
Subject: bug in toolbar

Posted by [bushman](#) on Wed, 12 Nov 2014 03:47:57 GMT

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

Test case below causes app to crash by moving mouse pointer quickly back and forth over ToolBar buttons for a while. Crash seems to occur more easily when doing it right after button tip popup.

```
#include <CtrlLib/CtrlLib.h>

using namespace Upp;

class App : public TopWindow
{
public:
    typedef App CLASSNAME;
    ToolBar bar;

    void DoNothing() {}

    void HandleBar(Bar& bar) {
        bar.Add("tip1", CtrlImg::copy(), THISBACK(DoNothing));
        bar.Add("tip2", CtrlImg::paste(), THISBACK(DoNothing));
    }

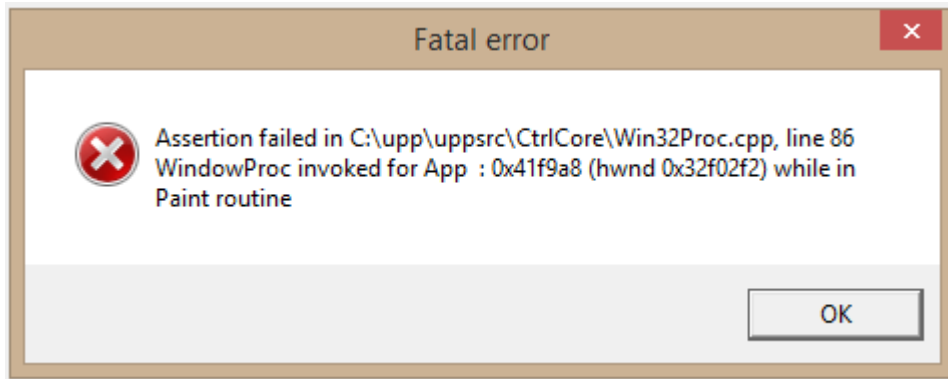
    App()
    {
        Add(bar.HSizePos().TopPos(0, 30));
        bar.Set(THISBACK(HandleBar));
    }
};

GUI_APP_MAIN
{
    App().Run();
}
```

Help appreciated, tks in advance.

File Attachments

1) [MsgBox.PNG](#), downloaded 1142 times



Subject: Re: bug in toolbar

Posted by [mirek](#) on Wed, 12 Nov 2014 10:34:02 GMT

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

Spent 5 minutes waving the mouse to reproduce with no success.

Perhaps I am not doing it right - any tips?

Mirek

P.S.: Trying with recent trunk, right?

Subject: Re: bug in toolbar

Posted by [bushman](#) on Wed, 12 Nov 2014 23:30:18 GMT

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

Thanks Mirek, for your prompt response.

Version updated to 7884, problem still persists.

In my case, it happens quite variably, from just a few jiggles all the way up to some 40 or 50 back and forth jerks, but consistently always within a time span below 1 minute. BTW, I tried to record a clip to share, but never got a crash when doing so, which makes me wonder if the clip recording software has something to do with it. I mean, maybe the video frame capturing interrupt delays gives Windows a chance to resync its Paint operations somehow, just by slowing down the whole system.

The only tip I could give you is that you should first hover the mouse pointer over a button and wait for the tip popup to show up before vigorously moving mouse pointer quickly from one button to the other. Anyway, couldn't it be the case that Windows invokes WindowProc for tooltip popup while still in Button paint routine?

tk!

Subject: Re: bug in toolbar
Posted by [mirek](#) on Thu, 13 Nov 2014 19:45:38 GMT
[View Forum Message](#) <> [Reply to Message](#)

Unlikely.

Anyway, perhaps you could run in the debugger, send me the stack trace?

Mirek

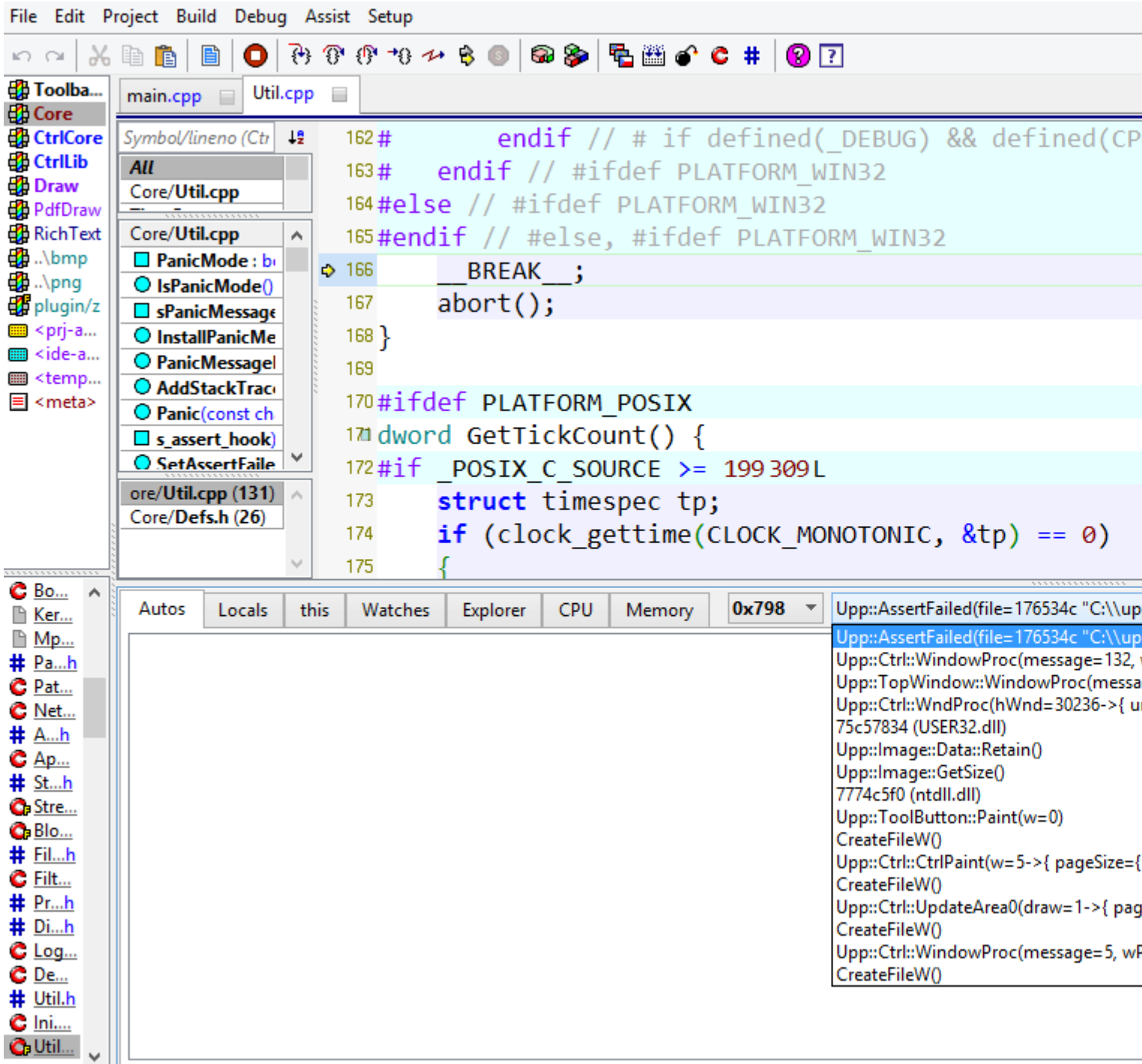
Subject: Re: bug in toolbar
Posted by [bushman](#) on Thu, 13 Nov 2014 20:33:00 GMT
[View Forum Message](#) <> [Reply to Message](#)

I hope it helps...

tk!

File Attachments

1) [toolbarbug.png](#), downloaded 1000 times



Subject: Re: bug in toolbar

Posted by [bushman](#) on Fri, 14 Nov 2014 13:36:19 GMT

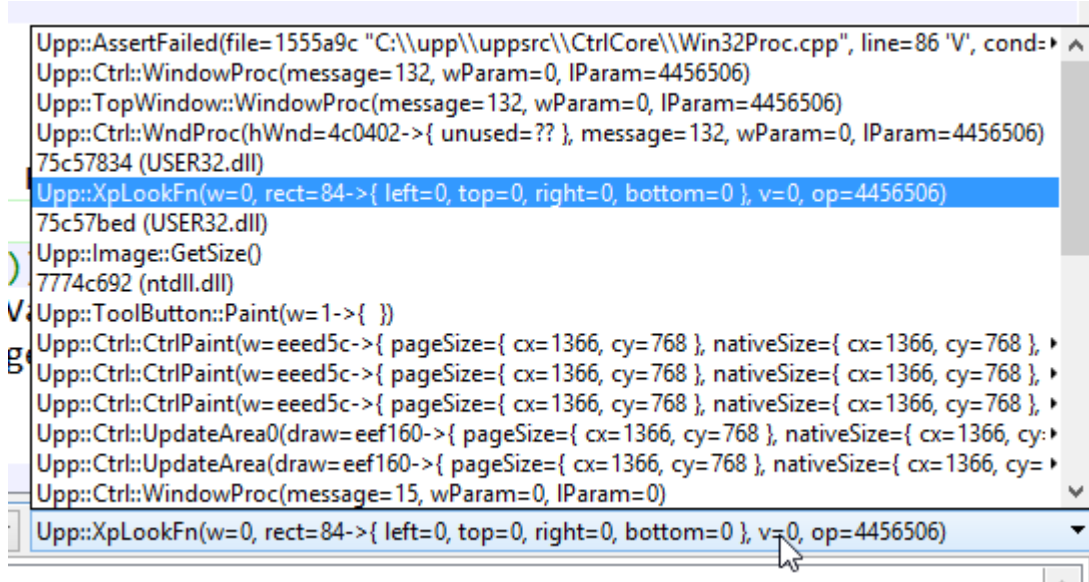
[View Forum Message](#) <> [Reply to Message](#)

Here is another stack trace:

thanks!

File Attachments

1) [toolbarbug1.png](#), downloaded 1023 times



```
Upp::AssertFailed(file=1555a9c "C:\\uppl\\uppsrc\\CtrlCore\\Win32Proc.cpp", line=86 'V', cond=▶
Upp::Ctrl::WindowProc(message=132, wParam=0, lParam=4456506)
Upp::TopWindow::WindowProc(message=132, wParam=0, lParam=4456506)
Upp::Ctrl::WndProc(hWnd=4c0402->{ unused=?? }, message=132, wParam=0, lParam=4456506)
75c57834 (USER32.dll)
Upp::XpLookFn(w=0, rect=84->{ left=0, top=0, right=0, bottom=0 }, v=0, op=4456506)
75c57bed (USER32.dll)
Upp::Image::GetSize()
7774c692 (ntdll.dll)
Upp::ToolButton::Paint(w=1->{ })
Upp::Ctrl::CtrlPaint(w=eeced5c->{ pageSize={ cx=1366, cy=768 }, nativeSize={ cx=1366, cy=768 }, ▶
Upp::Ctrl::CtrlPaint(w=eeced5c->{ pageSize={ cx=1366, cy=768 }, nativeSize={ cx=1366, cy=768 }, ▶
Upp::Ctrl::CtrlPaint(w=eeced5c->{ pageSize={ cx=1366, cy=768 }, nativeSize={ cx=1366, cy=768 }, ▶
Upp::Ctrl::UpdateArea0(draw=eef160->{ pageSize={ cx=1366, cy=768 }, nativeSize={ cx=1366, cy=▶
Upp::Ctrl::UpdateArea(draw=eef160->{ pageSize={ cx=1366, cy=768 }, nativeSize={ cx=1366, cy=▶
Upp::Ctrl::WindowProc(message=15, wParam=0, lParam=0)
Upp::XpLookFn(w=0, rect=84->{ left=0, top=0, right=0, bottom=0 }, v=0, op=4456506)
```

Subject: Re: bug in toolbar

Posted by [mirek](#) on Fri, 14 Nov 2014 17:10:38 GMT

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

First wild guess is that this looks like problem with Image cache.

Can you please try to deactivate paintonly:

```
void SysImageRealized(const Image& img)
{
    return;
```

(and alternatively same for SysImageReleased)

and report if it is still crashing?

Mirek

Subject: Re: bug in toolbar

Posted by [bushman](#) on Fri, 14 Nov 2014 20:14:34 GMT

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

Hi, Mirek,

mirek wrote on Fri, 14 November 2014 12:10

Can you please try to deactivate paintonly:

```
void SysImageRealized(const Image& img)
{
    return;
}
```

(and alternatively same for SysImageReleased)

and report if it is still crashing?

Did that on both calls, didn't work, still crashing...

tk.

Subject: Re: bug in toolbar
Posted by [mirek](#) on Sat, 15 Nov 2014 08:31:13 GMT
[View Forum Message](#) <> [Reply to Message](#)

That's bad. OK, let us add some LOGs:

```
Size Image::GetSize() const          { DDUMP(data); return data ? data->buffer.GetSize() :
Size(0, 0); }
```

```
void ToolButton::Paint
    DLOG("PAINT");
```

Also, in debug menu there is 'Copy backtrace', which should make posting backtraces easier...

Mirek

Subject: Re: bug in toolbar
Posted by [bushman](#) on Sat, 15 Nov 2014 15:20:41 GMT
[View Forum Message](#) <> [Reply to Message](#)

You got it:

```
Upp::AssertFailed(file=1545aac "C:\\upp\\uppsrc\\CtrlCore\\Win32Pro".., line=86 'V',
cond=2f20058 "WindowProc invoked for App : 0"..)
Upp::Ctrl::WindowProc(message=132, wParam=0, lParam=11075744)
Upp::TopWindow::WindowProc(message=132, wParam=0, lParam=11075744)
Upp::Ctrl::WndProc(hWnd=6040c->{ unused=?? }, message=132, wParam=0, lParam=11075744)
```

```

75c57834 (USER32.dll)
Upp::String0::Free()
Upp::AString<Upp::String0>::~~AString<Upp::String0>()
Upp::String0::Free()
Upp::ToolButton::Paint(w=7ef0c0->{ })
Upp::Ctrl::CtrlPaint(w=7ef0c0->{ pageSize={ cx=1366, cy=768 }, nativeSize={ cx=1366, cy=768 },
nativeDpi={ cx=96 '', cy=96 '' }, palette=0, color16=0, is_mono=0, native=0, actual_offset_bak={
x=8319220, y=19146303 }, cloff={ vector={ vector=1130190, items=6, alloc=8 } }, drawingclip={
left=0, top=0, right=22, bottom=22 }, .. }, clip=7eec44->{ left=-22, top=0, right=22, bottom=22 })
Upp::Ctrl::CtrlPaint(w=7ef0c0->{ pageSize={ cx=1366, cy=768 }, nativeSize={ cx=1366, cy=768 },
nativeDpi={ cx=96 '', cy=96 '' }, palette=0, color16=0, is_mono=0, native=0, actual_offset_bak={
x=8319220, y=19146303 }, cloff={ vector={ vector=1130190, items=6, alloc=8 } }, drawingclip={
left=0, top=0, right=22, bottom=22 }, .. }, clip=7eedf0->{ left=3, top=2, right=47 '/', bottom=24 })
Upp::Ctrl::CtrlPaint(w=7ef0c0->{ pageSize={ cx=1366, cy=768 }, nativeSize={ cx=1366, cy=768 },
nativeDpi={ cx=96 '', cy=96 '' }, palette=0, color16=0, is_mono=0, native=0, actual_offset_bak={
x=8319220, y=19146303 }, cloff={ vector={ vector=1130190, items=6, alloc=8 } }, drawingclip={
left=0, top=0, right=22, bottom=22 }, .. }, clip=7ef18c->{ left=3, top=2, right=47 '/', bottom=24 })
Upp::Ctrl::UpdateArea0(draw=7ef4c4->{ pageSize={ cx=1366, cy=768 }, nativeSize={ cx=1366,
cy=768 }, nativeDpi={ cx=96 '', cy=96 '' }, palette=0, color16=0, is_mono=0, native=0,
actual_offset_bak={ x=0, y=13338048 }, cloff={ vector={ vector=2f90bc0, items=1, alloc=4 } },
drawingclip={ left=-1073741823, top=-1073741823, right=1073741823, .. }, .. }, clip=7ef18c->{
left=3, top=2, right=47 '/', bottom=24 }, backpaint=2)
Upp::Ctrl::UpdateArea(draw=7ef4c4->{ pageSize={ cx=1366, cy=768 }, nativeSize={ cx=1366,
cy=768 }, nativeDpi={ cx=96 '', cy=96 '' }, palette=0, color16=0, is_mono=0, native=0,
actual_offset_bak={ x=0, y=13338048 }, cloff={ vector={ vector=2f90bc0, items=1, alloc=4 } },
drawingclip={ left=-1073741823, top=-1073741823, right=1073741823, .. }, .. }, clip=7ef3b8->{
left=3, top=2, right=47 '/', bottom=24 })
Upp::Ctrl::WindowProc(message=15, wParam=0, lParam=0)
Upp::TopWindow::WindowProc(message=15, wParam=0, lParam=0)
Upp::Ctrl::WndProc(hWnd=6040c->{ unused=?? }, message=15, wParam=0, lParam=0)
75c57834 (USER32.dll)
75c5930f (USER32.dll)
75c57bed (USER32.dll)
Upp::Ctrl::sProcessMSG(msg=7ef7c4->{ hwnd=17dd6b0, message=0, wParam=15, lParam=0,
time=0, pt={ x=18227988, y=2004061488 } })
Upp::Ctrl::ProcessEvent(quit=7ef8b7->0)
Upp::Ctrl::ProcessEvents(quit=7ef8b7->0)
Upp::Ctrl::EventLoop(ctrl=7ef968->{ parent=0, top=bd0040, exitcode=NULL -2147483648, prev=0,
next=0, firstchild=7efa78, lastchild=7efa78, pos={ x={ data=1078100987 }, y={ data=1078854136 }
}, rect={ left=133, top=156, right=1152, bottom=660 }, frame={ count=1, vector=1, elem0=7ef9a4

Upp::TopWindow::Run(appmodal=0)
GuiMainFn_()
Upp::AppExecute__(app=11589ea)
WinMain(hInstance=1150000->{ unused=9460301 }, __formal=0, lpCmdLine=c10e98 "",
nCmdShow=10)
__tmainCRTStartup()
WinMainCRTStartup()

```

75a7919f (KERNEL32.DLL)
77760bbb (ntdll.dll)
77760b91 (ntdll.dll)

thanks!

File Attachments

1) [ToolbarTest.log](#), downloaded 390 times

Subject: Re: bug in toolbar

Posted by [mirek](#) on Sat, 15 Nov 2014 16:55:55 GMT

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

Still no clue

Well, based on log and backtrace provided, I have abandoned theory about a problem in Image cache.

The next hypothesis is that ToolTip somehow overwrites random memory.

To test it, could you please commend out INITBLOCK in CtrlLib/ToolTip.cpp at line 137

```
/*  
INITBLOCK {  
  LLOG("Ctrl::InstallMouseHook(ToolTipHook)");  
  Ctrl::InstallMouseHook(ToolTipHook);  
  Ctrl::InstallStateHook(ToolTipStateHook);  
}  
*/
```

and test?

Also, I have noticed that Copy backtrace function does not list line numbers (and fixed that), so I would like to ask you to update ide sources, recompile and use this new ide for backtraces (if any)

Thanks for helping me with all this.

Mirek

Subject: Re: bug in toolbar

Posted by [mirek](#) on Sat, 15 Nov 2014 16:58:07 GMT

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

Ah, perhaps I owe you one explanation: That WindowProc issue is most likely caused by simple crashing (most likely invalid pointer because of overwritten memory); Windows then displays that "Program has been terminated" dialog, but that invokes WindowProc (e.g. WM_DEACTIVATE) and leads to that misleading error....

Subject: Re: bug in toolbar

Posted by [bushman](#) on Sat, 15 Nov 2014 19:57:03 GMT

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

mirek wrote on Sat, 15 November 2014 11:55

To test it, could you please commend out INITBLOCK in CtrlLib/ToolTip.cpp at line 137

```
/*  
INITBLOCK {  
  LLOG("Ctrl::InstallMouseHook(ToolTipHook)");  
  Ctrl::InstallMouseHook(ToolTipHook);  
  Ctrl::InstallStateHook(ToolTipStateHook);  
}  
*/
```

and test?

Did test as described above and got no more crashes, which should be ok at first sight; however, I should point out that I derived the test case I submitted in this topic from a much larger project I'm working on now and that crashes do persist in it after removing the tool tip hook as you suggest, which makes me even more confused.

mirek wrote on Sat, 15 November 2014 11:55

Also, I have noticed that Copy backtrace function does not list line numbers (and fixed that), so I would like to ask you to update ide sources, recompile and use this new ide for backtraces (if any

Did that, i.e., downloaded 7896.tar.gz from nightly builds, compiled fresh theide.exe, ran it, but I'm afraid backtrace function doesn't seem to list any line numbers yet (if I follow you correctly).

mirek wrote on Sat, 15 November 2014 11:55

Thanks for helping me with all this.

My pleasure! Actually, I'm learning a lot about how to debug with TheIde and I appreciate that tks again.

Subject: Re: bug in toolbar

Posted by [bushman](#) on Sat, 15 Nov 2014 20:45:48 GMT

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

Att. Mirek,

Pls, never mind when I said that the test case no longer crashes when commenting out tool tip mouse hook: after a few attempts, it does!

Besides, I tried with just one single ToolButton on the tool bar and it crashes the same way, just by moving the mouse pointer over the tool bar button and then around the test case window frame, outside it and back again into its workarea. Sometimes the tip label I got for the test case ("tip1") even shows unexpectedly over any other tool button on a background window, like when I slide the mouse pointer swiftly over the IDE toolbar buttons and hover it back again into the test case window work area, when it then crashes... Looks like its messier than I thought. The crash pattern is consistently always the same in the backtrace (Image::Data::Retain() -> USER32.dll ->...crash.

Thank you!

Subject: Re: bug in toolbar

Posted by [mirek](#) on Sat, 15 Nov 2014 22:01:03 GMT

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

Hm, well, I guess we can rule out the tooltip.

Another thing worth trying is to gradually remove (comment out, or place early return) parts of ToolButton::Paint and see if it is still crashing...

It will cause visual glitches, but events will still work.

Can you do that please?

Mirek

Subject: Re: bug in toolbar

Posted by [mirek](#) on Sat, 15 Nov 2014 22:03:37 GMT

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

kropniczki wrote on Sat, 15 November 2014 20:57

Also, I have noticed that Copy backtrace function does not list line numbers (and fixed that), so I would like to ask you to update ide sources, recompile and use this new ide for backtraces (if any

You have to use svn trunk for that... (will build to nightly this night .

Mirek

Subject: Re: bug in toolbar

Posted by [bushman](#) on Sun, 16 Nov 2014 22:20:07 GMT

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

Mirek,

Source of bug isolated, but resolution still pending: crashes occur when ToolButton::Paint calls GetMouseLeft() (see Win32Proc.cpp #71) to see if left mouse button was pressed. For some (unknown) reason, the call to user32.dll GetKeyState function trips the crashes, which I could resolve completely by just returning false from GetMouseLeft():

// in ToolButton.cpp:

```
void ToolButton::Paint(Draw& w)
{
    ...

    int li = IsEnabled() ? HasMouse() ? GetMouseLeft() ? CTRL_PRESSED
        : checked ? CTRL_HOTCHECKED : CTRL_HOT
        : checked ? CTRL_CHECKED : CTRL_NORMAL
        : CTRL_DISABLED;
    ...
}
```

// in Win32Proc.cpp, around line 71:

```
// returning false here was enough to cease all crashes!!
bool GetMouseLeft() { return false; } //!(GetKeyState(VK_LBUTTON) & 0x8000); }
```

The point now is to find out why. I checked the Win SDK specs but couldn't get any clues. Any ideas?

tnks!

Subject: Re: bug in toolbar

Posted by [bushman](#) on Sun, 16 Nov 2014 23:58:32 GMT

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

Subject: Re: bug in toolbar

Posted by [mirek](#) on Fri, 21 Nov 2014 13:20:58 GMT

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

Does deleting the last message mean in crashes again?

Mirek

Subject: Re: bug in toolbar

Posted by [bushman](#) on Fri, 21 Nov 2014 17:36:51 GMT

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

Yep! I'm afraid it does! Updating windows library hasn't worked. Actually, it happens and disappears just like it comes. I'm working on the hypothesis that some other process might be messing with user32.dll. I'll try to track which process(es) is(are) using it and block its(their) execution to see what happens. Any suggestions?

tk.

Subject: Re: bug in toolbar

Posted by [mirek](#) on Mon, 24 Nov 2014 10:19:43 GMT

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

kropniczki wrote on Fri, 21 November 2014 18:36 Yep! I'm afraid it does! Updating windows library hasn't worked. Actually, it happens and disappears just like it comes. I'm working on the hypothesis that some other process might be messing with user32.dll. I'll try to track which process(es) is(are) using it and block its(their) execution to see what happens. Any suggestions?

tk.

What about putting a big loop doing just GetMouseLeft into ToolButton::Paint? Then try in some more trivial situation (like single normal Button::Paint).

Mirek

Subject: Re: bug in toolbar

Posted by [bushman](#) on Mon, 01 Dec 2014 23:11:35 GMT

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

Have been busy lately, sorry.

1. Added big Button (200 X 200) to GUI

```
class App : public TopWindow
{
public:
typedef App CLASSNAME;
ToolBar bar;
```

```

Button btn;

void DoNothing() {}

void HandleBar(Bar& bar) {
    bar.Add("tip1", CtrlImg::copy(), THISBACK(DoNothing));
}

App()
{
    Add(bar.HSizePos().TopPos(0, 30));
    bar.Set(THISBACK(HandleBar));
    btn.SetLabel("test").Tip("crash test");
    Add(btn.LeftPos(8, 200).TopPos(40, 200));
}
};

GUI_APP_MAIN
{
    App().Run();
}

```

and inserted for-loop below into Button::Paint

```

void Button::Paint(Draw& w)
{
    for(int i = 0; i < 10000; i++)
        GetMouseLeft();
    ...
}

```

App crashes right after mouse pointer either enters or leaves Button rect, while jiggling mouse pointer inside Button area is not a problem.

Subject: Re: bug in toolbar
 Posted by [bushman](#) on Mon, 01 Dec 2014 23:24:24 GMT
[View Forum Message](#) <> [Reply to Message](#)

A fact: no crashes observed when checking IsPainting() before executing loop

```

void Button::Paint(Draw& w)
{
    if(!IsPainting()) // <- No Crashes!!
        for(int i = 0; i < 10000; i++)

```

```
GetMouseLeft();
```

Hope it helps. Tks.

Subject: Re: bug in toolbar
Posted by [bushman](#) on Tue, 02 Dec 2014 02:43:46 GMT
[View Forum Message](#) <> [Reply to Message](#)

Patching Win32Proc.cpp line #71 with

```
bool GetMouseLeft() { AvoidPaintingCheck__(); return !(GetKeyState(VK_LBUTTON) & 0x8000); }
```

resolved the problem, in spite of having added loops like

```
for(int i = 0; i < 10000; i++)  
    GetMouseLeft();
```

to both ToolButton::Paint and Button::Paint. In other words, I was able to 'switch' problem on/off just by removing/adding patch above, respectively.

Just don't know about eventual (unforeseen) consequences (if any) of patch to Upp code as a whole...

tk!

Subject: Re: bug in toolbar
Posted by [mirek](#) on Sat, 06 Dec 2014 17:27:07 GMT
[View Forum Message](#) <> [Reply to Message](#)

kropniczki wrote on Tue, 02 December 2014 03:43 Patching Win32Proc.cpp line #71 with

```
bool GetMouseLeft() { AvoidPaintingCheck__(); return !(GetKeyState(VK_LBUTTON) & 0x8000); }
```

resolved the problem, in spite of having added loops like

```
for(int i = 0; i < 10000; i++)  
    GetMouseLeft();
```

to both ToolButton::Paint and Button::Paint. In other words, I was able to 'switch' problem on/off just by removing/adding patch above, respectively.

Just don't know about eventual (unforeseen) consequences (if any) of patch to Upp code as a whole...

tnks!

Ha! That actually makes some sense: It is possible that something in your Windows setup routes GetKeyState through Window proc for some reason...

Thanks!

Mirek

Subject: Re: bug in toolbar
Posted by [mirek](#) on Sat, 06 Dec 2014 17:42:33 GMT
[View Forum Message](#) <> [Reply to Message](#)

Fix: (committed, please check)

```
dword GetKeyStateSafe(dword what) {  
    bool h = Ctrl::painting;  
    Ctrl::painting = false;  
    dword r = GetKeyState(what);  
    Ctrl::painting = h;  
    return r;  
}
```

```
bool GetShift()    { return !!(GetKeyStateSafe(VK_SHIFT) & 0x8000); }  
bool GetCtrl()    { return !!(GetKeyStateSafe(VK_CONTROL) & 0x8000); }  
bool GetAlt()     { return !!(GetKeyStateSafe(VK_MENU) & 0x8000); }  
bool GetCapsLock() { return !!(GetKeyStateSafe(VK_CAPITAL) & 1); }  
bool GetMouseLeft() { return !!(GetKeyStateSafe(VK_LBUTTON) & 0x8000); }  
bool GetMouseRight() { return !!(GetKeyStateSafe(VK_RBUTTON) & 0x8000); }  
bool GetMouseMiddle() { return !!(GetKeyStateSafe(VK_MBUTTON) & 0x8000); }
```

Subject: Re: bug in toolbar
Posted by [bushman](#) on Sun, 07 Dec 2014 13:27:42 GMT
[View Forum Message](#) <> [Reply to Message](#)

mirek wrote on Sat, 06 December 2014 12:42Fix: (committed, please check)

Done that, fix applied, many thanks!
