

wxWidgets

Eine GUI-Bibliothek für plattformübergreifende Applikationen

Oliver Schonefeld

Universität Bielefeld
Fakultät für Linguistik und Literaturwissenschaften

28. November 2006

Agenda

1 Motivation

2 wxWidgets in der Theorie

3 wxWidgets in der Praxis

- Minimale wxWidgets Applikation
- Event Handling
- XML Resources (XRC)
- Advanced User Interface (AUI)

4 Ausblick

Was ist wxWidgets?

- Toolkit für Desktop- und Handheld-Applikationen
- *multi-platform C++ Framework*
- primär für graphische Oberflächen (GUIs)
- jedoch auch Abstraktion von Files, Streams, Threads, Konfigurationseinstellungen, ...

Warum wxWidgets?

- *natives* Look-und-Feel auf (fast) allen Ports
- einheitliches *multi-platform* Framework
- *Open Source*
- *wide industry support* u. a. AOL, AMD, CALTECH, Lockheed Martin und NASA nutzen wxWidgets
- wird von viele Open Source Projekten verwendet, z. B. Xara Xtreme, Audacity, pgAdmin III, aMule, Tortoise SVN, ...
- grosser Compiler Support, z. B. gcc, Visual Studio, Borland C++, ...)

wxWidgets Screenshot (1)

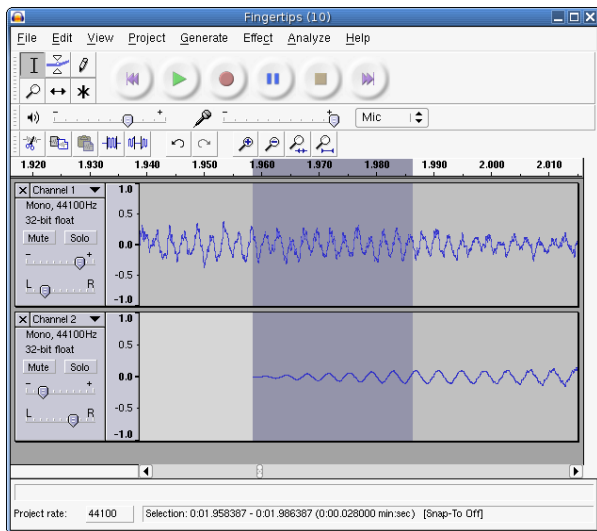


Abbildung: Screenshot von Audacity unter Linux

wxWidgets Screenshot (2)

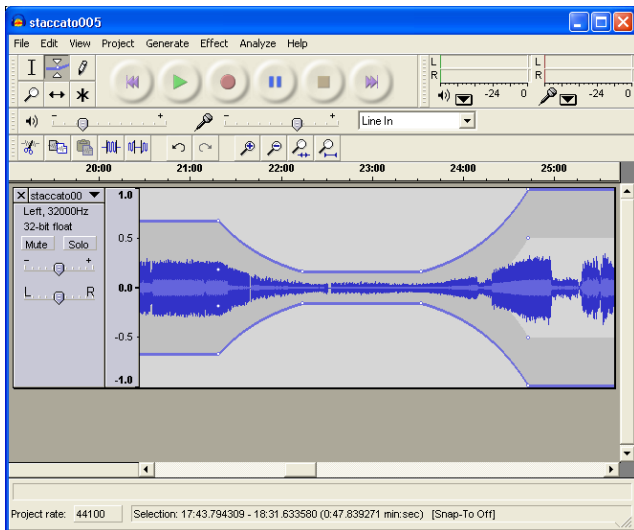


Abbildung: Screenshot von Audacity unter Windows

wxWidgets Screenshot (3)

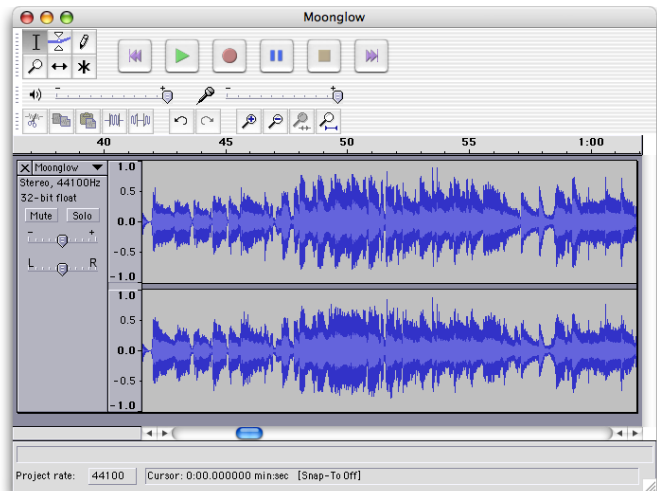


Abbildung: Screenshot von Audacity unter Mac OS X

Eine kurze Geschichte ... (1)

1992 wxWindows als Universitätsprojekt von Julian Smart gestartet

Sep. 1992 wxWindows 1.0 (XView und MFC 1.0)

... Pure-Win32 Implementation; Motif Port;
wxWindows wird mehr und mehr verwendet

1997 wxWindows 2 API

1998 GTK+ Port

1998 Mac OS Port (wird begonnen)

Eine kurze Geschichte ... (2)

1999 wxHTML und HTML-basierter Help-Viewer;
wxUniversal Port

2002 X11 Port

Jul. 2003 Windows CE Port

2004 „wxWindows“ wird auf „Druck“ von Microsoft in
„wxWidgets“ umbenannt; Mac OS X Port

Apr. 2005 wxWidgets 2.6

Dez. 2006 wxWidgets 2.8

Lizenz – wxLicense

- generell GPL/LGPL, aber Erweiterungen:
 - Projekte, die wxWidgets benutzen können unter beliebigen Lizenzen stehen
 - GPL-Code, der den wxWidgets-Quellen zugefügt wird, kann GPL beleiben
 - eigene Erweiterungen/Modifikationen der Library können diese Erweiterungen anlehnen

(einige) Ports

- wxMSW** 32- und 64-bit Versionen von Microsoft Windows sowie Windows CE
- wxGTK** GTK+ (1.x und 2.x) Port für „unixoide“ Plattformen
- wxX11** Xlib Port ohne natives Widget set; „schlank“
- wxMotif** Motif-Port (Motif, OpenMotif, LessTif)
- wxMac** Port für Mac OS 9 und höher
- wxCocoa** Mac OS X Port (Cocoa-API)
- wxOS2** Presentation Manager von OS/2

wxWidgets Architektur

wxWidgets API						
wxMSW	wxGTK	wxX11	wxMotif	wxMac	wxCocoa	wxOS2
Win32	GTK+	Xlib	Motif	Carbon	Cocoa	PM
Win/Win CE	Unix/Linux			OS 9/OS X	OS X	OS/2

Abbildung: wxWidgets Architektur

Minimale wxWidgets Applikation (1)

```
class MyApp : public wxApp {
public:
    virtual bool OnInit();
}; // class MyApp

// implementiert MyApp& wxGetApp()
DECLARE_APP(MyApp)

bool MyApp::OnInit() {
    MyFrame *frame =
        new MyFrame(wxT("Minimal wxWidgets App"));
    frame->Show(frame);
    return (true);
}
IMPLEMENT_APP(MyApp)
```

Minimale wxWidgets Applikation (2)

```
class MyFrame : public wxFrame {
public:
    MyFrame(const wxString& title);

    void OnAbout(wxCommandEvent &event);
    void OnQuit(wxCommandEvent &event);
private:
    DECLARE_EVENT_TABLE()
}; // class MyApp
```

Minimale wxWidgets Applikation (3)

```
MyFrame::MyFrame(const wxString& title) :  
    wxFrame(NULL, wxID_ANY, title) {  
  
    wxMenu *menu = new wxMenu();  
    menu->Append(wxID_ABOUT, wxT("&About ..."),  
        wxT("Show about dialog"));  
    menu->Append(wxID_EXIT, wxT("&Exit"),  
        wxT("Show about dialog"));  
  
    wxMenuBar *menubar = new wxMenuBar();  
    menubar->Append(menu, wxT("&File"));  
  
    SetMenuBar(menubar);  
}
```

Minimale wxWidgets Applikation (4)

```
void MyFrame::OnAbout(wxCommandEvent &event) {
    // About Dialog anzeigen
}

// ...

BEGIN_EVENT_TABLE(MyFrame, wxFrame)
    EVT_MENU(wxID_ABOUT, MyFrame::OnAbout)
    EVT_MENU(wxID_EXIT, MyFrame::OnQuit)
END_EVENT_TABLE()
```

Event Handling

- wxWidgets ist „event-driven“
- Events werden von weitergeleitet, bis sich ein EventHandler um das Event „kümmert“
- `wxEvt::Skip()` um Events an übergeordnete EventHandler weiterzuleiten
- Events verarbeiten:
 - von der bestehenden Klassen (z.B. `wxFrame`) ableiten und Events über Event-Table an eigene Methoden weiterleiten; bestehende (virtuelle) Methoden, falls vorhanden, können überschrieben werden, um das Verhalten eines Windows zu modifizieren (z. B. `OnPaint`, ...)
 - „pluggable event handlers“, d.h. eigene Klasse von `wxEvtHandler` ableiten und mit `PushEventHandler` zu einem Window hinzufügen

XML Resources (1)

- Oberfläche nicht statisch „codieren“, sondern dynamisch aus Beschreibung generieren
- Vorteile:
 - neues Übersetzen bzw. Linken bei Änderung der Ressourcen nicht notwendig
 - Separation von Code und Ressourcen
 - ermöglicht alternative Ressourcen
 - Sizer-based Layout für bessere platform-unabhängigkeit
 - GUI-Designer zum schnellen Erstellen der Oberfläche, z. B. DialogBlocks, wxDesigner, . . .
 - XML jedoch auch mit jedem (XML-)Editor bearbeitbar

XML Resources (2) – Resource Data

```
<resource version="2.3.0.1"
  xmlns="http://www.wxwidgets.org/wxxrc">
  <object class="wxDialog" name="aboutdialog"
    subclass="AboutDialog">
    <style>wxDEFAULT_DIALOG_STYLE</style>
    <exstyle>wxWS_EX_BLOCK_EVENTS</exstyle>
    <title>About</title>
    <object class="wxBoxSizer">
      <!-- ... -->
      <object class="wxPanel" name="wxID_ANY">
        <!-- ... -->
      </object>
    </object>
  </object>
</resource>
```

XML Resources (3) – Code

```
void MyApp::OnInit() {
    wxXmlResource::Get()->InitAllHandlers();
    if (!wxXmlResource::Get()->Load(wxT("about.xrc")))
        return (false);
    // ...
}
```

```
bool AboutDialog::Create(wxWindow *parent,
                        wxWindowID WXUNUSED(id)) {
    if (!wxXmlResource::Get()->LoadDialog(this,
                                           parent, wxT("aboutdialog")))
        return (false);
    // ...
}
```

XML Resources (4) – DialogBlocks

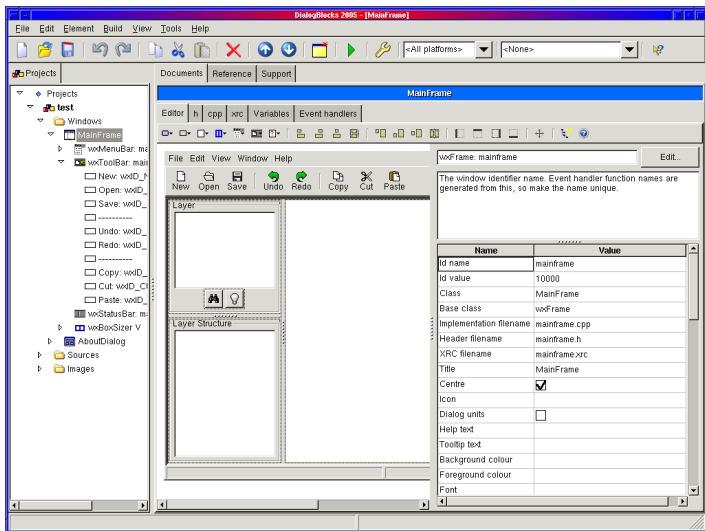


Abbildung: Screenshot von Dialogblocks

Advanced User Interface (1)

- “cutting-edge“ graphische Oberfläche
- Vereinigt folgende Aspekte:
 - Frame Management
 - Toolbars
 - Modeless Controls
 - Look and Feel
- Entwicklung von Kirix Corp.; jetzt in den wxWidgets Quellen

Advanced User Interface (2)

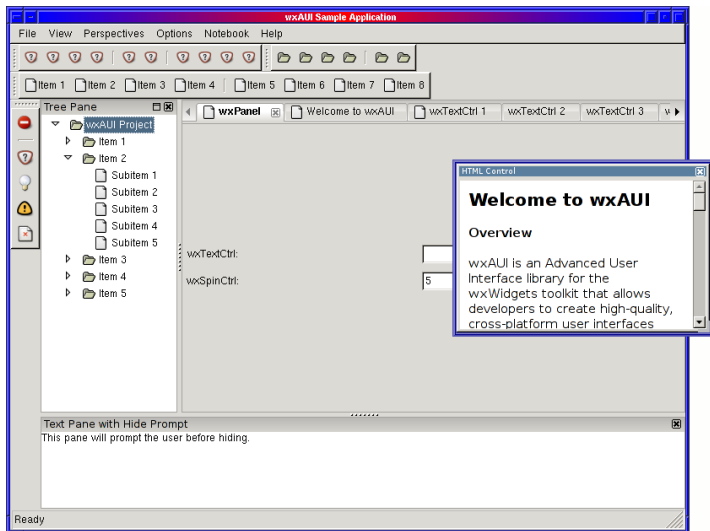


Abbildung: Screenshot der AUI-Demo

Andere Interessante Dinge

- *wxStyledTextCtrl* Quelltext-Editor (auf Basis von Scintilla)
- *wxRichTextCtrl* generischer WYSIWYG-Editor
- *Netzwerkklassen* verschiedene Klassen für Socket-basierte Netzwerkanwendungen
- *Validatoren* Filterung und Überprüfung von Benutzer-Eingaben
- *wxDB* ODBC-Schnittstelle
- „Language Bindings“, z B. wxPython, wxPerl, wxRuby, wxLua, ...

wxWidgets in der Zukunft ...

- Paket Management Tool (für 3rd-party Komponenten ⇒ z. B. <http://wxCode.sf.net>)
- besserer Support für Embedded Systeme
- alternativer Event-Handling Mechanismus
- erweitere Widgets (kombiniertes Tree/List Widget)
- wxHTML 2 („full webcapabilities on all platforms“)
- erweiterte Kompatibilität und Nutzung von Standards (z. B. STL, Boost)
- Palm OS 6 Port

(einige) Alternativen

- Qt – <http://www.trolltech.com/products/qt/>
- FOX – <http://www.fox-toolkit.org/>
- FLTK – <http://www.fltk.org/>

Quellen und Links

- Smart, Julian, Hock, Kevin und Csomor, Stefan:
Cross-Platform GUI Programming with wxWidgets.
Prentice Hall, 2005.
- wxWidgets Homepage – <http://www.wxwidgets.org>
- wxWidgets Tutorials –
<http://www.wxwidgets.org/docs/tutorials.htm>
- wxWiki – <http://wiki.wxwidgets.org/>
- wxForum – <http://www.wxforum.org/>
- wxGuide – <http://wxguide.sf.net/>
- wxCode – <http://wxcode.sf.net/>