
Subject: LibX11 error & lock-up in debug mode with GridCtrl

Posted by [dolik.rce](#) on Mon, 16 Jan 2012 21:22:09 GMT

[View Forum Message](#) <> [Reply to Message](#)

Hi everyone,

I've been experiencing very weird problem with CtrlCore and GridCtrl in debug mode on Linux. Everything works just fine in optimal or when I don't use GridCtrl package. As soon as I just add GridCtrl to my package, without even including the GridCtrl.h, things go wrong. The application throws this on stderr:Quote:The program 'guitest' received an X Window System error.

This probably reflects a bug in the program.

The error was 'RenderBadPicture (invalid Picture parameter)'.

(Details: serial 118 error_code 160 request_code 148 minor_code 7)

(Note to programmers: normally, X errors are reported asynchronously; that is, you will receive the error a while after causing it.

To debug your program, run it with the --sync command line option to change this behavior. You can then get a meaningful backtrace from your debugger if you break on the gdk_x_error() function.)

Then it runs forever, taking 100% of CPU resources. When I run it in debugger and stop it at random, I always get this backtrace:Quote:#0 0xb771172d in _XReply () from /usr/lib/libX11.so.6

#1 0xb770cec6 in XSync () from /usr/lib/libX11.so.6

#2 0xb76ecf96 in XCloseDisplay () from /usr/lib/libX11.so.6

#3 0x0822f4b3 in Upp::s__sF26_46_fn () at

/home/h/upp-production/uppsrc/CtrlCore/DrawX11.cpp:50

#4 0xb71ce0e1 in __run_exit_handlers () from /lib/libc.so.6

#5 0xb71ce16d in exit () from /lib/libc.so.6

#6 0xb7b13500 in ?? () from /usr/lib/libgdk-x11-2.0.so.0

#7 0xb7713ca3 in _XError () from /usr/lib/libX11.so.6

#8 0xb771093d in ?? () from /usr/lib/libX11.so.6

#9 0xb7710997 in ?? () from /usr/lib/libX11.so.6

#10 0xb7711850 in _XReply () from /usr/lib/libX11.so.6

#11 0xb76f4e0e in XGetGeometry () from /usr/lib/libX11.so.6

#12 0xb7b14a2c in gdk_pixmap_foreign_new_for_display () from /usr/lib/libgdk-x11-2.0.so.0

#13 0xb7b14a88 in gdk_pixmap_foreign_new () from /usr/lib/libgdk-x11-2.0.so.0

#14 0x081952e3 in Upp::GetGTK (widget=0x86ef8e0, state=0, shadow=2, detail=0x83e7d1d "radiobutton", type=18, cx=17, cy=17, rect=...)

at /home/h/upp-production/uppsrc/CtrlLib/ChGtk0.cpp:146

#15 0x081979bb in Upp::GtkIml (uii=0, w=0x86ef8e0, shadow=2, state=0, detail=0x83e7d1d "radiobutton", type=18, cx=17, cy=17, rect=...)

at /home/h/upp-production/uppsrc/CtrlLib/ChGtk0.cpp:417

#16 0x08197aec in Upp::GtkIml (uii=0, w=0x86ef8e0, shadow=2, detail=0x83e7d1d "radiobutton", type=18, cx=17, cy=17, rect=...)

at /home/h/upp-production/uppsrc/CtrlLib/ChGtk0.cpp:423

#17 0x0819d1fa in Upp::ChHostSkin () at /home/h/upp-production/uppsrc/CtrlLib/ChGtk.cpp:151

#18 0x0820ea2f in Upp::Ctrl::ReSkin () at /home/h/upp-production/uppsrc/CtrlCore/Ctrl.cpp:916

#19 0x0828dcd6 in Upp::Font::SetStdFont (font=...) at

/home/h/upp-production/uppsrc/Draw/Font.cpp:111

#20 0x082306c1 in Upp::InitX11Draw (display=0x8674c00) at

```
/home/h/upp-production/uppsrc/CtrlCore/DrawX11.cpp:220
#21 0x082308a9 in Upp::InitX11Draw (dispname=0x0) at
/home/h/upp-production/uppsrc/CtrlCore/DrawX11.cpp:242
#22 0x0824cb2a in Upp::Ctrl::InitX11 (display=0x0) at
/home/h/upp-production/uppsrc/CtrlCore/X11App.cpp:400
#23 0x0804e9dc in main (argc=1, argv=0xbffff9b4, envp=0xbffff9bc) at
/home/h/MyApps/guitest/main.cpp:13
```

So as far as I can tell, some bad X11 error happens, U++ tries to cleanup and exit and the EXITBLOCK code from DrawX11.cpp results in infinite loop over or inside `_XReply()`. Since it happens even when GridCtrl.h is not included, I suspected some INIT/EXITBLOCKs in GridCtrl, but I haven't found any. Is there something else that registers itself even if the code is not called directly? Or perhaps something in CtrlCore that reacts to GridCtrl presence?

Oh, and I forgot to mention that this happens with both GCC 4.6.2 and Clang 3.0. It also happens with NOGTK flag, but then the error is different:Quote:X Error of failed request:

```
RenderBadPicture (invalid Picture parameter)
Major opcode of failed request: 148 (RENDER)
Minor opcode of failed request: 7 (RenderFreePicture)
Picture id in failed request: 0x17
Serial number of failed request: 10
Current serial number in output stream: 91
```

Can anyone reproduce this weird behavior? Or is it just something rotten in my system?

Best regards,
Honza
