Subject: WindowProc invoked while in Paint routine
Posted by NeilMonday on Wed, 24 Jul 2013 20:23:42 GMT

View Forum Message <> Reply to Message

I asked this in the newbie forum, but did not receive much of a response, so I thought I would try posting here.

I have an error, and cannot track down the solution. My application keeps asserting on line 86 of Win32Proc.cpp when I click on an Option control:

```
ASSERT_(!painting, "WindowProc invoked while in Paint routine");
```

I imagine this is because something is taking a long time to paint, or because `painting` is not being set to false when it is done.

This happens when I check a checkbox on a custom control, but there is nothing in the stack trace that references one of my classes.

Here is the stack trace:

```
Upp::AssertFailed...
Upp::Ctrl::WindowProc(533, 0, 0)
Upp::TopWindow::WindowProc(533, 0, 0)
Upp::Ctrl::WindowProc(hWnd, 533, 0, 0)
Upp::Ctrl::Refresh(x=17129912, y=1997078640, cx=22383264, cy=0);
_|find(key=0, base=1056210...)
```

This _|find() entry shows up in the stack trace about 14 times and is then followed by more Upp::Ctrl and Upp::Callbacks.

I am not really sure what else to post, so let me know what other info you would like to see. I am a bit lost here.

Here is some of my code:
First, here are a couple of helper classes that I am using:

```
class OptionWithKey : public Option
{
   typedef OptionWithKey CLASSNAME;

public:
   Callback1<int> WhenOptionWithKey;

   DEF_VAR(OptionWithKey&, Key, int, key, *this);

public:
   OptionWithKey() : key(-1)  { WhenAction = THISBACK(OnOptionAction); }
private:
    void OnOptionAction()  { WhenOptionWithKey(key); }
};

class OptionWithKeyArrayMap : public ArrayMap<int, OptionWithKey>
{
    typedef OptionWithKeyArrayMap CLASSNAME;

public:
    Callback1<int> WhenOption;

public:
    OptionWithKey& Add(int key)
    {
        OptionWithKey& option = ArrayMap<int, OptionWithKey>::Add(key);
        option.Key(key);
        option.WhenOptionWithKey = THISBACK(OnOptionAction);
        return option;
    }

private:
    void OnOptionAction(int key)  { WhenOption(key); }
};
```

I have a class that inherits from ChartCtrl called ANBarGraph:

```
class ANBarGraph : public ChartCtrl
{
    typedef ANBarGraph CLASSNAME;

...

public:
    OptionWithKeyArrayMap  instructorOptionArray;
private:
    void OnInstructorTrigger(int key);

...
};
```

In the constructor, I set the callback for the instructorOptionArray:

```
ANBarGraph::ANBarGraph()
{
    instructorOptionArray.WhenOption = THISBACK(OnInstructorTrigger);
}
```

I have a SetData function that sets up the instructorOptionArray:

```
void ANBarGraph::SetData(...)
{
...
   if (instructorOptionArray.Find(baseKey) < 0)
   {
      OptionWithKey& option = instructorOptionArray.Add(baseKey);
      option.Set(0).NoNotNull().SetLabel(t_("Instructor
trigger")).SetFont(SansSerif(14)).SetFrame(ThinInsetFrame());
   }
...
   if (instructorOptionArray.Find(groupKey) < 0)
   {
      OptionWithKey& option = instructorOptionArray.Add(groupKey);
      option.Set(0);
      groupInfo.triggerState =
ATaS::ANGroupInfo::TriggerState(int(instructorOptionArray.Get(groupKey)));
   }
}
```

I have found out that the problem happens in between the Pusher::LeftDown() and Pusher::LeftUp(). If I click the checkbox and hold the left mouse button down, the ASSERT gets hit before I release the mouse button.

I have a breakpoint set in "OnInstructorTrigger(int key)" function, but when I click the checkbox, it hits the ASSERT before it gets to the callback.

One thing that I saw that was strange is that the Refresh call always has huge numbers for x, y, and cx. While cy is always 0.