
Subject: Re: TheIDE stuck in infinite loop
Posted by [Didier](#) on Mon, 16 Dec 2024 19:29:27 GMT
[View Forum Message](#) <> [Reply to Message](#)

jjacksonRIAB wrote on Sat, 14 December 2024 18:43Didier,

Yes, I think you found it. That looks like the same backtrace I had in gdb. It's entirely possible I mistakenly thought clearing the clipboard externally fixed it when it was simply the timeout being reached at coincidentally the same time, although IIRC I have triggered it also during a copy+paste operation. When I tried to paste the contents externally to see what was in there the buffer appeared to be empty. Maybe it had something bad in there?

But mainly what I don't understand is why you could be trying to just type things normally, no special characters or commands and it's checking the clipboard. Is TheIDE doing this after every key event? There may be a good reason for this, but I can't think of one.

I'm gonna lookup the 'key' value : since 'InsertChar(key, count, true)' doesn't react as expected, I suppose it's due to the value of 'key'.

Context: I was trying to type " virtual void Skin(..."
'v' ==> 0x76
'V' ==> 0x56
'tab' ==> 0x09

The 'key' values caught from the backtrace:
67108873 => 0x0400 0009 looks like 'tab'
69206102 => 0x0420 0056 looks like 'V'

So I guess the problem will come from the 0x0400 0000
The only thing that I can find that is close to it is:
CtrlCoreFlags::K_KEYUP = 0x4000000

Looking at 'InsertChar(...)' the constants used have the following value on my Linux.
K_CHAR_LIM = 0x0200 000
K_ENTER = 0x0d
K_SHIFT_SPACE = 0x0080 0020

So with the 'key' values from backtrace

```
                0xXXX    1    true
bool LineEdit::InsertChar(dword key, int count, bool canow) {
    if(key == K_TAB && !processtab) **** TEST IS FALSE ****
        return false;
    if(filter && key >= 32 && key < K_CHAR_LIM) **** TEST IS FALSE ****
        key = (*filter)(key);
    if(!IsReadOnly() && (key >= 32 && key < K_CHAR_LIM || key == '\t' || key == '\n' ||
```

```

    key == K_ENTER && processenter || key == K_SHIFT_SPACE))    **** TEST IS FALSE ****
    {
    if(key >= 128 && key < K_CHAR_LIM && (charset != CHARSET_UTF8 && charset !=
CHARSET_UTF8_BOM)
    && FromUnicode((wchar)key, charset) == DEFAULTCHAR)
    return true;
    if(!RemoveSelection() && overwrite && key != '\n' && key != K_ENTER && canow) {
    int64 q = cursor;
    int i = GetLinePos64(q);
    if(q + count - 1 < GetLineLength(i))
    Remove((int)cursor, (int)count);
    }
    WString text(key == K_ENTER ? '\n' : key == K_SHIFT_SPACE ? ' ' : key, count);
    Insert((int)cursor, text, true);
    PlaceCaret(cursor + count);
    Action();
    return true;
    }
    return false;    **** Exit comes from here ****
}

```

The origin of the problem is definitely the 0x4000000 bit set in the value

Looking at the void Ctrl::Proc() in GtkEvent.cpp ==> CtrlCoreFlags::K_KEYUP is definitely used

I think the 0x0400 0000 value comes from the line :

```
DispatchKey(!pressed * K_KEYUP + kv, CurrentEvent.count);
```

==> Keyboard is malfunctioning : key returns up before being completely processed ??

Note: I do have a brand new Dell wireless keyboard ==> Is this it : a keyboard problem (or kbd driver problem)????