

---

Subject: Execute(), LPARAM and WPARAM  
Posted by [amando1957](#) on Thu, 13 Nov 2008 17:50:46 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Hi to all,

I tried an event handler like this:

```
class midi_in : public Withmidi_inLayout<TopWindow>
{
};
```

```
GUI_APP_MAIN
```

```
{
  midi_in inApp;
  // "lpGot" is just a dummy:
  LPARAM lpGot;

  for(;;)
    switch(inApp.Execute())
    {
      // for this I need the LPARAM and WPARAM:
      case MM_MIM_DATA:
        inApp.getMidiInData(lpGot);
        break;
      case WM_KEYDOWN:
        PromptOK(t_("Want to quit proggy?"));
        return;
    }
}
```

```
inApp.Run();
}
```

How can I read the LPARAM and WPARAM values now?  
Are there any get-functions?

Martin

---

---

Subject: Re: Execute(), LPARAM and WPARAM  
Posted by [mirek](#) on Fri, 14 Nov 2008 10:36:21 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

amando1957 wrote on Thu, 13 November 2008 12:50Hi to all,

I tried an event handler like this:

```
class midi_in : public Withmidi_inLayout<TopWindow>
{
};
```

```

GUI_APP_MAIN
{
midi_in inApp;
// "lpGot" is just a dummy:
LPARAM lpGot;

for(;;)
switch(inApp.Execute())
{
// for this I need the LPARAM and WPARAM:
case MM_MIM_DATA:
inApp.getMidiInData(lpGot);
break;
case WM_KEYDOWN:
PromptOK(t_("Want to quit proggy?"));
return;
}

inApp.Run();
}

```

How can I read the LPARAM and WPARAM values now?  
Are there any get-functions?

Martin

Not sure what "inApp.Execute" does and how it returns windows message.

However, to achieve what you need, override

```
virtual LRESULT Ctrl::WindowProc(UINT message, WPARAM wParam, LPARAM lParam);
```

Mirek

Subject: Re: Execute(), LPARAM and WPARAM  
Posted by [amando1957](#) on Fri, 14 Nov 2008 20:23:43 GMT  
[View Forum Message](#) <> [Reply to Message](#)

Made it like this now:  
LRESULT midi\_in::WindowProc(  
UINT message, WPARAM wParam, LPARAM lParam)  
{  
switch(message)

```
{
case MM_MIM_DATA:
  getMidInData(IParam);
  break;
case WM_KEYDOWN:
  if (wParam==27)
    if (PromptYesNo("Want to quit proggy?"))
      { Ctrl::ExitWin32();
      }
  return 0;
}

return Ctrl::WindowProc(message, wParam, lParam);
}
```

and that runs, given the call of the base function on bottom.  
Thanks for the hint.

Martin

---

---

Subject: Re: focus = transparent window  
Posted by [amando1957](#) on Sat, 15 Nov 2008 12:45:10 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

But makes a transparent window now:  
The desktop is being copied each time it gets the focus, looks like:

Dealing with focus I have found these, but cannot make sense from them:

```
void TopWindow::ActiveFocus0(Ctrl& ctrl)
void TopWindow::Activate()
```

Not to guess, as its not a child:  
void TopWindow::ChildGotFocus()

Made a seperate test case, but it did not repeat, there I got the expected grey window.

Martin

---

---

Subject: Re: focus = transparent window  
Posted by [mirek](#) on Sat, 15 Nov 2008 14:43:33 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

amando1957 wrote on Sat, 15 November 2008 07:45 But makes a transparent window now:  
The desktop is being copied each time it gets the focus, looks like:

Dealing with focus I have found these, but cannot make sense from them:

```
void TopWindow::ActiveFocus0(Ctrl& ctrl)
void TopWindow::Activate()
```

Not to guess, as its not a child:  
void TopWindow::ChildGotFocus()

Made a seperate test case, but it did not repeat, there I got the expected grey window.

Martin

IMO, event loop is not running.

Mirek

---

Subject: Re: focus = transparent window  
Posted by [amando1957](#) on Sun, 16 Nov 2008 12:29:15 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Quote:IMO, event loop is not running.

I made it like this now:  
GUI\_APP\_MAIN  
{  
  midi\_in inApp;  
  ///// not this:  
  // inApp.Run();  
  ///// but this:  
  TopWindow().Run();  
}

Now I have the grey window as desired, but the WM\_KEYDOWN is not handled anymore.  
I'll play around a little and find out former or later.  
Thanks for the hints.

Martin

---