the state between activated and deactivated. Using SET will display the state of the switch among many others.

- Even if BAIL is activated, it doesn't mean it will change isql behavior. An additional requirement should be met: the session should be non-interactive. A non-interactive session happens when the user calls isql in batch mode, giving it a script as input.

Example

```
isql -b -i my_fb.sql -o results.log -m -m2
```

Tip: However, if the user loads isql interactively and later executes a script with the input command, this is considered an interactive session even though isql knows it is executing a script.

Example

```
isql
Use CONNECT or CREATE DATABASE to specify a database
SQL> set bail;
SQL> input my_fb.sql;
SQL> ^Z
```

Whatever contents the script has, it will be executed completely, errors and all, even if the BAIL option is enabled.

back to top of page

-m2 to output stats and plans

C. Valderrama

This is a command-line option -m2 to send the statistics and plans to the same output file as the other output (via the -o[utput] switch).

When the user specifies that the output should be sent to a file, two possibilities have existed for years: either

- at the command line, the switch -o followed by a file name is used
- the command OUTput followed by a file name is used, either in a batch session or in the interactive isql shell. (In either case, simply passing the command OUTput is enough to have the output returned to the console). However, although error messages are shown in the console, they are not output to the file.

The -m command line switch was added, to meld (mix) the error messages with the normal output to wherever the output was being redirected.

This left still another case: statistics about operations (SET STATs command) and SQL plans as the server returns them. SET PLAN and SET PLANONLY commands have been treated as diagnostic messages and, as such, were always sent to the console.

What the -m2 command line switch does is to ensure that stats and plans information go to the same file the output has been redirected to.

Note: Neither -m nor -m2 has an interactive counterpart through a SET command. They are for use only as command-line isql options.

-r2 to pass a case-sensitive role name

C. Valderrama

The sole objective of this parameter is to specify a case-sensitive role name.

- The default switch for this parameter is -r. Roles provided in the command line are uppercased.
- With -r2, the role is passed to the engine exactly as typed in the command line.

back to top of page

New commands and enhancements

The following commands have been added or enhanced.

Ctrl-C to cancel query output

M. Kubecek, A. dos Santos Fernandes

Feature request CORE-704

(**v. 2.1**) Output from a SELECT in an interactive isql session can now be stopped using Ctrl-C. Note, this merely stops fetching rows from the buffer, it does not cancel the query.

Extension of isql SHOW SYSTEM command

A. dos Santos Fernandes

Feature request CORE-978

(**v. 2.1**) The SHOW <object_type> command is meant to show user objects of that type. The SHOW SYSTEM command is meant to show system objects but, until now, it only showed system tables. Now it lists the predefined system UDFs incorporated into FB 2.

It may be enhanced to list system views if we create some of them in the future.

SHOW COLLATIONS command

A. dos Santos Fernandes

(**v. 2.1**) Lists all the character set/collation pairs declared in the database.

SET HEAD[ing] toggle

C. Valderrama

Some people consider it useful to be able to do a SELECT inside isql and have the output sent to a file, for additional processing later, especially if the number of columns makes isql display impracticable. However, isql by default prints column headers and, in this scenario, they are a nuisance.

Therefore, printing the column headers - previously a fixed feature - can now be enabled/disabled interactively or from a script by means of the

```
SET HEADing [ON | OFF]
```

command in the isql shell. As is the case with other SET commands, simply using SET HEAD will toggle the state between activated and deactivated.

Note: There is no command-line option to toggle headings off.

Using SET will display the state of SET HEAD, along with other switches that can be toggled on/off in the isql shell.

back to top of page

SET SQLDA_DISPLAY ON/OFF

A. dos Santos Fernandes

This SQLDA_DISPLAY command shows the input SQLDA parameters of INSERTs, UPDATEs and DELETEs. It was previously available only in DEBUG builds and has now been promoted to the public builds. It shows the information for raw SQLVARs. Each SQLVAR represents a field in the XSQLDA, the main structure used in the FB API to talk to clients transferring data into and out of the server.

Note: The state of this option is not included in the output when you type SET; in isql to see the current settings of most options.

SET TRANSACTION enhanced

C. Valderrama

The SET TRANSACTION statement has been enhanced so that, now, all TPB options are supported:

- NO AUTO UNDO
- IGNORE LIMBO
- LOCK TIMEOUT <number>

Example

```
SET TRANSACTION WAIT SNAPSHOT NO AUTO UNDO LOCK TIMEOUT 10
```

See also the document doc/sql.extensions/README.set_transaction.txt.

SHOW DATABASE now returns ODS version number

C. Valderrama

ODS (On-Disk Structure) version is now returned in the SHOW DATABASE command.

back to top of page

Ability to show the line number where an error happened in a script

C. Valderrama

In previous versions, the only reasonable way to know where a script had caused an error was using the switch -e for echoing commands, -o to send the output to a file and -m to merge the error output to the same file. This way, you could observe the commands isql executed and the errors if they exist. The script continued executing to the end. The server only gives a line number related to the single command (statement) that it's executing, for some DSQL failures. For other errors, you only know the statement caused problems.

With the addition of -b for bail as described under New switches, the user is given the power to tell isql to stop executing scripts when an error happens, but you still need to echo the commands to the output file to discover which statement caused the failure.

Now, the ability to signal the script-related line number of a failure enables the user to go to the script directly and find the offending statement. When the server provides line and column information, you will be told the exact line of DML in the script that caused the problem. When the server only indicates a failure, you will be told the starting line of the statement that caused the failure, related to the whole script.

This feature works even if there are nested scripts, namely, if script SA includes script SB and SB causes a failure, the line number is related to SB. When SB is read completely, isql continues executing SA and then isql continues counting lines related to SA, since each file gets a separate line counter. A script SA includes SB when SA uses the INPUT command to load SB.

Lines are counted according to what the underlying IO layer considers separate lines. For ports using EDITLINE, a line is what readline() provides in a single call. The line length limit of 32767 bytes remains unchanged.

Enhanced command-line help

M. Kubecek

When unknown parameters are used, isql now shows all of the command-line parameters and their explanations instead of just a simple list of allowed switches.

```
opt/firebird/bin] isql -?

Unknown switch: ?
usage: isql [options] [<database>]
```

-a(all)	Extract metadata incl. legacy non-SQL tables.
-b(ail)	Bail on errors (set bail on).
-c(ache) <num>	Number of cache buffers.
-ch(arset) <charset>	Connection charset (set names).
-d(atabase) <database>	Database name to put in script creation.
-e(cho)	Echo commands (set echo on).
-ex(tract)	Extract metadata.
-i(nput) <file>	Input file (set input).
-m(erge)	Merge standard error.
-m2	Merge diagnostic.
-n(oautocommit)	No autocommit DDL (set autoddl off).
-now(arnings)	Do not show warnings.
-o(utput) <file>	Output file (set output).
-pag(elength) <size>	Page length.
-p(assword) <password>	Connection password.
-q(uiet)	Do not show the message Use CONNECT....
-r(ole) <role>	Role name.
-r2 <role>	Role (uses quoted identifier).
-sqldialect <dialect>	SQL dialect (set sql dialect).
-t(erminator) <term>	Command terminator (set term).
-u(ser) <user>	User name.
-x	Extract metadata.
-z	Show program and server version.

back to top of page

gsec authentication manager

Changes to the gsec utility include:

gsec return code

C. Valderrama

gsec now returns an error code when used as a non-interactive utility. Zero indicates success; any other code indicates failure.

gfix server utility

Changes to the gfix utility include:

New shutdown states (modes)

N. Samofatov, D. Yemanov

The options for gfix -shut[down] have been extended to include two extra states or modes to govern the shutdown.

New syntax pattern

```
gfix <command> [<state>] [<options>]

<command> ::= {-shut | -online}
<state> ::= {normal | multi | single | full}
<options> ::= {-force <timeout> | -tran | -attach}
```

- *normal* state = online database
- *multi* state = multi-user shutdown mode (the legacy one, unlimited attachments of SYSDBA/owner are allowed)
- *single* state = single-user shutdown (only one attachment is allowed, used by the restore process)
- *full* state = full/exclusive shutdown (no attachments are allowed)

Note: Multi is the default state for -shut, *normal* is the default state for -online.

The modes can be switched sequentially:

```
normal <-> multi <-> single <-> full
```

Examples

```
gfix -shut single -force 0
gfix -shut full -force 0
gfix -online single
gfix -online
```

You cannot use -shut to bring a database one level “more online” and you cannot use -online to make a database more protected (an error will be thrown).

For example, these sequence-pairs are prohibited:

```
   gfix -shut single -force 0
   gfix -shut multi -force 0
::
   gfix -online
   gfix -online full
 ::
   gfix -shut -force 0
   gfix -online single
```

Timeout: As before, the timeout is in seconds. In the case of the -attach and -tran timeouts, the timeout determines how long the engine will wait for any attached clients to complete their work and log off. The shutdown request should return the SQLCode -902 message shutfail (ISC code 335544557), Database shutdown unsuccessful if there are still active attachments when the timeout expires.

However, there is a known issue with the implementation of the new modes. A regression occurred, whereby the said message is returned but the engine does not revert the database to the online state, as it should. It affects all versions of Firebird up to and including v.2.0.5 and v.2.1.3, and all v.2.5 alphas, betas and release candidates.

back to top of page

Builds and installs

Parameter for Instance name added to instsvc.exe

D. Yemanov

Feature request CORE-673

(**v.2.1**) instsvc.exe now supports multi-instance installations.

Revised Win32 installer docs

P. Reeves

(**v.2.1**) The documentation for command-line setup on Windows has been revised. See doc/install_windows_manually.txt.

Help on command-line switches

Feature request CORE-548

(**v.2.1**) On-line help is now available on the switches for command-line setup on Windows.

Gentoo/FreeBSD detection during install

A. Peshkov

Feature request CORE-1047

Gentoo or FreeBSD is now detected during configuration, making it more likely that the binary install will work “out of the box” on these platforms.