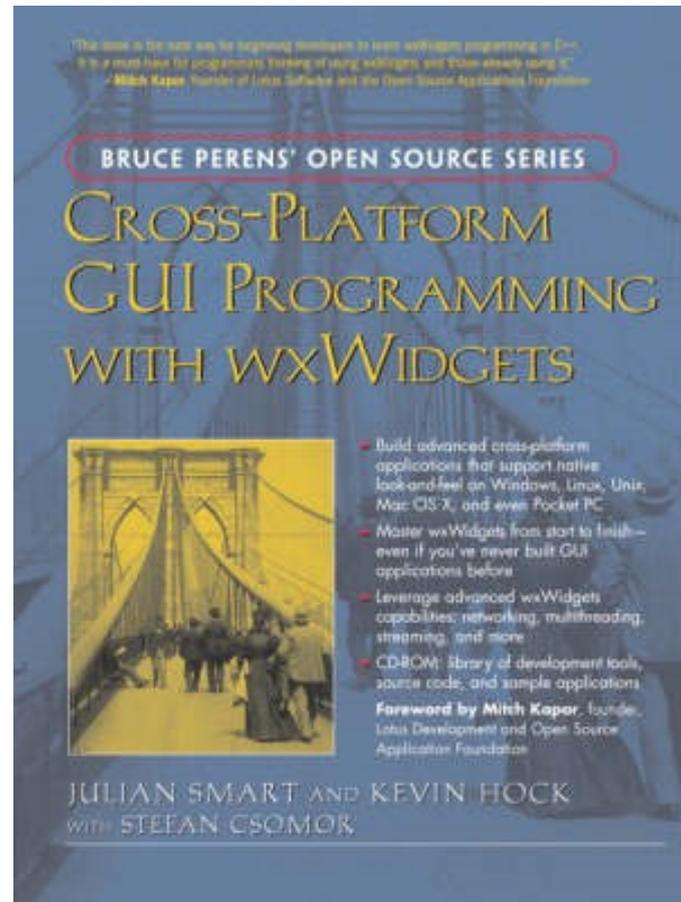
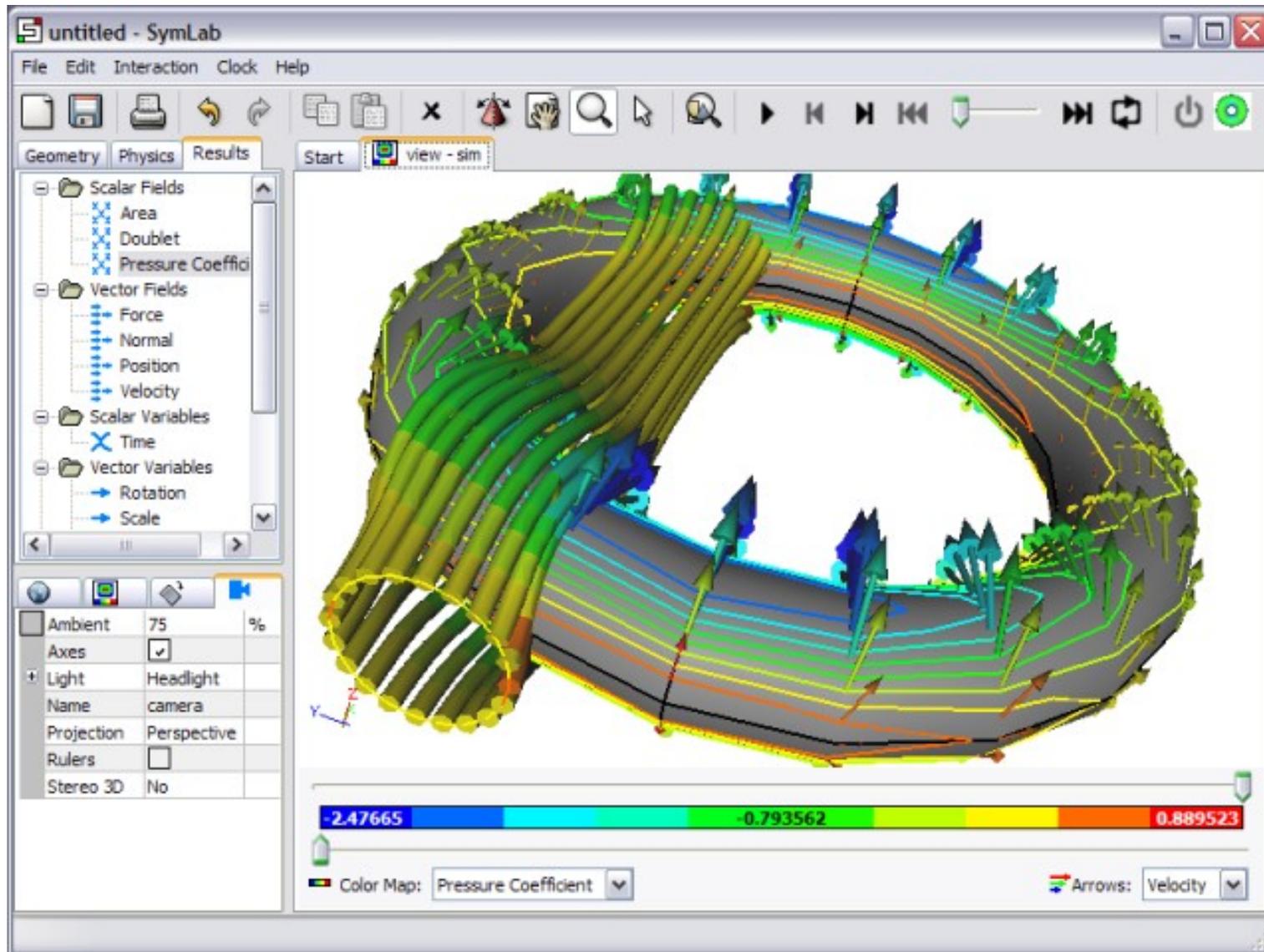


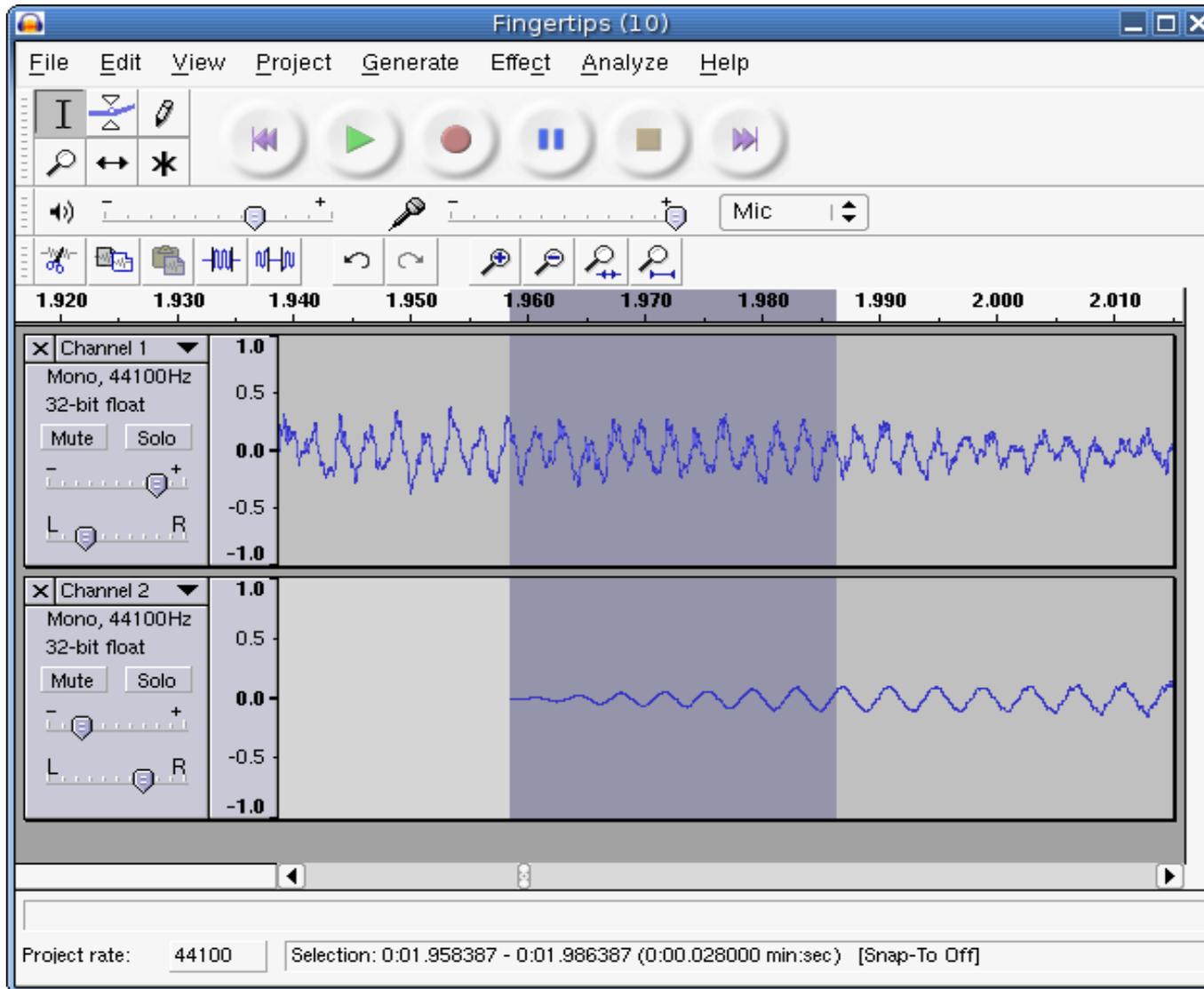
WxPerl Course



That's why you're interested



Audacity



Small Comparison Sheet



- TK: first, (TCL), used by many, less boilerplate, not native, many addons, maintenance mode



- TK+: nice API, very active, often not native

PerlQt: dead, but g code project perlqt4 active

FLTK: revived, lightweight, not native

Prima: nearly pure perl, perlsh API, consistent, nonnative, minimal design

Small Comparison Sheet



- TK: first, (TCL), used by many, less boilerplate, **not native**, many addons, maintenance mode



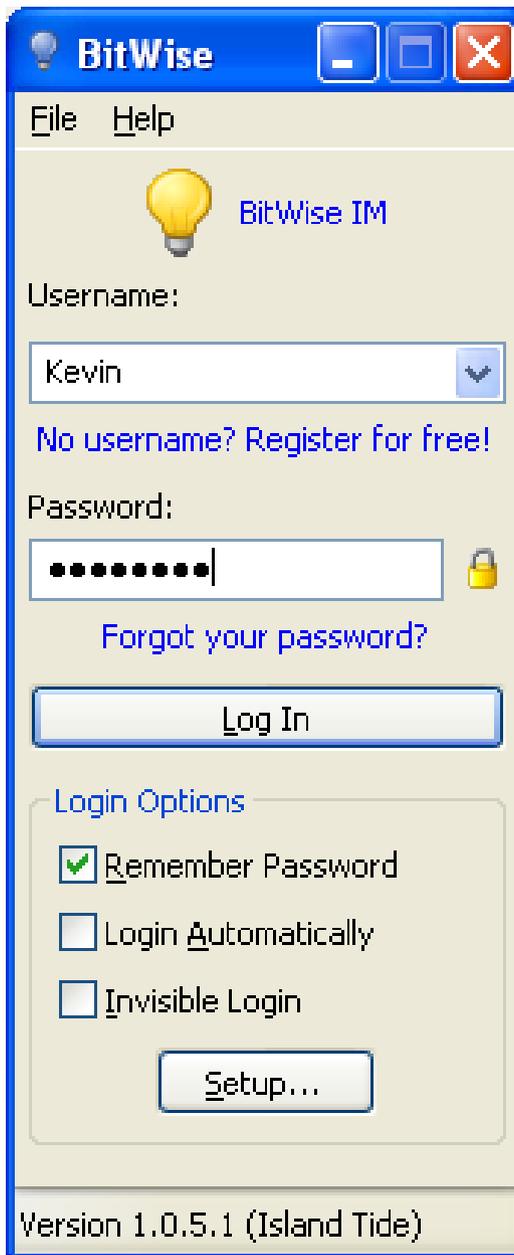
- **TK+**: nice API, very active, often **not native**

PerlQt: dead, but g code project perlqt4 active

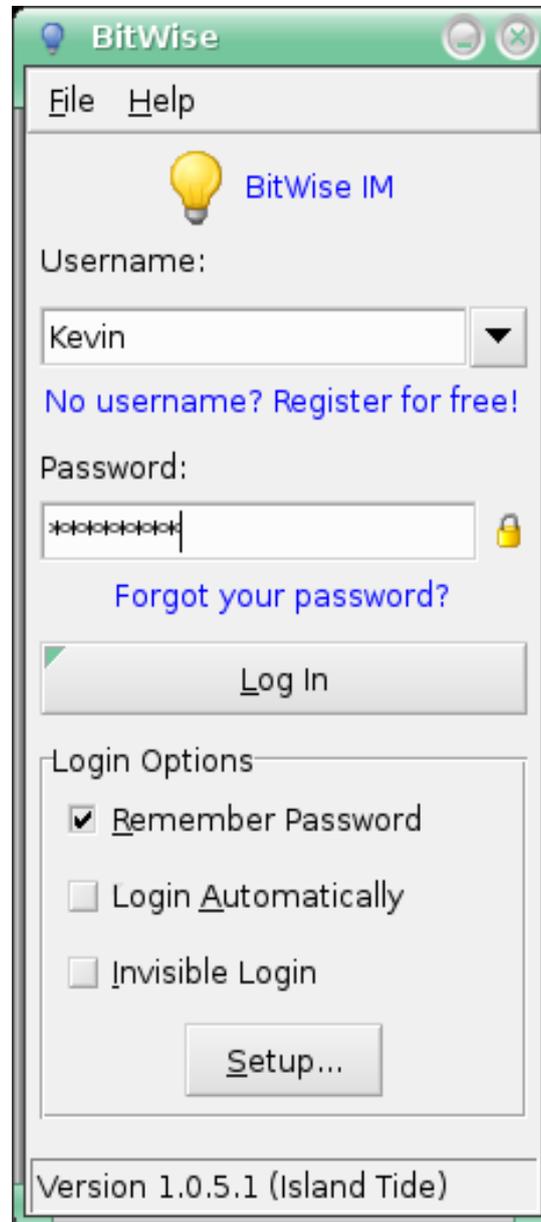
FLTK: revived, lightweight, **not native**

Prima: nearly pure perl, perlsh API,
consistent, **nonnative**, minimal design

native crossplatform



native crossplatform

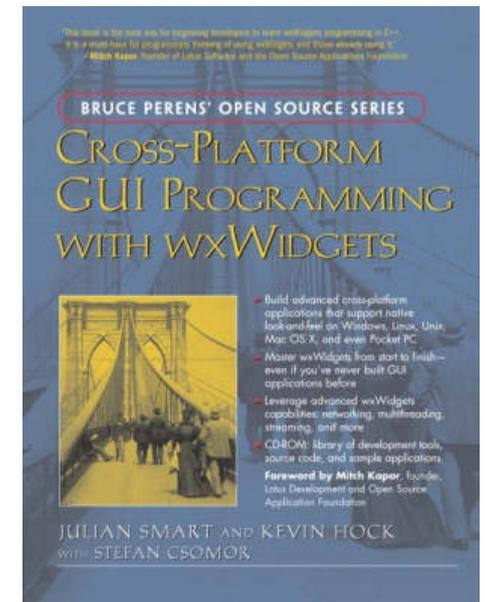


native crossplatform



What offers WxWidgets ???

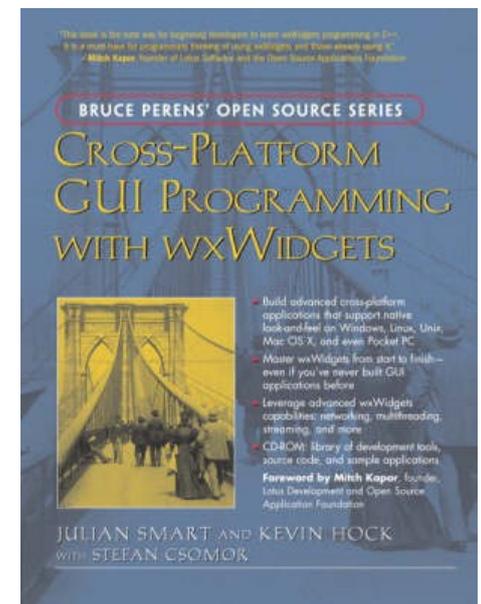
- Wx gives a native look 'n feel without skins



What offers WxWidgets ???

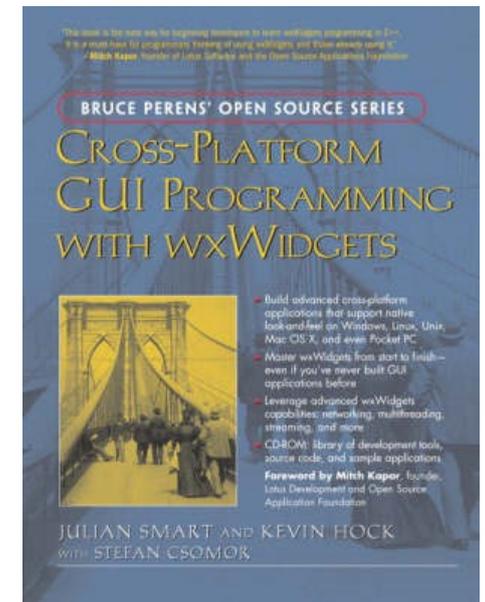
- Wx gives a native look 'n feel without skins

like Perl: cross-platform + platform specific



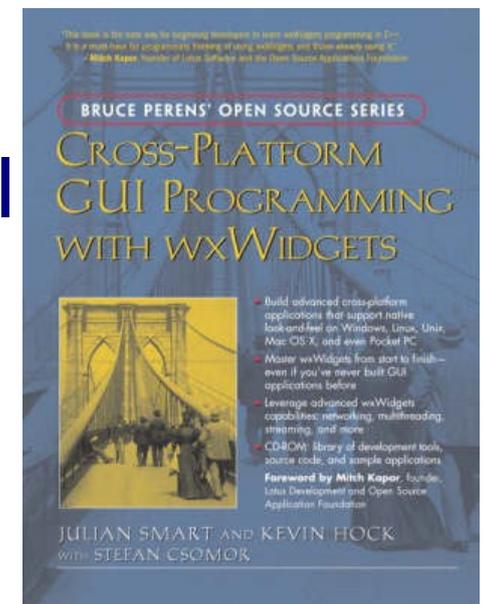
What offers WxWidgets ???

- Wx gives a native look 'n feel without skins
- fast & slim (compiled + delegates Rendering)



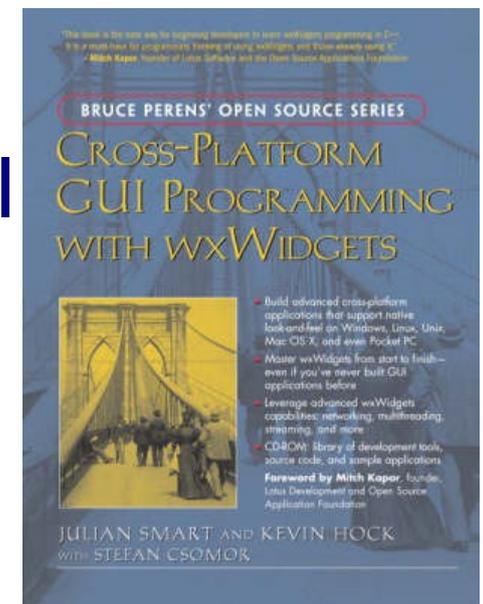
What offers WxWidgets ???

- Wx gives a native look 'n feel without skins
- fast & slim (compiled + delegates Rendering)
- consistent, cautiously maintained API



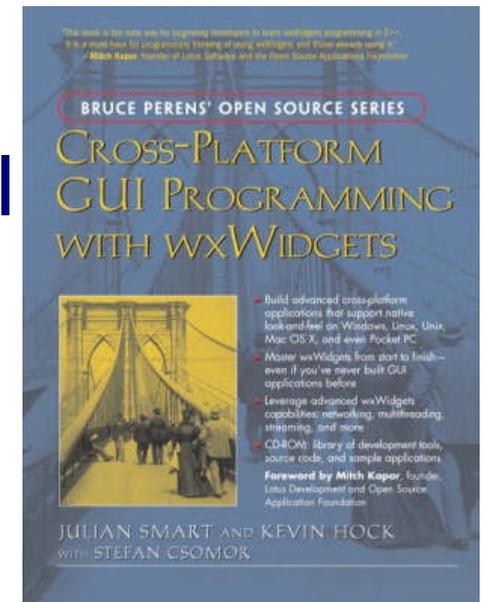
What offers WxWidgets ???

- Wx gives a native look 'n feel without skins
- fast & slim (compiled + delegates Rendering)
- consistent, cautiously maintained API
- featurerich (Framework for Apps)



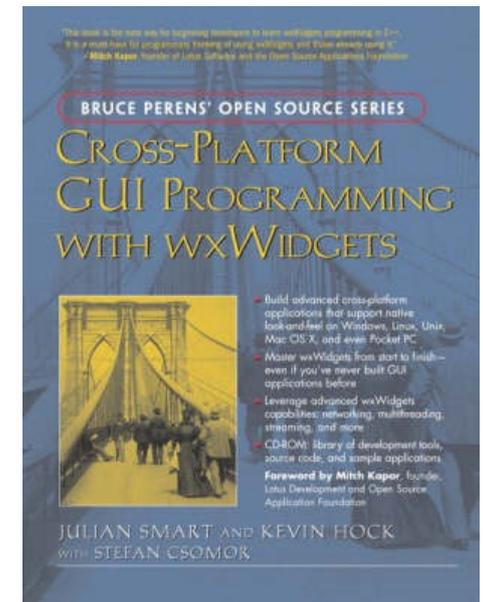
What offers WxWidgets ???

- Wx gives a native look 'n feel without skins
- fast & slim (compiled + delegates Rendering)
- consistent, cautiously maintained API
- **featurerich** (Framework for Apps)



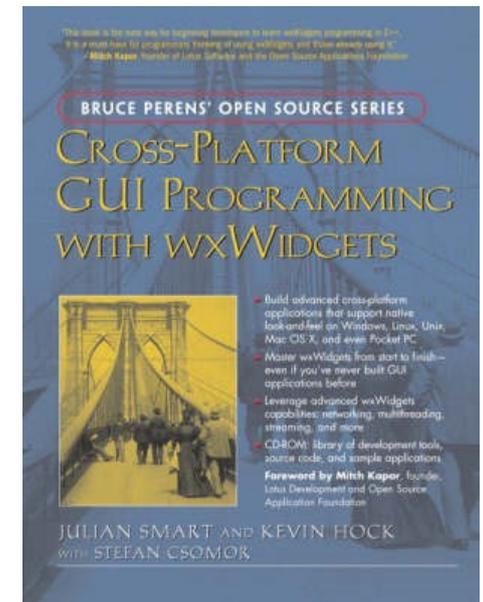
Usual GUI Stuff

- Buttons, Label, Inputfield
- Frames, Menu, Toolbar, Statusbar
- Dialogs, Selectors
- compound widgets



Extra Feature ?

- easy array / String / text file handling
- Regex / Unicode
- XML parser
- FTP / TCP / Zip / DateTime

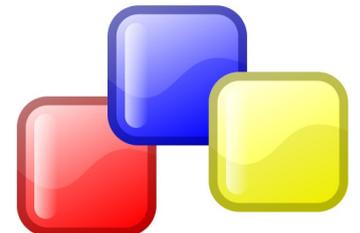


WxWidgets – the Factsheet

- founded 1992 by Julian Smart using C++

University Edinburgh project called Hardy

for XView and MFC 1.0

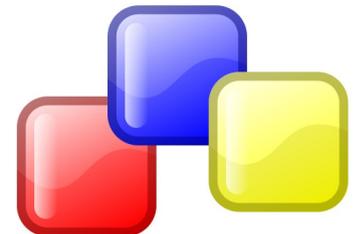


WxWidgets – the Factsheet

- founded 1992 by Julian Smart using C++
- **W**indows (W32, CE), **U**nix(Gtk,Motif, **X**11) & **M**ac (9&X)

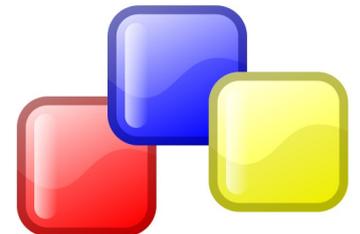
2004: wxWindows >> wxWidgets

today: **w**idgets **x**rossplatform



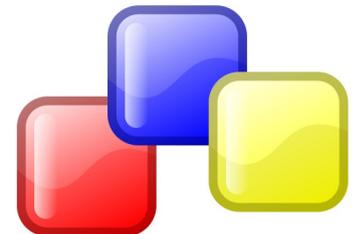
WxWidgets – the Factsheet

- founded 1992 by Julian Smart using C++
- Windows (W32, CE), Unix(Gtk, Motif, X11) & Mac (9&X)
- **LGPL** (WxWidgets License)



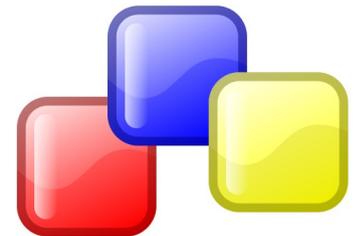
WxWidgets – the Factsheet

- founded 1992 by Julian Smart using C++
- Windows (W32, CE), Unix (Gtk, Motif, X11) & Mac (9&X)
- LGPL (WxWidgets License)
- mature, complete, in vital development



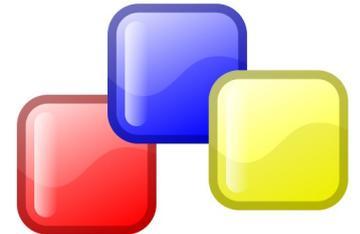
WxWidgets – development

- very friendly community



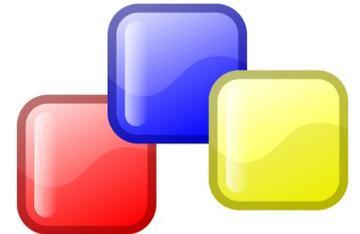
WxWidgets – development

- very friendly community
- conservative, conscientious, small core team



WxWidgets – development

- very friendly community
- conservative, conscientious, small core team
- good docs / lot examples on Perl side

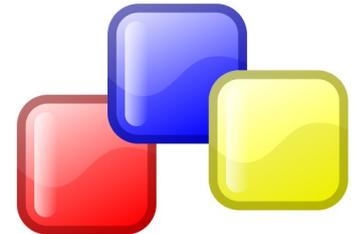


WxWidgets – development

- very friendly community
- conservative, conscientious, small core team
- good docs / lot examples on Perl side

- IDE: Code::Blocks, Dialog::Blocks

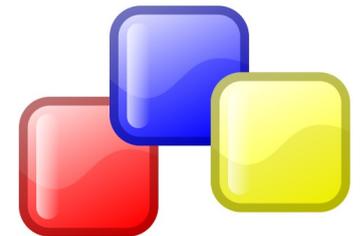
VisualWX, wxFormBuilder, wxDev-C++



WxWidgets – development

- very friendly community
- conservative, conscientious, small core team
- good docs / lot examples on Perl side

- IDE: Code::Blocks, Dialog::Blocks



VisualWX, wxFormBuilder, wxDev-C++

WxPerl – the Factsheet

- work of **Matia Barbon** (MBARBON) since **Nov 2000**

few, sporadic helper



WxPerl – the Factsheet

- work of Matia Barbon (MBARBON) since Nov 2000
- **XS** - Wrapper + Helper Classes



WxPerl – the Factsheet

- work of Matia Barbon (MBARBON) since Nov 2000
- XS - Wrapper + Helper Classes
- CPAN Module **Wx** (artistic license)



WxPerl – the Factsheet

- work of Matia Barbon (MBARBON) since Nov 2000
- XS - Wrapper + Helper Classes
- CPAN Module Wx (artistic license)
- uses **Alien::wxWidgets** since 0.57



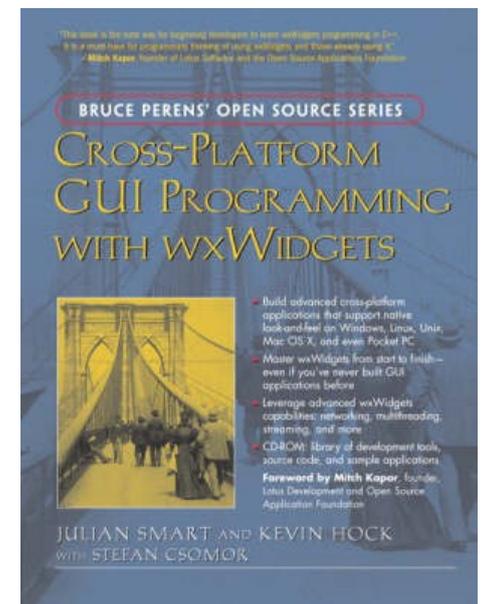
WxPerl – the Factsheet

- work of Matia Barbon (MBARBON) since Nov 2000
- XS - Wrapper + Helper Classes
- CPAN Module Wx (artistic license)
- uses Alien::wxWidgets since 0.57
- **all inclusive**



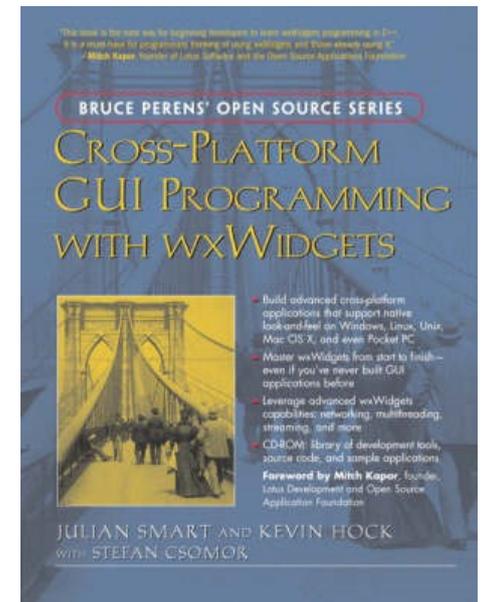
included

- nearly all GUI you know and more, HTML
- Printing, Drag' n Drop, Advanced UI
- medium image handling / logging
- XML based GUI abstraction format

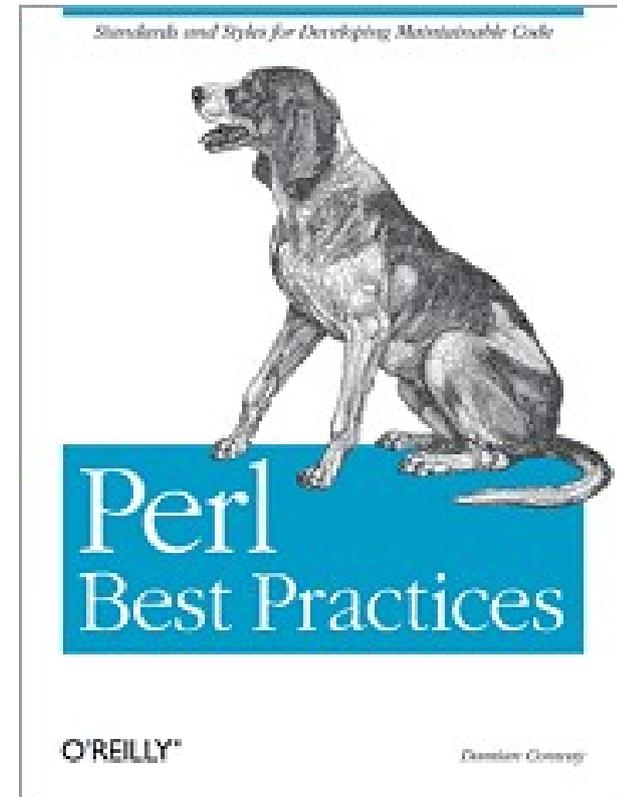


not included

- easy array / String / text file handling
- Regex / Unicode
- XML parser
- FTP / TCP / Zip / DateTime



Coding Style



Coding Style

- API is very consistent, but still smells like MFC



Coding Style

- API is very consistent, but still smells like MFC
- new API is a long term goal by Wx people



Coding Style

- API is very consistent, but still smells like MFC
- new API is a long term goal by Wx people
- some efforts on Perl side too

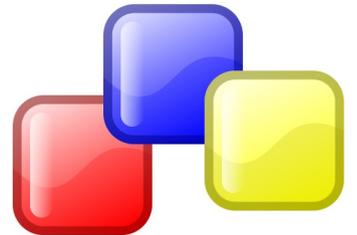


Coding Style

- API is very consistent, but still smells like MFC
- new API is a long term goal by Wx people
- some efforts on Perl side too
- by Mattia, me and other devs

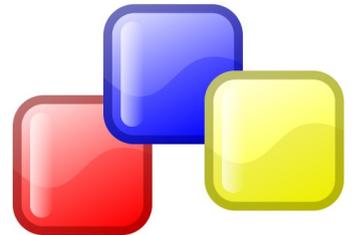


Everything is an Object



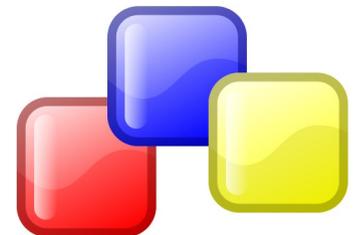
Everything is an Object

- except functions (boxes, clipboard)



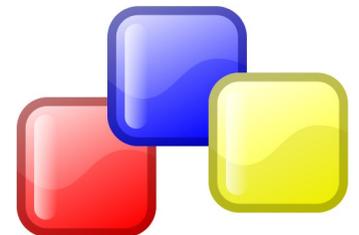
Everything is an Object

- except functions (boxes, clipboard)
- create them with `->new()`, use methods



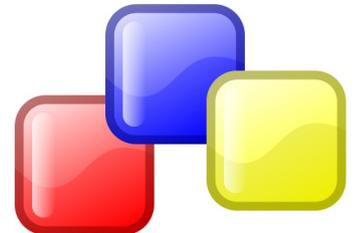
Everything is an Object

- except functions (boxes, clipboard)
- create them with `->new()`, use methods
- objects within objects (know hierarchy well !)



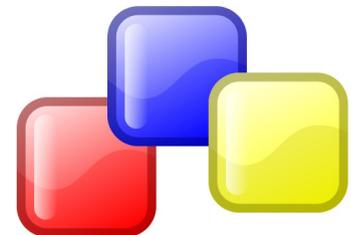
Everything is an Object

- except functions (boxes, clipboard)
- create them with `->new()`, use methods
- objects within objects (know hierarchy well !)
 - crucial for layout and event handling

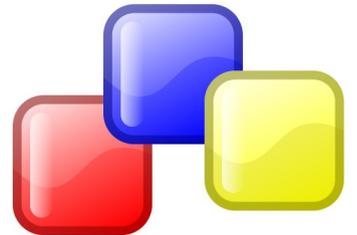


Everything is an Object

- except functions (boxes, clipboard)
- create them with `->new()`, use methods
- objects within objects (know hierarchy well !)
 - crucial for layout and event handling
 - some shortcuts in WxPerl

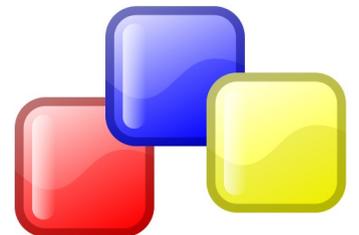


Positional Parameters



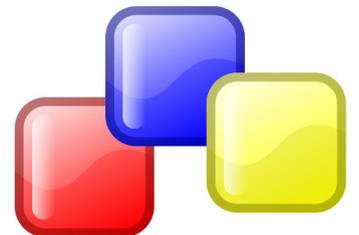
Positional Parameters

- position, not a name, defines meaning of data



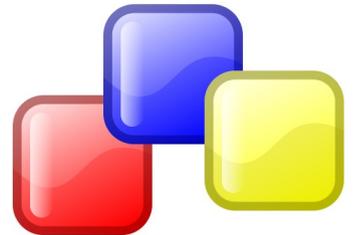
Positional Parameters

- position, not a name, defines meaning of data
- wxPerl::Constructors
 - effort to get named parameters



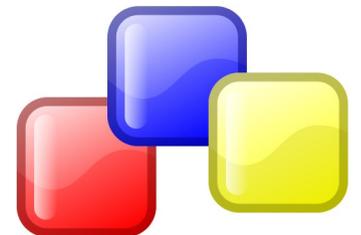
Positional Parameters

- general pattern for `->new()`:
 - parent, ID, content, pos, size, style, ...



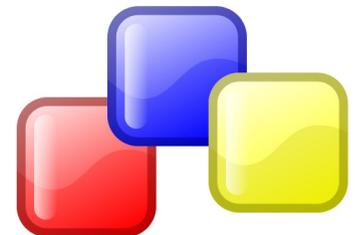
Positional Parameters

- general pattern for `->new()`:
 - parent, ID, content, pos, size, style, ...
- mostly sorted by importance
 - leave out the not needed tail



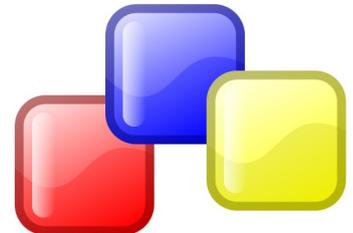
Positional Parameters

- general pattern for `->new()`:
 - parent, ID, content, pos, size, style, ...
- mostly sorted by importance
 - leave out the not needed tail
 - set parameters you can't leave out to `-1`

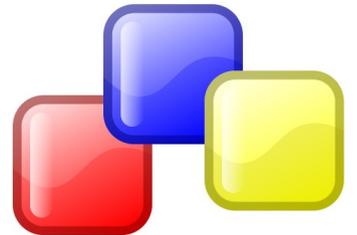


Positional Parameters

- general pattern for `->new()`:
 - parent, ID, content, pos, size, style, ...
- mostly sorted by importance
 - leave out the not needed tail
 - set parameters you can't leave out to `-1`
== default

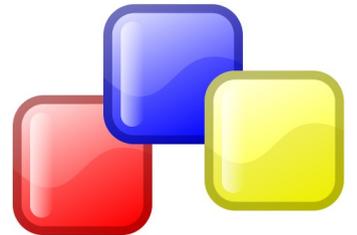


Named Constants



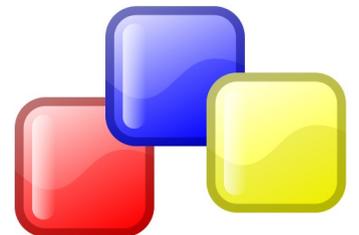
Named Constants

- human readable aliases for numbers



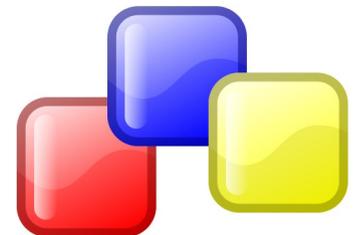
Named Constants

- human readable aliases for numbers
 - like `wxTE_MULTILINE` or `wxTE_READONLY`



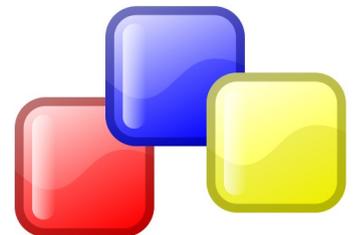
Named Constants

- human readable aliases for numbers
 - like `wxTE_MULTILINE` or `wxTE_READONLY`
 - combined: `wxTE_MULTILINE | ... | ...`



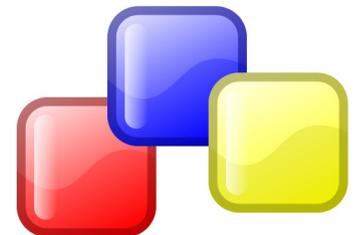
Named Constants

- human readable aliases for numbers
 - like `wxTE_MULTILINE` or `wxTE_READONLY`
 - combined: `wxTE_MULTILINE | ... | ...`
- portable = survive a wx version upgrade



Named Constants

- human readable aliases for numbers
 - like `wxTE_MULTILINE` or `wxTE_READONLY`
 - combined: `wxTE_MULTILINE | ... | ...`
- portable = survive a wx version upgrade
- mostly in style parameter, define behaviour

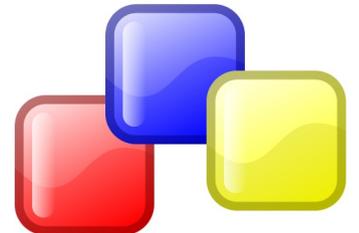


Named Constants

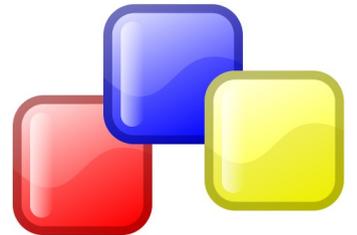
- human readable aliases for numbers
 - like `wxTE_MULTILINE` or `wxTE_READONLY`
 - combined: `wxTE_MULTILINE | ... | ...`
- portable = survive a wx version upgrade
- mostly in style parameter, define behaviour



- have to be imported

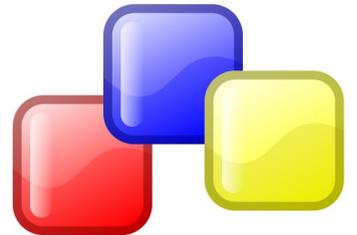


Events



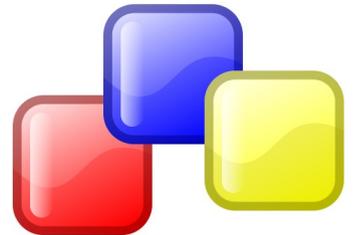
Events

- also imported, but from `Wx::Event`



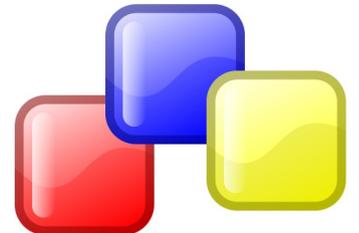
Events

- also imported, but from `Wx::Event`
- uppercase - look like C macros (`EVT_BUTTON`)



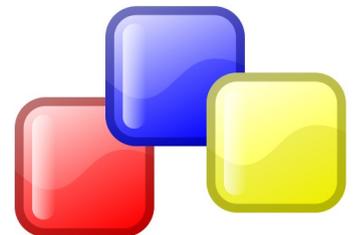
Events

- also imported, but from `Wx::Event`
- uppercase - look like C macros (`EVT_BUTTON`)
 - because there were, not in `WxPerl`

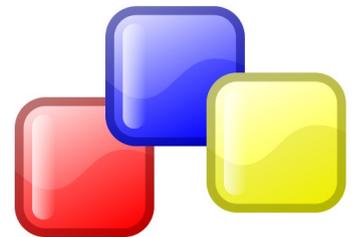


Events

- also imported, but from `Wx::Event`
- uppercase - look like C macros (`EVT_BUTTON`)
 - because there were, not in `WxPerl`
- some widgets listen to up to 20 Events



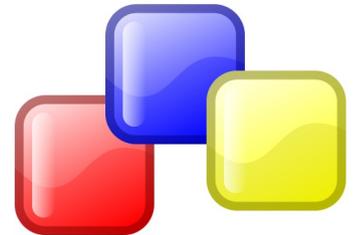
Return Values



Return Values

take care of them

they often define behaviour

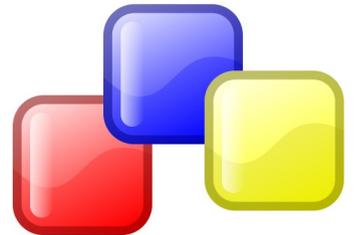


Names

of all classes and methods

are CamelCase

without _



Docs and more

you have all docs offline

+ bonus material

in the „help“ subfolder



let us code ...

