

The Experienced Man's Notes for
Cross-Platform GUI Programming with wxWidgets
Version 1.0

Ryan Wilcox, Wilcox Development Solutions

2009-02-13

Abstract

Over the course of using Cross-Platform GUI Programming with wxWidgets every day for about two years, I amassed a bunch of clarifications to sections in the book. I'd like to share them with you here.

I recommend printing this book, then inserting each page into its logical location in the the wxWidgets book — to keep this information close at hand when you crack open your copy to look something up. Personally, I have taped these pages so they fold outwards, giving me three or four “open” pages at one time (without hiding the original book text).

See Appendix I in [Cross-Platform GUI Programming with wxWidgets](#) for a list of these and parameters for each.

An alternative way of writing the `MyTextCtrl::OnChar` function is:

```
void MyTextCtrl::OnChar(wxKeyEvent& event)
{
    if ( wxIsalpha( event.KeyCode() ) )
    {
        event.Skip(true);           //true is default parameter
    }
    else
    {
        // Illegal Keystroke. Eat the event up: we "handled" it
        wxBell();
        event.Skip(false);
    }
}
```

See page 176 for a list of key codes.

In the following chunk of code:

```
DECLARE_EVENT_TABLE(name, integer)
```

You could use some of the standard event IDs, for the integer parameter, found in `#include<wx/event.h>`.

See also macros `wx_DECLARE_EVT0`, `wx_DECLARE_EVT1`, and `wx_DECLARE_EVT2` to create event table entries that accept 0, 1, and 2 items, respectively.

Programatically Sending Events (additional comments)

```
wx...Event evt;           // e.g. wxKeyDownEvent, wxCommandEvent, etc.
evt.SetEventObject(obj);
// obj is what the event is about
// (logically a control, window, etc.).
```

```
bool res = GetEventHandler()->ProcessEvent(evt);
// top level-ish function.
// You might have to use wxGetTopLevelParent()
// to get our containing (frame, dialog, etc.).
```

Defining Custom Events

Page 37

Refer to Handling Events and Handling UI Updates on pages 249 and 250.

GetSize() returns the size of the window (how much the window would like to take up — either what you set it for via SetSize() or how much it wants to take up with sizers. . . at least, this seems to be the case).

GetBestSize() returns the minimal size the window's content takes up (equal to size used when Fit() is called).

For more information on wxListBox, wxCheckListBox, wxChoice, and wxComboBox, see pages 90–98.

The global `wxTopLevelWindows` variable is *a reference to* a `wxWindowList` object (which is of type `wxList`. For more information on `wxList`, see page 383). To use `wxTopLevelWindows` to get the oldest top-most frame, use `wxTopLevelWindows.GetFirst()`. Note that this might not be the front-most top level window.

To get the front-most window, use:

```
wxTopLevelWindow* output = NULL;
#ifdef __WXMAC__
    #include "wx/mac/private.h"
    output = static_cast<wxTopLevelWindow*>(wxFindWinFromMacWindow(FrontWindow()));
#else
    output = ::wxGetActiveWindow();
#endif

return (output);
```

wx-users Mailing List On This Topic

```
Newsgroups: comp.soft-sys.wxwindows
From: cso...@advancedconcepts.ch ('Stefan Csomor')
Date: Fri, 21 May 2004 18:34:46 +0000 (UTC)
Local: Fri, May 21 2004 1:34 pm
Subject: RE: Getting the frontmost window?
Hi Ryan
```

Either use `FrontWindow()` which is really the foremost window, ore use `ActiveNonFloatingWindow()` which is the foremost non-floater

HTH

Stefan

```
> -----Original Message-----
> From: Ryan Wilcox [mailto:ryanwil...@mac.com]
> Sent: Freitag, 21. Mai 2004 17:00
> To: wx-us...@lists.wxwidgets.org
> Subject: Getting the frontmost window?

> Hi folks,

> How would I go about getting the active (read: frontmost) window on
> 2.5.1(ish)[1] wxMac? I see ::wxGetActiveWindow(), but that's only for
> MS Windows.
> Thanks for the help,
> _Ryan Wilcox
```

wxNotebook::AddPage() call explained:

```
wxPanel* page = new wxPanel(...);  
wxString labelText = _("My tab");  
bool selected = false;           // (optional param)  
int wxImageListItem = 2;        // image index used in example (optional param)
```

```
notebook->AddPage(page, labelText, selected, wxImageListItem);
```

Refer to page 73 for further examples.

Table 4-20 wxBitmapButton Styles

Page 90

Another style is:

wxBORDER_NONE	No border for button. WARNING: On Mac OS X, this means your bitmap can only be of a standard size or it won't show (128x128, 48x48, 32x32, 16x16, 0x0).
---------------	--

Adding one more style, we have:

wxTE_PROCESS_ENTER	See wxTextCtrl documentation on pages 107–111, and notes on the next page of this document.
--------------------	---

Notes on wxComboBox events:

EVT_COMBOBOX(id, func)	Does NOT process the wxEVT_COMMAND_COMBOBOX_SELECTED command if a user <u>types in</u> a value instead of selecting one that is already in the list.
EVT_TEXT_ENTER(id, func)	Chooses an item from the menu generated when the ENTER key is pressed in the combo box (Applicable when using a generic combo box (one made up of a text control and a popup, instead of a unified control provided by the OS); this event is sent by the wxTextCtrl and is often useful in processing combo box selection).

wxComboBox Member Functions

Notes on wxComboBox member functions:

GetSelection()	Returns wxNOT_FOUND if the string entered is not equal to an item in the popup.
GetStringSelection()	Returns wxEmptyString if the string is not equal to item in the popup.
SetStringSelection()	Returns false if the value is not found in the list.
GetValue()	Used to get the value of the combo box if you don't care whether the user selected a value from the list or typed in a value.

End Of Sample

Like the idea? This is only a the first quarter of the book! Please buy the rest of this book at <http://www.wilcoxd.com/booklets/wxwidgets.html> .