Tracking down crashes on Linux

(Also works also for other POSIX systems)

By Alex Peshkov, May 2008.

One of the most efficient ways to track a crash of any program on Linux is to get a core dump of it at the moment of the crash. Using it you can easily create a stack backtrace, which in many cases makes it possible for developers to solve your problem. Or (if you are using Firebird debug binaries) the whole core file can be sent to the developers for deeper analysis.

This document describes the process of how to get a core dump of the Firebird server on Linux and then analyzing it with gdb, using Firebird's debuginfo package. Depending on OS version (and other issues) the file with the core dump may be named simply core or core.PID, where PID is a number of the process which of which the core was dumped. Always use the actual name of the core file rather than core in examples below.

Getting a core dump

Firebird 2.0.4 or higher

You should set parameter BugcheckAbort=1 in the firebird.conf file. Do not forget to restart superserver after you modify the firebird.conf. Resulting core dump will be found in the /tmp directory.

Firebird 2.0.3 or earlier

The first thing you should consider if you are using an old version of Firebird is to upgrade it. The Firebird development team regularly fixes various bugs in the code base, therefore it's quite possible that your problem has already solved. But if you do need to get core dump of an old Firebird version. You will need to perform the following actions:

To get dump of process's core you should satisfy the following requirements:

+ Set ulimit's RLIMIT_CORE value big enough to to allow it to handle the dump of the Firebird server (it's even simpler and better to make it unlimited). This is what the command ulimit -c unlimited normally does.

Make the working directory of Firebird server writable for the server process - the core file is always created in current directory.

Note: Just typing:

```
# ulimit -c unlimited
```

at the shell prompt does not affect this setting for any daemon running on your system – this includes Firebird. So here is how you get a core file for both Superserver and Classic.

Superserver

To make RLIMIT_CORE to unlimited, you should edit /etc/init.d/firebird file. This is the startup script for Firebird, however it's exact location may differ for on various versions of Linux, but in most cases it usually is in the /etc/init.d directory. See your OS documentation for exact location of the startup script. Script contents may also differ much depending on Linux version. In all files you can find a line starting with

```
export FIREBIRD=
```

or

```
FIREBIRD=
```

(followed by FbRoot directory, usually opt/firebird). Add right after this command:

```
ulimit -c unlimited
```

and restart Firebird.

Superserver always sets it's working directory to /tmp, which is normally writable to everyone. You should find the core dump in /tmp after next crash.

Classic server

The Classic server is normally started using xinetd, but there is no easy way to separately configure ulimit for services, started in this manner. The only way to is to set ulimit for xinetd itself. To do this, you should modify startup the script for xinetd. These scripts differ for the various Linux versions, but in most cases adding

```
ulimit -c unlimited
```

near the beginning of the script should work. Do not forget to restart xinetd after modifying the startup script.

Classic server has / (root) as it's working directory, which is normally writable only to root user. Therefore if you are not running Firebird as root, you will not get core dump.

To solve this problem do one of the following:

use the restoreRootRunUser.sh script to temporarily run firebird as root (to revert back to the Firebird runuser, user changeRunUser.sh). There is a slim possibility that some crashes may not be reproducible when running as root.

make / writable for user firebird. This is probably the best solution for systems with ACLs support.

Analyzing a core dump with gdb

To analyze the core dump you should download the debuginfo package from the main Firebird download pages (for non normal Firebird project supplied packages or OS's where the debuginfo packages are not available you may need to contact the package maintainer for the debuginfo package). Debuginfo package is not an install – just untar it in the root of your file system. After if you may use:

```
 # gdb /opt/firebird/bin/.debug/fb_inet_server.debug core
```

to analyze core dump of classic server or

```
 # gdb /opt/firebird/bin/.debug/fbserver.debug core
```

to analyze core dump of superserver.

Firebird's debuginfo has one problem – all binaries in it are linked as runtime, not debug. This is not critical for superserver, but can cause problems for classic. To analyze classics's core properly, do:

```
 # mv /opt/firebird/lib/libfbembed.so.1.5.5
/opt/firebird/lib/libfbembed.so.1.5.5.runtime
```

```
 # mv /opt/firebird/lib/.debug/libfbembed.so.1.5.5.debug
/opt/firebird/lib/libfbembed.so.1.5.5
```

After analyzing the core, rename the files back.

The sample is provided for Firebird 1.5.5, but if you substitute your version number, it's correct for any Firebird version.