

Introduzione a wxWidgets

La programmazione delle GUI resa facile

Davide “CD-RON” Rondini

Montebelluna Linux User Group

18 ottobre 2007 – Montebelluna



Licenza d'utilizzo

Copyright © 2007, Davide "CD-RON" Rondini.

Questo documento viene rilasciato secondo i termini della licenza Creative Commons (<http://creativecommons.org>).

L'utente è libero di:

distribuire, comunicare al pubblico, rappresentare o esporre in pubblico la presente opera

alle seguenti condizioni:

Attribuzione Deve riconoscere la paternità dell'opera all'autore originario.

Non commerciale Non può utilizzare quest'opera per scopi commerciali.

No opere derivate Non può alterare, trasformare o sviluppare quest'opera.

In occasione di ogni atto di riutilizzo o distribuzione, deve chiarire agli altri i termini della licenza di quest'opera.

Se ottiene il permesso dal titolare del diritto d'autore, è possibile rinunciare a ciascuna di queste condizioni. Le utilizzazioni libere e gli altri diritti non sono in nessun modo limitati da quanto sopra. Questo è un riassunto in lingua corrente dei concetti chiave della licenza completa (codice legale), reperibile sul sito Internet

<http://creativecommons.org/licenses/by-nc-nd/2.0/legalcode>



- 1 Cosa sono le wxWidgets
 - Descrizione generale e caratteristiche
- 2 wxWidgets e gli altri
 - Confronto con Qt e .NET
- 3 Editor grafici di finestre
- 4 Un programma di esempio





wxWidgets ***Cross-Platform GUI Toolkit***

- wxWidgets (ex *wxWindows*): framework di programmazione per la creazione di GUI in ambiente *multiplatforma*.
- wxWidgets è completamente OpenSource
- È basato su C++, con binding per altri linguaggi



Perché wxWidgets?

Esiste un certo numero di librerie per lo sviluppo di GUI multiplatforma. Nessuna di queste però, secondo l'opinione degli sviluppatori di wxWidgets, rispetta tutti e 4 questi requisiti:

- 1 Basso costo.
- 2 Disponibilità del codice sorgente.
- 3 Semplicità della programmazione.
- 4 Supporto per un vasto numero di compilatori.



Per sopperire sono nate le *wxWidgets*...



Cosa offre wxWidgets

- Realmente **multiplatforma**
- Supporta praticamente qualsiasi compilatore
- Documentazione di altissima qualità
- Wrapper leggero e veloce
- Includono classi di alto livello
- Buona struttura di astrazione
- wxCode: database di progetti degli utenti
- Gratis
- Open Source
- Progetto nato nel 1992 e ormai robusto



Feature disponibili

- Un vasto numero di classi per finestre e controlli
- Gestore degli eventi evoluto
- Supporto per la stampa su tutte le piattaforme
- Supporto per il debug e il logging
- Integrazione a database
- Drag and drop
- Gestione di file di configurazione
- Integrazione con OpenGL
- Multi-threading
- Programmazione di rete trasparente
- Gestione di HTML e XML integrata
- Supporto per l'internazionalizzazione
- Supporto OLE

Piattaforme [2]

- Win32 (Win 9x/NT/2K)
- Win64 (Windows XP/2003)
- Win16 (Windows 3.1)
- Linux x86
- Linux S/390
- FreeBSD
- NetBSD
- OpenBSD
- Mac OS
- Solaris
- AIX
- HP-UX
- IRIX
- SCO UnixWare
- DEC OSF/1 (a.k.a. Tru64)
- OS/2
- OpenVMS



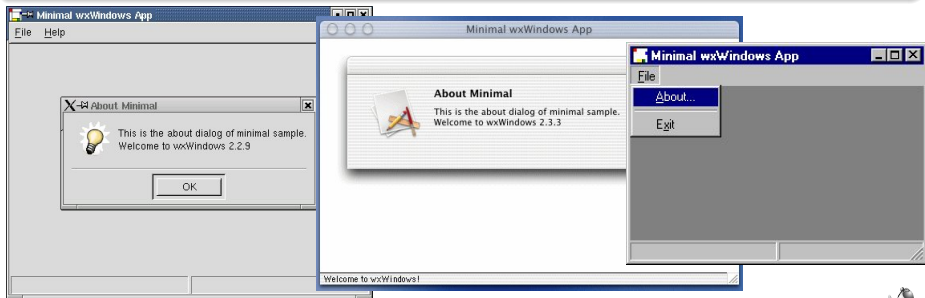
Binding [3]

- wxAda
- wxBasic
- wxD
- wxEuphoria
- wxHaskell
- wxJava
- jwx (alternativo a wxJava)
- wxJavaScript
- wxLua
- wxMax
- wxNet
- wxPerl
- wxPike
- wxPython
- wxRuby
- wxSqueak



wxWidgets e le API native

- Le funzioni non-GUI si appoggiano più possibile alle funzioni **ANSI C++** e alle **STL**
- Le widgets grafiche si appoggiano sempre sulle **API native** del sistema!



Porting stabili

- **wxGTK** Il port di default per Linux e Unix, basato su GTK+.
- **wxMSW** Il port per Microsoft[®] Windows, sia a 32 che 64 bit (98, 2000, NT, XP).
- **wxWinCE** Porting per Windows CE, comprese le edizioni Pocket PC e SmartPhone.
- **wxMac** Basato su Carbon per Mac OS 9 e OS X.
- **wxX11** Port diretto sulle librerie X11.
- **wxMGL** Port per il toolkit MGL sviluppato da SciTech Software Inc.
- **wxMotif** Per le varianti Linux e Unix che usano OpenMotif o Lesstif.



Porting in sviluppo

- **wxCocoa** Port per MacOS X, che si appoggia a Cocoa.
- **wxOS2** OS/2.
- **wxPalmOS** Porting per PalmOS.

wxEmbedded [4]

- Port commerciale per lavorare su piattaforme embedded^a.
- Sviluppato da un'azienda italiana.



^awxEmbedded[®] logo and name are registered trademarks of KOAN s.a.s. - ITALY





wxWidgets

wxWindows license

- Open Source (OSI approved)
- Compatibile L-GPLv2
- Consente di distribuire binari derivati con licenza a propria scelta.



Qt

Dual License

- GPLv2 per sviluppo Open Source
- Commerciale per applicazioni commerciali

.NET

.NET

Microsoft Reference License (Ms-RL)

- Consente solo di vedere il codice





wxWidgets

- Open Source
- Documentazione
- Facilità di programmazione
- Piattaforme e compilatori supportati



Qt

- Integrazione perfetta con KDE
- Molto evoluto
- Innovative
- Veloce

.NET

.NET

- Diffusione
- Supporto commerciale
- Nativo con Windows





wxWidgets

- Multi-threading poco evoluto
- Poco note



Qt

- Licenze commerciali costose
- Rognose da compilare in Windows

.NET

.NET

- Essenzialmente limitato a Windows
- Documentazione
- Vincolato alla politica commerciale Microsoft



DialogBlocks

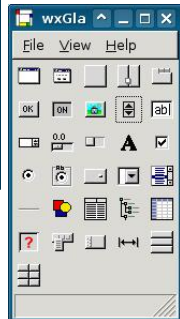
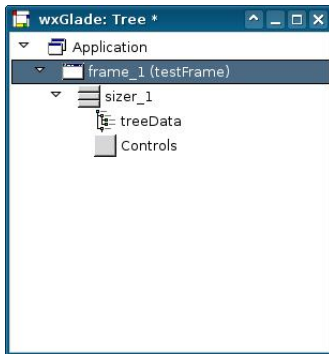
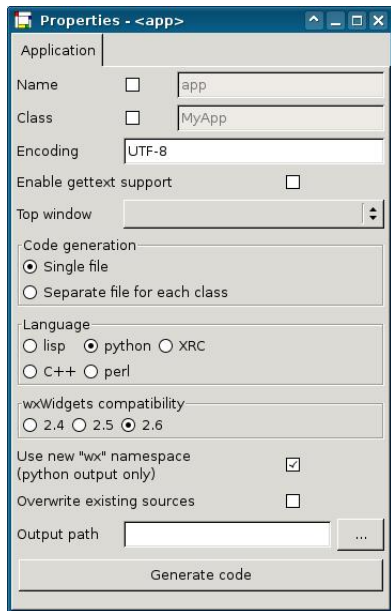
- IDE commerciale multiplatforma
- Sviluppato da Julian Smart, project leader di wxWidgets

wxGlade

- Widget editor
- Open source
- Scritto in wxPython
- Salva codice C++, python, Perl e XRC (XML Resource Compiler)

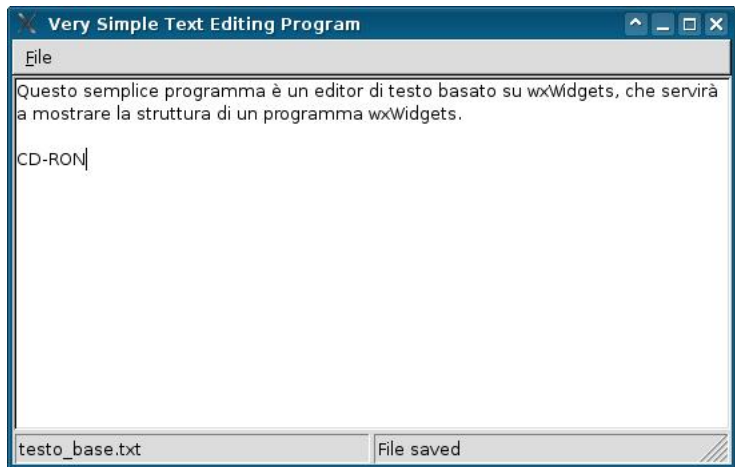


wxGlade screenshot



Un semplice tutorial

- Nel pacchetto delle wxWidgets ci sono tantissimi utili esempi
- Il programma di esempio sarà questo semplice editor di testo



```
class wxTestapp : public wxApp      1
{                                     2
    public:                           3
        virtual bool OnInit();      4
};                                    5
```

- L'applicazione viene costruita derivando una classe da wxApp
- Bisogna reimplementare il metodo virtuale OnInit()



```
class wxTestFrame : public wxFrame           1
{                                             2
    public:                                   3
        wxTestFrame( const wxString& title ,  4
                    const wxPoint& pos ,     5
                    const wxSize& pos );     6
        void OnQuit( wxCommandEvent& event ); 7
        void OnAbout( wxCommandEvent& event ); 8
        void OpenFile( wxCommandEvent& event ); 9
        void SaveFile( wxCommandEvent& event ); 10
        void SaveFileAs( wxCommandEvent& event ); 11
    private:                                  12
        DECLARE_EVENT_TABLE()                 13
        wxTextCtrl *textFrame;                14
        wxString currentFile;                 15
};                                             16
                                             17
```

La gestione delle finestre

- Per costruire una finestra, è sufficiente derivare da `wxFrame` o altre classi simili
- Tutti gli oggetti inseriti sono figli della finestra
- Il distruttore si occupa automaticamente di distruggere tutti i figli
- Per distruggere esplicitamente una finestra si usa il metodo `Destroy()`

I tipi di finestra

- `wxWindow` La finestra di base, senza decorazione. Usata spesso nelle sue derivazioni.
- `wxFrame` La finestra con decorazione, con attivabili menù e status bar.
- `wxDialog` Una finestra di dialogo, con i pulsanti Annulla e Ok.
- `wxWizard` Per creare wizard semplicemente e rapidamente.

Dichiarazione degli eventi

- La gestione degli eventi viene fatta associando degli ID a metodi particolari
- I metodi di gestione degli eventi come unico parametro un `wxEvent` e ritornano sempre `void`.
- Va assegnata la tabella degli ID degli eventi
- Se una classe gestisce eventi deve dichiararla con la macro `DECLARE_EVENT_TABLE()`

Gli ID degli eventi

```
enum                                1
{                                    2
    Menu_File_Quit = 100,          3
    Menu_File_About ,              4
    Menu_File_Open ,               5
    Menu_File_Save ,               6
    Menu_File_Save_As              7
};                                   8
```



La tabella degli eventi

```
#include <wx/wx.h> 1
#include <wx/textfile.h> 2
#include "wxtest.h" 3

BEGIN_EVENT_TABLE( wxTestFrame, wxFrame ) 4
    EVT_MENU( Menu_File_Quit, wxTestFrame::OnQuit ) 5
    EVT_MENU( Menu_File_About, wxTestFrame::OnAbout ) 6
    EVT_MENU( Menu_File_Open, wxTestFrame::OpenFile ) 7
    EVT_MENU( Menu_File_Save, wxTestFrame::SaveFile ) 8
    EVT_MENU( Menu_File_Save_As, wxTestFrame::SaveFileAs ) 9
    EVT_MENU( Menu_File_Save_As, wxTestFrame::SaveFileAs ) 10
END_EVENT_TABLE() 11
```



- La riga `BEGIN_EVENT_TABLE()` prende come parametri la classe e la sua genitrice `wxWidgets`, che definisce i tipi di eventi che sono implementabili
- Esiste una macro per ciascun tipo di evento (menu, mouse, comando, ecc)
- ogni macro associa l'ID dell'evento al metodo da chiamare quando questo evento viene generato
- Gli eventi sono *“propagativi”*: passano all'oggetto genitore se non processati.



L'implementazione di wxApp

```
IMPLEMENT_APP( wxTestapp ) 1
2
bool wxTestapp::OnInit() 3
{ 4
    wxTestFrame *frame = new wxTestFrame( 5
        wxT( "wxWidgets_Simple_Text_Editing_Program" ), 6
        wxPoint(50,50), wxSize(450,340) ); 7
8
    frame->Show(TRUE); 9
    SetTopWindow( frame ); 10
    return TRUE; 11
} 12
```

IMPLEMENT_APP() è la macro che implementa il main a partire dalla classe wxApp. È sempre necessaria in un programma wxWidgets.

Il costruttore di `wxFrame`

```
wxTestFrame :: wxTestFrame( const wxString& title ,           1
                             const wxPoint& pos ,             2
                             const wxSize& size )           3
: wxFrame((wxFrame *)NULL, -1, title , pos , size ,          4
          wxDEFAULT_FRAME_STYLE, wxT("Base_frame"))         5
```

Parametri

- 1 Puntatore alla `wxWindow` genitrice. NULL per la finestra principale.
- 2 ID della finestra, utile per alcuni metodi di gestione esplicita. -1 o `wxID_ANY` lasciano a `wxWidgets` impostare un valore.
- 3 Posizione della finestra (in pixel)
- 4 Dimensione di partenza della finestra (in pixel)
- 5 flag che definiscono lo stile della finestra

Creazione del menù File

All'interno del costruttore di `wxTestFrame...`

```
currentFile.Empty(); 1
wxMenu *menuFile = new wxMenu; 2
 3
menuFile->Append( Menu_File_Open, wxT("&Open..."), 4
                wxT("Open_text_file")); 5
// ... 6
menuFile->AppendSeparator(); 7
menuFile->Append( Menu_File_Quit, wxT("E&xit"), 8
                wxT("Exit_the_editor")); 9
10
wxMenuBar *menuBar = new wxMenuBar; 11
menuBar->Append( menuFile, wxT("&File") ); 12
SetMenuBar( menuBar ); 13
```



All'interno del costruttore di `wxTestFrame...`

```
CreateStatusBar(2); 1
SetStatusText( wxT( "Welcome_to_wxTextEditor!" ) ); 2
3
textFrame = new wxTextCtrl(this, -1, 4
                             wxT( "Write_Here\n" ), 5
                             wxDefaultPosition, 6
                             wxDefaultSize, 7
                             wxTE_MULTILINE | wxTE_AUTO_URL); 8
```

- 1 La status bar può essere divisa in più parti.
- 2 `wxTextCtrl` serve sia a creare semplici caselle di input, che a gestire editor avanzati con formattazione tipo RTF.



Apertura di un file

```
void wxTestFrame::OpenFile( wxCommandEvent& event )      1
{                                                         2
    wxFileDialog *openFileDialog =                       3
        new wxFileDialog( this ,                          4
                          wxT( "Open_a_text_file" ),     5
                          wxT( "/home/davide/" ),        6
                          wxT( "" ), fileTypees ,       7
                          wxOPEN, wxDefaultPosition );  8
    if ( openFileDialog->ShowModal() == wxID_OK ) {      9
        currentFile = openFileDialog->GetPath();         10
        textFrame->Clear();                               11
        textFrame->LoadFile( currentFile );              12
                                                         13
        SetStatusText( openFileDialog->GetFilename(), 0); 14
        SetStatusText( openFileDialog->GetDirectory(), 1); 15
    }                                                     16
}                                                         17
```

Salvataggio di un file

```
void wxTestFrame::SaveFile( wxCommandEvent& event ) 1
{ 2
    if (currentFile.IsEmpty()) { 3
        SaveFileAs(event); 4
    } else { 5
        if (textFrame->SaveFile(currentFile)) { 6
            SetStatusText(currentFile, 0); 7
            SetStatusText(wxT("File_saved"), 1); 8
        } else { 9
            wxMessageBox( 10
                wxT("An_error_occoured_while_saving_file!"), 11
                wxT("Save_error"), 12
                wxOK | wxICON_ERROR, this); 13
        }; 14
    }; 15
} 16
```

Salvataggio di un file con nome

```
void wxTestFrame::SaveFileAs( wxCommandEvent& event ) 1
{ 2
    wxFileDialog *saveFileDialog = 3
        new wxFileDialog( this , wxT("Save_text_file") , 4
            wxT("/home/davide/"), wxT(""), 5
            fileTypees , wxSAVE, wxDefaultPosition ); 6
    if ( saveFileDialog->ShowModal() == wxID_OK ) { 7
        currentFile = saveFileDialog->GetDirectory() 8
            + wxT("/") 9
            + saveFileDialog->GetFilename(); 10
        if ( textFrame->SaveFile(currentFile) ) { 11
            SetStatusText( saveFileDialog->GetFilename() , 0 ); 12
            SetStatusText( wxT("File_saved") , 1 ); 13
        } else { 14
            wxMessageBox( /* ... */ ); 15
        }; 16
    }; 17
} 18
```

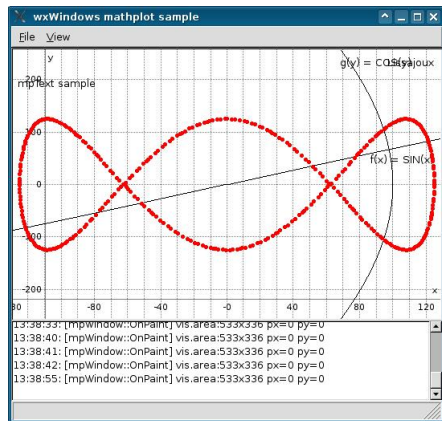
Conclusioni

Infine...

Abbiamo realizzato un editor di testo grafico multi-piattaforma con **147 righe di codice C++!**

Riferimenti

- <http://www.montellug.it>
- montellug@montellug.it
- davide.rondini@gmail.com



Lecture consigliate per approfondire...



[wxWidgets Home Page](http://www.wxwidgets.org/)

<http://www.wxwidgets.org/>



[wxWidgets supported platforms](http://www.wxwidgets.org/wiki/index.php/Supported_Platforms)

http://www.wxwidgets.org/wiki/index.php/Supported_Platforms



[wxEmbedded](http://www.koansoftware.com/it/prd_svil_wxembedded.htm)

http://www.koansoftware.com/it/prd_svil_wxembedded.htm



[wxWidgets bindings](http://www.wxwidgets.org/wiki/index.php/General_Information)

http://www.wxwidgets.org/wiki/index.php/General_Information



[Licenze dei framework](http://www.wxwidgets.org/about/newlicen.htm)

wxWidgets: <http://www.wxwidgets.org/about/newlicen.htm>

Qt: <http://trolltech.com/products/qt/licenses/licensing/licensingoverview>

.NET: <http://www.microsoft.com/resources/sharedsource/>

[licensingbasics/referencelicense.mspx](http://www.microsoft.com/resources/sharedsource/licensingbasics/referencelicense.mspx)

