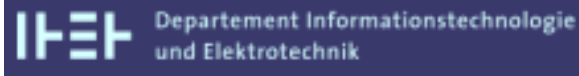


Sponsoren der /ch/open



Blue Saturn
THE OPEN SOURCE INTEGRATOR



Ptech
your full service provider



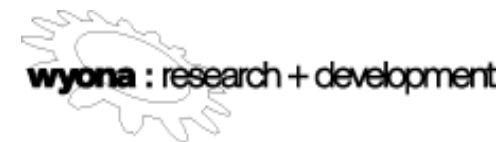
Novell.



RealStuff
Informatik AG



transtec



wxWidgets/ wxWindows

„An open source C++ GUI framework to make cross-platform programming child's play (almost)“



Inhaltsverzeichnis

- Motivation
- Einführung in wxWidgets
- Cross-Plattform Funktionalität
- Fallbeispiele
- wxWidgets & Python = wxPython
- wxDesigner
- Links/ Ressourcen
- Aussichten/ Zusammenfassung



Motivation

- Entwickler
 - einmal erworbenes Wissen kann auf verschiedenen Plattformen wiederverwendet werden
 - aktive Entwicklergemeinde
 - gute Dokumentation, viele Beispiele
- Firma
 - Investitionsschutz durch Plattformunabhängigkeit
 - grosszügiges Lizenzmodell



Inhaltsverzeichnis

- Motivation
- Einführung in wxWidgets
- Cross-Plattform Funktionalität
- Fallbeispiele
- wxWidgets & Python = wxPython
- wxDesigner
- Links/ Ressourcen
- Aussichten/ Zusammenfassung



Vergleich Cross-Plattform Frameworks

- Swing
 - pluggable Look and Feel (UIDelegates) / laufzeit / sehr langsam
- SWX (Eclipse)
 - native Look and Feel / gelinkt / langsam
- wxWidgets
 - native Look and Feel / gelinkt / schnell / robust



Was ist wxWidgets?

- C++ GUI (und mehr) Framework
- Support von...
 - Multi-Plattformen, Multi-Language, Multi-Device
- verwendet kleinster gemeinsamer C++ Nenner
- native Look and Feel
- ausgereifte Technik
(Projektstart 1992)
- einsetzbar für Closed- und Open-Source Projekte
(wxWidgets Lizenz v3 ist „entschärfte“ LGPL v2)

Native Multi-Plattform Support

- native Widgets werden gekapselt
- ein identisches API auf allen Plattformen
- eine Library pro unterstützter Plattform

Same program source for all platforms							
wxWindows API							
wxMSW	wxX11	wxGTK	wxMotif	wxMac	wxBase		
GDI	Xlib/X11	GTK+	Motif	Mac	without GUI		
Windows	Unix ⁽¹⁾			MacOS	Windows	Unix ⁽¹⁾	MacOS OS/2

(C)www.KoanSoftware.com

<http://www.wxwindows.org/platform.htm>
<http://www.wxwindows.org/supported.htm>

Universal Multi-Plattform Support

- Widgets werden in wxWidgets selber implementiert (benutzt Plattform GDI)
- Portierung auf neue Plattformen einfach/ schnell
- identisches API wie für native Plattform Support

Same program source for all platforms				
wxWindows API				
wxUniversal				
wxMSW	wxX11	wxGTK	wxOS2	wxMGL
GDI	Xlib/X11	GTK+	GDI	
Windows	Unix ⁽¹⁾		OS/2	Various OS

(C)www.KoanSoftware.com

(1) linux, solaris, aix

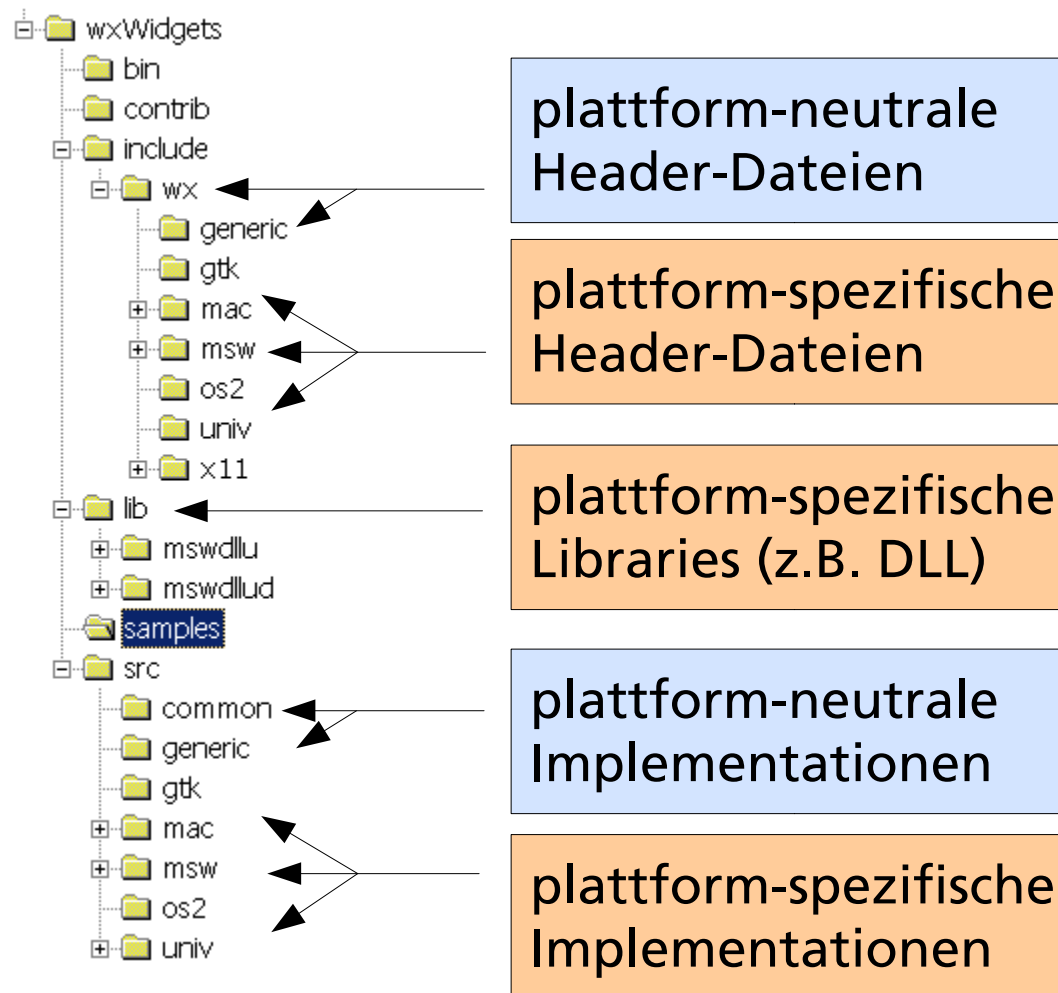
Multi-Compiler/-Language Support

- Unterstützung von diversen Compilern
 - VC++, BC++, GCC (MinGW), Sun CC, HP CC
 - <http://www.wxwindows.org/platform.htm>
- Unterstützung für diverse Programmiersprachen (Bindings) vorhanden
 - Python, Java, C# (.NET/Mono), Ruby,...



(C)www.KoanSoftware.com

Library Organisation



Library Konfiguration

- eine Library pro Plattform & Konfiguration
- Libraries mittels Make-/ Projekt-Files erstellen
- pro Plattform gibt es ein Header-File welches die Library konfiguriert.

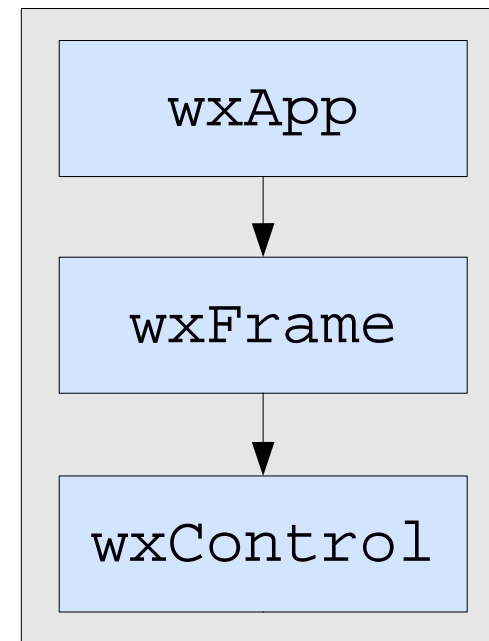
– ../wxWindows/include/wx/**msw**/setup.h

```
// Use standard C++ streams if 1.  
// If 0, use wxWin streams implementation.  
#define wxUSE_STD_IOSTREAM 0  
  
// Set to 1 to enable virtual ZIP filesystem  
#define wxUSE_FS_ZIP 1
```

- Resultat: native Look & Feel, statisch/dyn. gelinkt

C++ Hello World

- Eine Applikation besteht aus
 - Applikations-Objekt
 - Frame-Objekt
(kann Menüs, Statuszeile, etc. enthalten)
 - Control-Objekten
(z.B. Buttons, Text, etc.)
 - Kommunikation zwischen den UI-Objekten basiert auf Events



Inhaltsverzeichnis

- Motivation
- Einführung in wxWidgets
- Cross-Plattform Funktionalität
- Fallbeispiele
- wxWidgets & Python = wxPython
- wxDesigner
- Links/ Ressourcen
- Aussichten/ Zusammenfassung



Cross-Platform Funktionalität

- Mode/ View Datenaustausch
- Layout/ Sizers, XML Ressourcen
- Unicode and Localisation (i18n)
- Multithreading, Networking, File Handling
- Printing, Drawing (ein Interface, vs. DC's)
- Help System, Tool Tips
- Drag and Drop



Model/ View Datenaustausch (I)

- Problemstellung
 - Datenaustausch zwischen Controls/ Datenstrukturen
 - Validierung der eingegebenen Daten
- Lösung: „*Validators*“
 - können dynamisch hinzugefügt/ entfernt werden
 - wirken (falls nötig) rekursive auf alle Child-Controls
 - werden von den Default-Eventhandler für *Ok*, *Apply* und *Cancel* unterstützt
 - erhalten Events vor den Controls (Eventfilterung)

Model/ View Datenaustausch (II)

- Vorhandene Validatoren
 - wxTextValidator
(Datentransfer, Filterung, nur wxTextCtrl)
 - wxGenericValidator
(nur Datentransfer, für diverse Controls)
- Code Beispiel

```
// Validator hinzufügen  
aTextControl->SetValidator(wxTextValidator(wxFILTER_ALPHA, &m_text));  
  
// Validator entfernen  
aTextControl->SetValidator(wxDefaultValidator);
```

Model

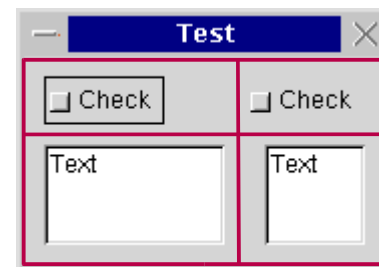
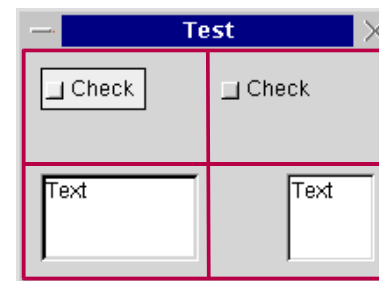
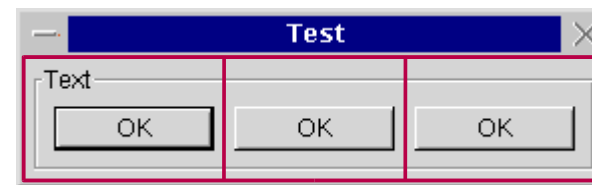
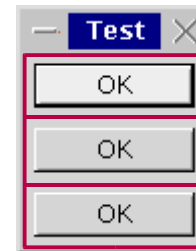
Layout/ Sizers (I)

- Eigenschaften
 - dynamische Bestimmung der Grösse und Position eines Controls zur Laufzeit
 - benutzt eine eigene Hierarchie die unabhängig von der Window/ Control-Hierarchie ist
- Vorteile
 - Layout ist plattformunabhängig
 - Grösse der Textfelder immer korrekt (mehrsprachige Applikationen)
 - Dialoge sind „resizeable“ (auch Windows)



Layout/ Sizers (II)

- Sizer-Typen
 - wxBoxSizer
wxStaticBoxSizer
(horizontal, vertical)
 - wxGridSizer
 - wxFlexGridSizer



Layout/ Sizers (III)

- Layout-Beschreibung
 - „hard coded“
 - XML Ressourcen (XRC)
- Layout-Erzeugung
 - von Hand
 - mit Tools
(z.B. wxDesigner, XRCed, wxGlade)
- Demo folgt später (im Teil wxPython)



Unicode and Localisation

- Library muss mit Unicode-Support compilieren
- Unicode Support
 - wenn immer möglich `wxString` verwenden
 - `wxString str = _T("hello world");`
 - `wxChar ch = _T('*');`
- Localisation
 - `wxString str = _("hello world");`
 - `_("")` Makro ersetzt Strings zur Laufzeit mit der gewünschten Sprachimplementation

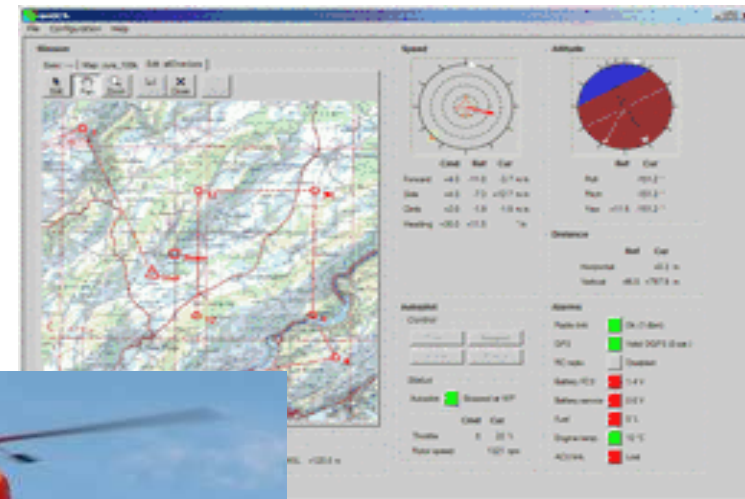
Inhaltsverzeichnis

- Motivation
- Einführung in wxWidgets
- Cross-Plattform Funktionalität
- Fallbeispiele
- wxWidgets & Python = wxPython
- wxDesigner
- Links/ Ressourcen
- Aussichten/ Zusammenfassung



Fallbeispiel „weGCS“

- Ground Control System (GCS) für ein autonomes Flugzeug/ Helikopter (weControl GmbH)
 - Steuerbefehle mittels Joystick
 - Visualisierung UAV Daten und Position
 - Flugwegplanung mittels grafischem Editor
 - Kalibration von gescannten Karten



Inhaltsverzeichnis

- Motivation
- Einführung in wxWidgets
- Cross-Plattform Funktionalität
- Fallbeispiele
- wxWidgets & Python = wxPython
- Demo: wxPython und wxDesigner
- Links/ Ressourcen
- Aussichten/ Zusammenfassung



Telinfo



Adressenansicht QuickLocate

Märki Informatik
Aehrenweg 6
8634 Hombrechtikon
Tel. *055 264 21 75

In.	Rubrik	Name	Vorname	Strasse	Nr	PLZ/	Ort
		Märki	Rosmarie	Eichwisstr.	27	8634	Hombrech
		Märki Informatik		Aehrenweg	6	8634	Hombrech
		Märki	Leila, Hans, Paul u. S...	Aehrenweg	6	8634	Hombrech
		Möbel-Märki Han...		St. Gallerstr.	120	8645	Jona

Adressbox 1 803 von 906

Telinfo

- Warum: Microsoft stoppt Macintosh-Unterstützung für MFC
- Entscheidung: wxWidgets / C++
- wxMac zu grossen Teilen neu geschrieben:
CH-Beitrag von Stefan Csomor



Telinfo - Rückblick

- 1% plattformabhängiger Code
- Geringere Wartungskosten
- Wesentlich bessere Performance, vor allem auf Macintosh
- Rückblick: "Sicher ist, dass wir mit keinem anderen Framework auch nur den Hauch einer Chance gehabt hätten, eine 'Single Source' Applikation zu schreiben." - Werner Thie

Telinfo - OSS Gratwanderung?

- wxWidgets: Open Source (LGPL)
- Telinfo: Closed Source, kommerzielles Kompilat
- Kommerzielles Closed Source Produkt profitiert von OSS
- OSS profitiert von Closed Source Produkt
 - Werner Thie und Stefan Csomor unterstützen wxWidgets finanziell und durch Mitarbeit (wxMac)



Inhaltsverzeichnis

- Motivation
- Einführung in wxWidgets
- Cross-Plattform Funktionalität
- Fallbeispiele
- wxWidgets & Python = wxPython
- Demo: wxPython und wxDesigner
- Links/ Ressourcen
- Aussichten/ Zusammenfassung



wxPython

- Python
 - Sehr schöne OO-Skriptsprache
 - Umfangreiche Libraries
- wxPython
 - Python-Wrapper für wxWidgets
 - Benötigt: Python + wxWidgets + wxPython



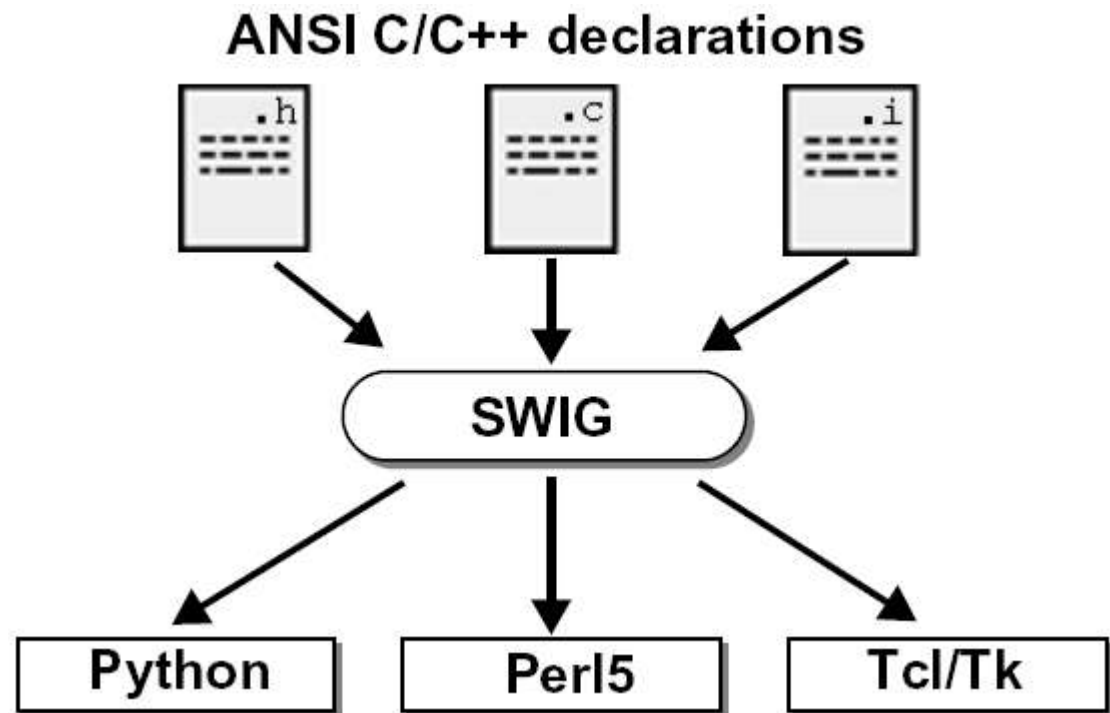
wxPython - Demo Applikation

- Demonstration
 - Core Windows Controls
 - Window Layout



wxPython - SWIG

- Simplified Wrapper and Interface Generator
- www.swig.org



wxPython - SWIG

- Die wxWidgets-C++-Dokumentation gilt auch für wxPython
- Unterstützte Sprachen
 - Python
 - Perl
 - PHP
 - Java
 - C#
 - ...



Inhaltsverzeichnis

- Motivation
- Einführung in wxWidgets
- Cross-Plattform Funktionalität
- Fallbeispiele
- wxWidgets & Python = wxPython
- Demo: wxPython und wxDesigner
- Links/ Ressourcen
- Aussichten/ Zusammenfassung



wxDesigner

- Erstellen von GUI's
- Proprietär, Closed Source. www.roebling.de
- Weniger leistungsfähig als Java-Designer, aber...
- Cross-Platform
- Cross-Language



wxDesigner - wxPython: Demo

- Was ich zeigen werde
 - Projekt Wizard
 - Dialog mit wxDesigner erstellen
 - Dialog aus wxApp aufrufen
 - Logik in Python auscodieren
 - C++-Dokumentation
 - Ziel: Einblick in Rapid Prototyping



Ihre ersten wxWidgets Gehversuche

- wxWidgets
 - wxWidgets herunterladen
 - 'demo' starten.
 - Viele Sample-Programme
- wxPython
 - python, wxWidgets und wxPython herunterladen.
 - 'demo' starten.



wxPython auf Suse9.0

- Packages installieren
 - wxGTK
 - python-wxGTK
 - python-wxGTK-examples
- wxPython-Demo starten
 - `/usr/share/doc/packages/python-wxGTK/demo/demo.py`



Inhaltsverzeichnis

- Motivation
- Einführung in wxWidgets
- Cross-Plattform Funktionalität
- Fallbeispiele
- wxWidgets & Python = wxPython
- Demo: wxPython und wxDesigner
- Links/ Ressourcen
- Aussichten/ Zusammenfassung

Links

- <http://www.wxWidgets.org>
- <http://wiki.wxWidgets.org>
- <http://wxguide.sourceforge.net>
- <http://www.wxPython.org>
- <http://wiki.wxPython.org>
- <http://www.roebling.de> (wxDesigner)



Aussichten

- Namhafte Unterstützung in der Weiterentwicklung von wxWidgets. OSAF Open Source Applications Foundation Chandler/Chandler/Personal Information Manager
- verstärkte Nachfrage nach OSS stärkt die Position von wxWidgets am Markt
 - staatliche Initiativen, kostengünstige Lösungen
- Vermehrter Einsatz von Linux im Desktopbereich
 - Multi-Plattform Support schützt Investition
- Expansion in den „embedded“ Bereich
 - wxWinCE, wxNano-X, wxMicrowindows



Zusammenfassung

- native Look and Feel, schnell und robust
- ein API für verschiedene Plattformen
- geeignet für Migration von MFC Applikationen
- geeignet für Rapid Prototyping
- gut dokumentiert, viele Beispiele vorhanden
- grosse, aktive Entwicklergemeinde (Mailing-Listen, Wiki)
- Kommerzieller Support



Apero offeriert von

