The Test Suite

MWA Software's Firebird Pascal API package (fbintf) comes with its own test suite. This is a console mode program that is designed to provide a comprehensive function and regression test for the package. It comprises eighteen individual tests, testing 115 features. The spreadsheet “doc/testsuite.ods” lists out the test features and identifies which tests cover which feature. A test coverage indication is also given identifying how many tests include use of a given feature.

The test suite may be found in the “fbintf/testsuite” directory.

Running the Test Suite

A script is provided to build and run the test suite. The script also compares the results against a reference log and shows the differences between the reference and the output of the test. It can then be assessed as to whether there are issues with the results, or whether the tests have passed.

Under Linux: the script is called “runtest.sh”.

Under Windows (FPC): the script is called “runtest.bat”

Under Windows (Delphi): the script is called “rundelphitest.bat”

The script should be run from a console window with the current directory set to the fbintf/testsuite directory.

FPC (min version 3.0.0) or Delphi (min version 2010) must be installed, as must, as a minimum, the Firebird Client library. It is recommended that the testsuite is run on a system on which the full Firebird distribution has been installed from a distribution package downloaded from http://www.firebirdsql.org. The Firebird server installation must include the example “employee” database (installed by default by the Firebird installation package) and the server must be running.

The test suite does not require Lazarus to be installed.

After any necessary customisations (see below), the test suite is run by executing the test suite script.

- Under Linux: enter “./runtest.sh”
- Under Windows (FPC): enter “runtest.bat”
- Undgit remote add origin https://github.com/MWASoftware/ibx4lazarus.git
- er Windows (Delphi): enter “rundelphitest.bat”

Note that under the Delphi (free) Berlin Edition, the command line compiler has been disabled and hence the testsuite has to be compiled under the IDE before running the test script.

Customising the Test Suite

Linux

The following environment variables are defined at the start the script:

```
TESTOUTDIR=/tmp/fbintf-testsuite
USERNAME=SYSDBA
PASSWORD=masterkey
EMPLOYEEDB=localhost:employee
NEWDBNAME=localhost:$TESTOUTDIR/testsuite1.fdb
NEWDBNAME2=localhost:$TESTOUTDIR/testsuite2.fdb
BAKFILE=$TESTOUTDIR/testsuite.gbk
```

The Username and password should be changed if the defaults are not suitable for the server.

The databases are assumed to be located on the “localhost”. If this is not possible then localhost should be replaced with the domain name of a suitable server.

The BAKFILE must always be on the local system. It does not have to be in the same directory as the new (temporary) databases created and removed by the test suite.

Windows (FPC)

The following environment variables are defined at the start the script:

```
set FPCDIR=C:\lazarus\fpc\3.0.0
set FPCBIN=%FPCDIR%\bin\i386-win32
set TESTOUTDIR=%TEMP%\fbintf-testsuite
set USERNAME=SYSDBA
set PASSWORD=masterkey
set EMPLOYEEDB=localhost:employee
set NEWDBNAME=localhost:%TESTOUTDIR%\testsuite1.fdb
set NEWDBNAME2=localhost:%TESTOUTDIR%\testsuite2.fdb
set BAKFILE=%TESTOUTDIR%\testsuite.gbk
```

The FPCDIR should be set to the correct installation directory for FPC.

FPCBIN is currently set for win32. It should be changed from “i386-win32” to “x86_64-win64” for use with the 64-bit compiler.

The Username and password should be changed if the defaults are not suitable for the server.

The databases are assumed to be located on the “localhost”. If this is not possible then localhost should be replaced with the domain name of a suitable server.

The BAKFILE must always be on the local system. It does not have to be in the same directory as the new (temporary) databases created and removed by the test suite.

Windows (Delphi)

The following environment variables are defined at the start the script:

```
set DELPHIBIN=C:\Program Files\Embarcadero\RAD Studio\7.0\bin
set DIFF=C:\Program Files\GnuWin32\bin\diff.exe
set TESTOUTDIR=%TEMP%\fbintf-testsuite
set USERNAME=SYSDBA
set PASSWORD=masterkey
set EMPLOYEEDB=localhost:employee
set NEWDBNAME=localhost:%TESTOUTDIR%\testsuite1.fdb
set NEWDBNAME2=localhost:%TESTOUTDIR%\testsuite2.fdb
set BAKFILE=%TESTOUTDIR%\testsuite.gbk
```

DELPHIBIN is currently set for the default installation for RAD Studio 2010 and should be adjusted for the currently installed version. The DIFF environment variable needs to point to a working installation of the GNU Diff utility. This may be downloaded from http://gnuwin32.sourceforge.net/packages/diffutils.htm

If FPC is also installed then the diff utility provided by FPC is used.

The other environment variables are as for FPC.

Interpreting the Results

The reference log was created on a Linux Mint 64 bit system running Firebird 2/3/4 with a version of the example employee database untouched since installation. When the test suite is run on a similar system, there will be very little difference between the reference log and the test output. However, other platforms will give different results.

Linux and Firebird 2.5

The following output was seen when running the test suite on a Linux system with Firebird 2.5 installed:

```
Starting Testsuite

Running Test 1: Create and Drop a Database
Running Test 2: Open the employee database and run a query
Running Test 3: ad hoc queries
Running Test 4: Update, Insert and Delete Queries
Running Test 5: Update Returning and Activity Check
Running Test 6: Blob Handling
Running Test 7: Create and read back an Array
Running Test 8: Create and read back an Array with 2 dimensions
Running Test 9: Database Information tests
Running Test 10: Event Handling
Running Test 11: Services API
Running Test 12: Character Sets
Running Test 13: Character Sets over two databases
Running Test 14: Non select procedures
Running Test 15: Blob Handling and BPBs
Running Test 16: Error handling
Heap dump by heaptrc unit
23089 memory blocks allocated : 2127320/2196056
23089 memory blocks freed     : 2127320/2196056
0 unfreed memory blocks : 0
```

```
True heap size : 131072
True free heap : 131072
Comparing results with reference log

5c5
< Client API Version = 3.768
---
> Client API Version = 2.5
8c8,10
< Create Database fails (as expected): Can not access lock files directory
/tmp/firebird/
---
> Create Database fails (as expected): I/O error during "open O_CREAT"
operation for file ""
> Error while trying to create file
> No such file or directory
19,20c21,22
< Database ID = 4 FB = /tmp/fbintf-testsuite/testsuite1.fdb SN = mint173
< ODS major = 12 minor = 0
---
> Database ID = 4 FB = /tmp/fbintf-testsuite/testsuite1.fdb SN = zeus
> ODS major = 11 minor = 2
23c25
< RDB$SECURITY_CLASS = SQL$363
---
> RDB$SECURITY_CLASS =
25d26
< RDB$LINGER =
166,168c167
< Select Expression
<     -> First N Records
<         -> Table "EMPLOYEE" Full Scan
---
> PLAN (EMPLOYEE NATURAL)
1576,1577c1575,1576
< Database ID = 4 FB = /opt/firebird/examples/empbuild/employee.fdb SN =
mint173
< Pages =307
---
> Database ID = 4 FB =
/usr/share/doc/firebird2.5-common-doc/examples/empbuild/employee.fdb SN =
zeus
> Pages =272
1581,1585c1580,1584
< ODS minor = 0
< ODS major = 12
< Page Size = 8192
< Version = 1: LI-V6.3.0.32483 Firebird 3.0
< Server Memory = 18984656
---
> ODS minor = 2
> ODS major = 11
> Page Size = 4096
> Version = 1: LI-V6.3.2.26540 Firebird 2.5
> Server Memory = 9137296
1587c1586
< Max Memory  = 19023160
---
> Max Memory  = 9274808
1591,1593c1590,1592
< Fetches  = 491
< Writes  = 4
< Reads  = 51
---
> Fetches  = 503
> Writes  = 2
> Reads  = 47
1606,1607d1604
< Table ID = 4
< Count = 28
1609,1611c1606
< Count = 5
< Table ID = 9
< Count = 1
---
> Count = 11
1615c1610
< Count = 15
---
> Count = 24
1617c1612
< Count = 1
---
> Count = 4
1619c1614
< Count = 1
---
> Count = 4
1623c1618
< Count = 94
---
> Count = 88
1625c1620
< Count = 2
---
> Count = 1
1653,1654c1648,1649
< Server Version = LI-V3.0.0.32483 Firebird 3.0
< Implementation = Firebird/Linux/AMD/Intel/x64
---
> Server Version = LI-V2.5.2.26540 Firebird 2.5
> Implementation = Firebird/linux AMD64
```

```
1657,1658c1652,1653
< Message File = /opt/firebird/
< Security File = /opt/firebird/security3.fdb
---
> Message File = /usr/share/firebird/2.5/
> Security File = /var/lib/firebird/2.5/system/security2.fdb
1665a1661,1696
> First Name = Sql
> Middle Name = Server
> Last Name = Administrator
> User ID = 0
> Group ID = 0
> User Name = TESTER
> First Name =
> Middle Name =
> Last Name =
> User ID = 0
> Group ID = 0
1678c1709
< Capabilities = 6
---
> Capabilities = 774
1903,1904c1934
< Failed to locate host machine.
< The specified name was not found in the hosts file or Domain Name Services
---
> Failed to locate host machine
```

The first lines of the output report the tests as they are being run followed by a report from the heaptrc unit indicating that, as expected, there are no memory leaks. The difference between the reference log and the test output is then shown.

Note that the full test output is saved in “testsuite/testout.log” and the diff in “testsuite/diff.log”:

- The first difference is the API version, 2.5 for this run, while the reference has 3.768. This is entirely expected as the test system uses Firebird 2.5, while the reference system used Firebird 3.
- A test for the create database error condition is next shown. The problem here is that the error messages differ between the Firebird versions. Again, no problem.
- The Database ID, which includes the system name is naturally different.
- The ODS version has changed between the two Firebird Versions hence the next difference.
- The RDB$SECURITY_CLASS and RDB$LINGER differences come from a dump of the RDB$DATABASE file and again differ correctly between Firebird versions.
- The next difference comes from a report of a query plan and illustrates a very different way of reporting the query plan for Firebird 3 compared with Firebird 2.5.
- The next set of differences start at line 1581 and come from Test 9. This is a database

information report. There are many differences here including the location of the employee database, ODS version and table access statistics. All of this should be expected.

- At line 1653 onwards, the differences come from Test 11 (Services API). Again all these differences should be expected from the different server versions. The differences in the list of user names reported will also be dependent on your system's security database and which users are configured.
- Finally, at line 1903, we see a more explanatory error message from Firebird 3, but otherwise no difference.

Windows and Firebird 2.5

The next example test results come from running script under Windows 7 with Firebird 2.5.

```
Starting Testsuite

Running Test 1: Create and Drop a Database
Running Test 2: Open the employee database and run a query
Running Test 3: ad hoc queries
Running Test 4: Update, Insert and Delete Queries
Running Test 5: Update Returning and Activity Check
Running Test 6: Blob Handling
Running Test 7: Create and read back an Array
Running Test 8: Create and read back an Array with 2 dimensions
Running Test 9: Database Information tests
Running Test 10: Event Handling
Running Test 11: Services API
Running Test 12: Character Sets
Running Test 13: Character Sets over two databases
Running Test 14: Non select procedures
Running Test 15: Blob Handling and BPBs
Running Test 16: Error handling
Heap dump by heaptrc unit
23251 memory blocks allocated : 1932734/1992640
23251 memory blocks freed     : 1932734/1992640
0 unfreed memory blocks : 0
True heap size : 393216
True free heap : 392576
Should be : 393216
Comparing results with reference log

5c5
< Client API Version = 3.768
---
> Client API Version = 2.5
8c8,9
< Create Database fails (as expected): Can not access lock files directory
/tmp/
firebird/
---
> Create Database fails (as expected): I/O error during "open" operation for
```

```
fil
e ""
> database or file exists
19,20c20,21
< Database ID = 4 FB = /tmp/fbintf-testsuite/testsuite1.fdb SN = mint173
< ODS major = 12 minor = 0
---
> Database ID = 4 FB = C:\USERS\A USER\APPDATA\LOCAL\TEMP\FBINTF-TESTSUITE\
TESTS
UITE1.FDB SN = WIN7ZEUS
> ODS major = 11 minor = 2
23c24
< RDB$SECURITY_CLASS = SQL$363
---
> RDB$SECURITY_CLASS =
25d25
< RDB$LINGER =
166,168c166
< Select Expression
<     -> First N Records
<         -> Table "EMPLOYEE" Full Scan
---
> PLAN (EMPLOYEE NATURAL)
767c765
< TITLE = Blob Test ┬⌐Γé¼ (Charset Id = 4 Codepage = 65001)
---
> TITLE = Blob Test ⌐Ç (Charset Id = 4 Codepage = 65001)
790c788
< TITLE = Blob Test ┬⌐Γé¼ (Charset Id = 4 Codepage = 65001)
---
> TITLE = Blob Test ⌐Ç (Charset Id = 4 Codepage = 65001)
809c807
< The pangs of despised love, the lawΓÇÖs delay,
---
> The pangs of despised love, the lawÆs delay,
865c863
< TITLE = Blob Test ┬⌐Γé¼ (Charset Id = 4 Codepage = 65001)
---
> TITLE = Blob Test ⌐Ç (Charset Id = 4 Codepage = 65001)
884c882
< The pangs of despised love, the lawΓÇÖs delay,
---
> The pangs of despised love, the lawÆs delay,
906c904
< TITLE = Blob Test ┬⌐Γé¼ (Charset Id = 4 Codepage = 65001)
---
> TITLE = Blob Test ⌐Ç (Charset Id = 4 Codepage = 65001)
925c923
< The pangs of despised love, the lawΓÇÖs delay,
---
> The pangs of despised love, the lawÆs delay,
1180c1178
< TITLE = Blob Test ┬⌐Γé¼ (Charset Id = 4 Codepage = 65001)
---
> TITLE = Blob Test ⌐Ç (Charset Id = 4 Codepage = 65001)
1203c1201
< TITLE = Blob Test ┬⌐Γé¼ (Charset Id = 4 Codepage = 65001)
---
> TITLE = Blob Test ⌐Ç (Charset Id = 4 Codepage = 65001)
1222c1220
< The pangs of despised love, the lawΓÇÖs delay,
---
> The pangs of despised love, the lawÆs delay,
1278c1276
< TITLE = Blob Test ┬⌐Γé¼ (Charset Id = 4 Codepage = 65001)
---
> TITLE = Blob Test ⌐Ç (Charset Id = 4 Codepage = 65001)
1297c1295
< The pangs of despised love, the lawΓÇÖs delay,
---
> The pangs of despised love, the lawÆs delay,
1319c1317
< TITLE = Blob Test ┬⌐Γé¼ (Charset Id = 4 Codepage = 65001)
---
> TITLE = Blob Test ⌐Ç (Charset Id = 4 Codepage = 65001)
1338c1336
< The pangs of despised love, the lawΓÇÖs delay,
---
> The pangs of despised love, the lawÆs delay,
1469c1467
< TITLE = Blob Test ┬⌐Γé¼ (Charset Id = 4 Codepage = 65001)
---
> TITLE = Blob Test ⌐Ç (Charset Id = 4 Codepage = 65001)
1471c1469
< NOTES = ├ëcoute moi (Charset Id = 4 Codepage = 65001)
---
> NOTES = ╔coute moi (Charset Id = 4 Codepage = 65001)
1484c1482
< TITLE = Blob Test ┬⌐Γé¼ (Charset Id = 4 Codepage = 65001)
---
> TITLE = Blob Test ⌐Ç (Charset Id = 4 Codepage = 65001)
1486c1484
< NOTES = ├ëcoute moi (Charset Id = 4 Codepage = 65001)
---
> NOTES = ╔coute moi (Charset Id = 4 Codepage = 65001)
1494c1492
< TITLE = Blob Test ┬⌐Γé¼ (Charset Id = 4 Codepage = 65001)
---
> TITLE = Blob Test ⌐Ç (Charset Id = 4 Codepage = 65001)
1496c1494
< NOTES = ├ëcoute moi (Charset Id = 4 Codepage = 65001)
---
```

```
> NOTES = ╓coute moi (Charset Id = 4 Codepage = 65001)
1576,1577c1574,1575
< Database ID = 4 FB = /opt/firebird/examples/empbuild/employee.fdb SN =
mint173

< Pages =307
---
> Database ID = 4 FB = C:\PROGRAM FILES\FIREBIRD\FIREBIRD_2_5\EXAMPLES\
EMPBUILD\
EMPLOYEE.FDB SN = WIN7ZEUS
> Pages =292
1579c1577
< Implementation = 11,7,0,3,66,1,66,
---
> Implementation = 11,7,0,3,68,1,68,
1581,1585c1579,1583
< ODS minor = 0
< ODS major = 12
< Page Size = 8192
< Version = 1: LI-V6.3.0.32483 Firebird 3.0
< Server Memory = 18984656
---
> ODS minor = 2
> ODS major = 11
> Page Size = 4096
> Version = 1: WI-V6.3.5.26952 Firebird 2.5
> Server Memory = 9271232
1587c1585
< Max Memory  = 19023160
---
> Max Memory  = 9287112
1591,1593c1589,1591
< Fetches  = 491
< Writes  = 4
< Reads  = 51
---
> Fetches  = 503
> Writes  = 2
> Reads  = 47
1606,1607d1603
< Table ID = 4
< Count = 28
1609,1611c1605
< Count = 5
< Table ID = 9
< Count = 1
---
> Count = 11
1615c1609
< Count = 15
---
```

```
> Count = 24
1617c1611
< Count = 1
---
> Count = 4
1619c1613
< Count = 1
---
> Count = 4
1623c1617
< Count = 94
---
> Count = 88
1625c1619
< Count = 2
---
> Count = 1
1653,1654c1647,1648
< Server Version = LI-V3.0.0.32483 Firebird 3.0
< Implementation = Firebird/Linux/AMD/Intel/x64
---
> Server Version = WI-V2.5.5.26952 Firebird 2.5
> Implementation = Firebird/x86-64/Windows NT
1656,1658c1650,1652
< Lock Directory = /tmp/firebird/
< Message File = /opt/firebird/
< Security File = /opt/firebird/security3.fdb
---
> Lock Directory = C:\ProgramData\firebird\
> Message File = C:\Program Files\Firebird\Firebird_2_5\
> Security File = C:\Program Files\Firebird\Firebird_2_5\security2.fdb
1666,1668c1660,1662
< First Name =
< Middle Name =
< Last Name =
---
> First Name = Sql
> Middle Name = Server
> Last Name = Administrator
1678c1672
< Capabilities = 6
---
> Capabilities = 1542
1699a1694,1711
>
>
>
>
>
>
>
```

```
>
>
>
>
>
>
>
>
>
>
>
1872c1884
< ├ëcoute moi
---
> ╓coute moi
1878c1890
< Error Handled: I/O error during "open" operation for file "Malformed Name"
---
> Error Handled: I/O error during "CreateFile (open)" operation for file
"Malfor
med Name"
1880c1892
< No such file or directory
---
> The system cannot find the file specified.
1903,1904c1915
< Failed to locate host machine.
< The specified name was not found in the hosts file or Domain Name Services
---
> Failed to locate host machine
```

Here we have a similar set of differences to the Linux results except for additional differences reported at line 767 etc. e.g.

```
< TITLE = Blob Test ┬⌐Γé¼ (Charset Id = 4 Codepage = 65001)
---
> TITLE = Blob Test ⌐Ç (Charset Id = 4 Codepage = 65001)

```

These are artefacts of the Windows console character set. The reference log is in UTF8. However, the output of the test is written using “writeln” statements. These always use the Windows console character set and the log comparison spots problems when non-ASCII characters are output. The definitive test for character handling is Test 12 and no differences are found here – which is the important result.

If you look at the testout.log file, the formatting of these lines should be correct (i.e. “Blob Test ©€” is the actual text) confirming that the issue is with console mode output.

Testing Using Mutiple Firebird Servers

The script dotest.sh is made available as an example of how testing with multiple Firebird servers and multiple versions of the FPC compiler can be scripted.

The general idea is that the directory /opt/firebirdVersions contains one or more subdirectories each of which contain a copy of the Firebird installation. These subdirectories are given names that reflect the release e.g. 4.0.4 and the files within them are just a copy of the files distributed in a Firebird installation binary (but with no attempt to run the installation script. The main role of the dotest.sh script is to set up the FIREBIRD environment variable and to invoke "runtest.sh". It can also select the appropriate FPC compiler.

Each Firebird installation is configured by:

1. Editing firebird.conf to set the server port to some obscure but unique port no.

2. The employee database is decompressed #!/bin/sh

```
cd /opt
for FB in `ls -d firebirdVersions/*`; do
  if [ -z "`ps ax|grep /opt/$FB/bin/fb_smp_server|grep -v grep`" ] && [ -z
"`ps ax|grep /opt/$FB/bin/firebird|grep -v grep`" ]; then
    echo "starting $FB/bin/fbguard"
    export FIREBIRD=/opt/$FB
    export LD_LIBRARY_PATH=/opt/$FB/lib
    $FB/bin/fbguard&
  else
    echo "$FB not started"
  fi
done
```

if necessary and otherwise made available for use.

3. The SYSDBA password is enabled and set to (e.g.) masterkey by

```
export FIREBIRD=/opt/firebirdVersions/4.0.4
export LD_LIBRARY_PATH=$FIREBIRD/lib
$FIREBIRD/bin/isql -user SYSDBA employee
CREATE USER SYSDBA PASSWORD 'masterkey';
exit;
```

The servers can then be started by a simple script such as:

```
#!/bin/sh
cd /opt
for FB in `ls -d firebirdVersions/*`; do
  if [ -z "`ps ax|grep /opt/$FB/bin/fb_smp_server|grep -v grep`" ] && [ -z
"`ps ax|grep /opt/$FB/bin/firebird|grep -v grep`" ]; then
    echo "starting $FB/bin/fbguard"
    export FIREBIRD=/opt/$FB
    export LD_LIBRARY_PATH=/opt/$FB/lib
    $FB/bin/fbguard&
  else
    echo "$FB not started"
  fi
done
```

and dotest.sh should then be able to do its magic.

Conclusion

The test suite should complete with no memory leaks and with log file differences that can all be explained by considering the local environment. A successful run of the test suite gives confidence that the package is working to specification on your system. Further tests will be added to the test suite when and if any more bugs are reported and fixed, thereby further increasing confidence in the correct functioning of the package.