
Subject: Re: TheIDE stuck in infinite loop
Posted by [Didier](#) on Sat, 14 Dec 2024 09:14:19 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hello,

Since last week I have the exact same problem (never had this one in all the years using Ide before) : Ide freezes when I type a letter (It comes back to life after some time I didn't measure the time but about 20s or more ==> the poll timeout ?)

It all started when I created a new sub package for my app and directly decided to erase it (using delete form Ide menu).

Since it started on an action I rarely do: I thought some config or tmp files were messes up, so I tried erasing all the working data in the directories but Ide would still always freeze once I type the first letter.

To make it work again I had to rename my Ide to something else in order to provoke a fresh install (and therefor new .cache and working directories).

Since this morning, Here it is again

I compiled theide in debug mode (I have the latest git version no more than a few days old) and launched in debug ==> same freeze happens !

I erased the clipboard ==> still getting freeze

Here is the backtrace I get:

Thread 1 "ide" received signal SIGINT, Interrupt.

```
0x00007ffff632736d in __GI___poll (fds=0x1aa5340, nfds=3, timeout=20) at
../sysdeps/unix/sysv/linux/poll.c:29
29     return SYSCALL_CANCEL (poll, fds, nfds, timeout);
```

(gdb) bt

```
#0 0x00007ffff632736d in __GI___poll (fds=0x1aa5340, nfds=3, timeout=20) at
../sysdeps/unix/sysv/linux/poll.c:29
#1 0x00007ffff6db2629 in g_main_context_poll (priority=<optimized out>, n_fds=3,
fds=0x1aa5340, timeout=<optimized out>, context=0x1ac67f0) at ../glib/gmain.c:4584
#2 g_main_context_iterate.isra.0 (context=0x1ac67f0, block=1, dispatch=1, self=<optimized out>)
at ../glib/gmain.c:4271
#3 0x00007ffff6d53aff in g_main_loop_run (loop=0x1fd3420) at ../glib/gmain.c:4479
#4 0x00007ffff779dc8e in gtk_clipboard_wait_for_contents
(clipboard=clipboard@entry=0x1fd3340, target=0x8b) at ../gtk/gtkclipboard.c:1436
#5 0x00007ffff77a0038 in gtk_clipboard_wait_is_text_available (clipboard=0x1fd3340) at
../gtk/gtkclipboard.c:1720
#6 0x000000000d86992 in Upp::Ctrl::Gclipboard::IsAvailable (this=0x19f7278
<Upp::Ctrl::gclipboard()::c>, fmt=...) at GtkClip.cpp:133
#7 0x000000000d87b40 in Upp::IsClipboardAvailableText () at GtkClip.cpp:299
#8 0x0000000004b4695 in Ide::EditorMenu (this=0x7fffffcfa0, bar=...) at Insert.cpp:333
#9 0x00000000061211a in Upp::callback<Ide, Ide, Upp::Bar&>(Ide*, void
(Ide::*)(Upp::Bar&))::lambda(Upp::Bar&)#1::operator()(Upp::Bar&) const (this=0x7ffff48b59d0,
```

```

a1=...) at /home/.../git_trunk/uppsrc/Core/CallbackNP.i:38
#10 0x0000000000612051 in Upp::Function<void (Upp::Bar&)>::Wrapper<Upp::callback<Ide, Ide,
Upp::Bar&>(Ide*, void (Ide::*)(Upp::Bar&))::lambda(Upp::Bar&)#1>::Execute(Upp::Bar&)
(this=0x7ffff48b59c0, args=...)
    at /home/.../git_trunk/uppsrc/Core/Function.h:17
#11 0x0000000000813132 in Upp::Function<void (Upp::Bar&)>::operator()(Upp::Bar&) const
(this=0x7fffffceba0, args=...) at /home/.../git_trunk/uppsrc/Core/Function.h:86
#12 0x000000000073897b in Upp::Bar::Scan(Upp::Function<void (Upp::Bar&)>, unsigned int)
(proc=..., key=67108873) at Bar.cpp:525
#13 0x0000000000748913 in Upp::LineEdit::Key (this=0x7fffffd1068, key=67108873, count=1) at
LineEdit.cpp:1258
#14 0x0000000000712e32 in Upp::CodeEditor::Key (this=0x7fffffd1068, code=67108873,
count=1) at CodeEditor.cpp:1274
#15 0x000000000048e7ba in AssistEditor::Key (this=0x7fffffd1068, key=67108873, count=1) at
Assist.cpp:1073
#16 0x0000000000d6d282 in Upp::Ctrl::DispatchKey (keycode=67108873, count=1) at
CtrlKbd.cpp:101
#17 0x0000000000d84292 in Upp::Ctrl::Proc (this=0x7fffffcfca0) at GtkEvent.cpp:690
#18 0x0000000000d84bcf in Upp::Ctrl::ProcessEvent0 (quit=0x0, fetch=true) at GtkEvent.cpp:766
#19 0x0000000000d84d15 in Upp::Ctrl::ProcessEvents0 (quit=0x0, fetch=true) at
GtkEvent.cpp:793
#20 0x0000000000d5234f in Upp::Ctrl::ProcessEvents (quit=0x0) at GtkEvent.cpp:810
#21 0x0000000000d720d4 in Upp::Ctrl::EventLoop (ctrl=0x7fffffcfca0) at GtkEvent.cpp:844
#22 0x0000000000d7606c in Upp::TopWindow::Run (this=0x7fffffcfca0, appmodal=false) at
TopWindow.cpp:323
#23 0x00000000004e62c3 in GuiMainFn_ () at main.cpp:436
#24 0x00000000009346ec in Upp::AppExecute__ (app=0x4e4c30 <GuiMainFn_(>) at
App.cpp:479
#25 0x00000000004e4b7d in main (argc=1, argv=0x7fffffdaf8, envptr=0x7fffffdaf8) at
main.cpp:177

```

Note the key pressed should be a 'v' or 'tab' (I may be mistaken)

I tried to get other backtraces in other conditions : I always get the same backtrace result
It seems quite strange : I type a letter, it freezes, and the backtrace points at Upp::Bar::Scan()
which, at the end waits for clipboard entry ???

My exact Git version is:

```

* commit b99c26b267b2a484ee709024ae0e1e0a1e5e1258 (HEAD -> master, origin/master,
origin/HEAD)
| Author: Mirek Fidler <mirek.fidler@gmail.com>
| Date: Sat Dec 7 12:57:52 2024 +0100
|
| ide: Error list skin sensitivity fixed
|

```

Here is the code for Upp::LineEdit:1258

```
bool LineEdit::Key(dword key, int count) {
    NextUndo();
    switch(key) {
    case K_CTRL_UP:
        ScrollUp();
        return true;
    case K_CTRL_DOWN:
        ScrollDown();
        return true;
    case K_INSERT:
        OverWriteMode(!IsOverWriteMode());
        break;
    }
    bool sel = key & K_SHIFT;
    doirectsel = key & K_ALT;
    dword k = key & ~K_SHIFT;
    if((key & (K_SHIFT|K_ALT)) == (K_SHIFT|K_ALT))
        k &= ~K_ALT;
    switch(k) {
    case K_CTRL_LEFT:
        {
            PlaceCaret(GetPrevWord(cursor), sel);
            break;
        }
    case K_CTRL_RIGHT:
        {
            PlaceCaret(GetNextWord(cursor), sel);
            break;
        }
    case K_LEFT:
        MoveLeft(sel);
        break;
    case K_RIGHT:
        MoveRight(sel);
        break;
    case K_HOME:
        MoveHome(sel);
        break;
    case K_END:
        MoveEnd(sel);
        break;
    case K_UP:
        MoveUp(sel);
        break;
    case K_DOWN:
        MoveDown(sel);
    }
```

```

break;
case K_PAGEUP:
    MovePageUp(sel);
    break;
case K_PAGEDOWN:
    MovePageDown(sel);
    break;
case K_CTRL_PAGEUP:
case K_CTRL_HOME:
    MoveTextBegin(sel);
    break;
case K_CTRL_PAGEDOWN:
case K_CTRL_END:
    MoveTextEnd(sel);
    break;
case K_CTRL_C:
case K_CTRL_INSERT:
    Copy();
    break;
case K_CTRL_A:
    SelectAll();
    break;
default:
    dorectsel = false;
    if(IsReadOnly())
        return MenuBar::Scan(WhenBar, key);
    switch(key) {
    case K_DELETE:
        DeleteChar();
        break;
    case K_BACKSPACE:
    case K_SHIFT|K_BACKSPACE:
        Backspace();
        break;
    case K_SHIFT_TAB:
        AlignChar();
        break;
    case K_CTRL_Y:
    case K_CTRL_L:
        if(cutline) {
            CutLine();
            break;
        }
    default:
        if(InsertChar(key, count, true))
            return true;
        return MenuBar::Scan(WhenBar, key); ***** L1258 *****
    }
}

```

```
    return true;
}
dorectsel = false;
Sync();
return true;
}
```

I guess InsertChar() didn't work for some reason ?

Note : all is running again ... and I didn't do anything to correct it (Assist is still running)
