



Cross-Platform Approaches from a Macintosh Perspective

Jonathan Hoyle
AdHoc/MacHack 20
7/29/05

Who am I?

- Jonathan Hoyle
- Just a Mac developer with an opinion
- Made enough bad mistakes in cross-platform projects to learn a little bit
- Currently working for *Eastman Kodak*
- Not representing Kodak, just me
- My vacation time here at MacHack

What this *is* / What this *isn't*

- **Is:**

- A survey of cross-platform frameworks (zoology not biology)
- Overview on using REALbasic with C/C++
- Highly opinionated rantings by the author

- **Isn't:**

- An in-depth tutorial on every framework
- An complete introduction on REALbasic

**Although not coverage of every approach,
after 18 pages it should feel like it is.**



Paper Outline

1. Motivation
2. A Word About Java
3. Development Considerations
4. Legacy Cross-Platform Frameworks
5. Modern Cross-Platform Frameworks
6. REALbasic with C/C++
7. 5 Rules for a Cross-Platform Project
8. Summary

1. Motivation

- Mac has a smaller user base
- Difficult to justify separate development efforts for a small market gain
- Many cross-platform approaches
- Not all are Macintosh “friendly”
- Focus on development for:
 - Mac OS X
 - Windows
 - Classic & Linux (if available)

2. A Word About Java

- Great cross-platform environment, but
 - Two Javas: Language front-end/bytecode back-end. Not always the same
 - Performance penalties due to JVM
 - Java's future? (Microsoft vs. Sun)
 - "Lowest common denominator" look & feel
 - Many Java apps are Windows-only
 - Java not exclusive with C++: JNI
 - Recommended compilers:
 - **Free:** *Eclipse*
 - **Paid:** *Idea* from IntelliJ

3. Development Considerations

- a. C/C++ Compilers
- b. Mac OS X on Intel
- c. Architecting with MVC
(Model-View-Controller)

3a. C/C++ Compilers

- Essentially two choices:
 - Metrowerks CodeWarrior
 - Xcode 2.1



vs.



The best compiler for the future?

3a. C++: CodeWarrior



- Dominant for over 10 years
- 90% of shipping Mac apps
- Mac & Win compilers (v9.4)
- Supports Classic & OS X
- Better ANSI compliance (until gcc 4)
- Arguably much better user interface
- Faster compiler, more optimal builds

but... Future very much in doubt

Bash Metrowerks session

Friday 3PM Venice Room

3a. C++: Xcode 2.1



- Ships free with Mac OS X
- gcc-based
- Improving ANSI compliance
- Universal Binaries
- G5 optimizations and 64-bit compilation
- Improved UI with multiple workspaces
- Distributed Builds, Fix & Continue, etc.
 - Mac OS X-only
 - Apple supported

3a. Best Compiler?

- CodeWarrior's twin compilers was ideal, **BUT**
- Metrowerks sold off x86 compiler in '05
- No Mac commitment since v9.0 in '03
- Already written off by most Mac developers
- Xcode has complete support from Apple
- Only Xcode supports Universal Binaries
- Xcode supports G5, 64-bit, forward thinking
- Apple needs to improve Xcode's GUI more

FINAL ANALYSIS: If CodeWarrior does not support Universal Binaries by 2006, developers will have *no choice* but to choose Xcode.

3b. Mac OS X on Intel

- Xcode's simple checkbox makes it easy
- Be careful about byte-swapping
- Most frameworks will support Intel:
 - CPLAT
 - wxWidgets
 - Qt
- Other IDE's will build Universal Binaries
 - Xcode
 - REALbasic
 - *CodeWarrior?*

Supporting Intel Mac's: Jonathan Johnson

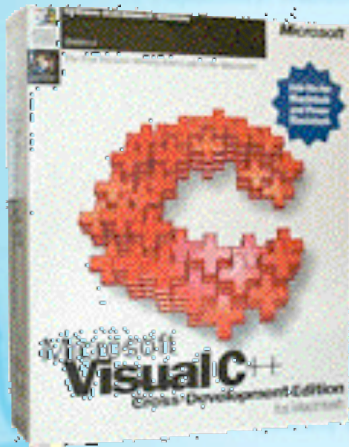
3c. Architecting with MVC

- Separate coding of application:
 - *Model*: Core data and business logic
 - *View*: User Interface
 - *Controller*: connects Model & View
- An MVC architected application does not require a x-platform framework
- Model can be written in standard C++
 - View can the be separate, eg:
 - Interface Builder on Mac
 - Visual C# on Windows

4. Legacy X-Platform Frameworks

- a. Visual C++ Cross-Compiler
- b. Yellow Box for Windows
- c. Mac2Win
- d. PowerPlant for Windows

4a. Visual C++ Cross-Compiler



- Ported MFC apps to Mac
- Windows NT-hosted
- Add-on to VC++ compiler
- 68K first, PowerPC with v4.2
- Obscenely expensive: \$1999 (just for the add-on, \$495 VC++ not included)
- Built notoriously slow & clunky apps
- Discontinued in 1996. Remaining inventory slashed to \$199.

4b. Yellow Box for Windows



- Part of Rhapsody
- NeXTStep API (known today as Cocoa API)
- ProjectBuilder allowed the building of Intel-based applications
- Ran on Rhapsody x86 or Windows
 - Windows runtime: \$249 per PC
 - Killed with the advent of Mac OS X

4c. Mac2Win



- Libraries emulating Mac Toolbox
- ~80% Mac API's ported
- Very expensive, royalty-based
- Used to create many Windows ports:
 - *Metrowerks CodeWarrior*
 - *Claris Works*
 - *Macromedia Director*
- Latest versions Carbonized, but barely:
 - No Carbon Events
 - No ultra-modern calls

4d. PowerPlant for Windows



- PowerPlant: the most widely used framework on the Mac
- Using Latitude, created a Windows version in 2001/2002
- Embraced by Adobe
- Outrageously Expensive:
 - \$15,000
 - plus 1% royalty on sales > \$1.5M
 - capping at \$150,000
- Killed in early 2004



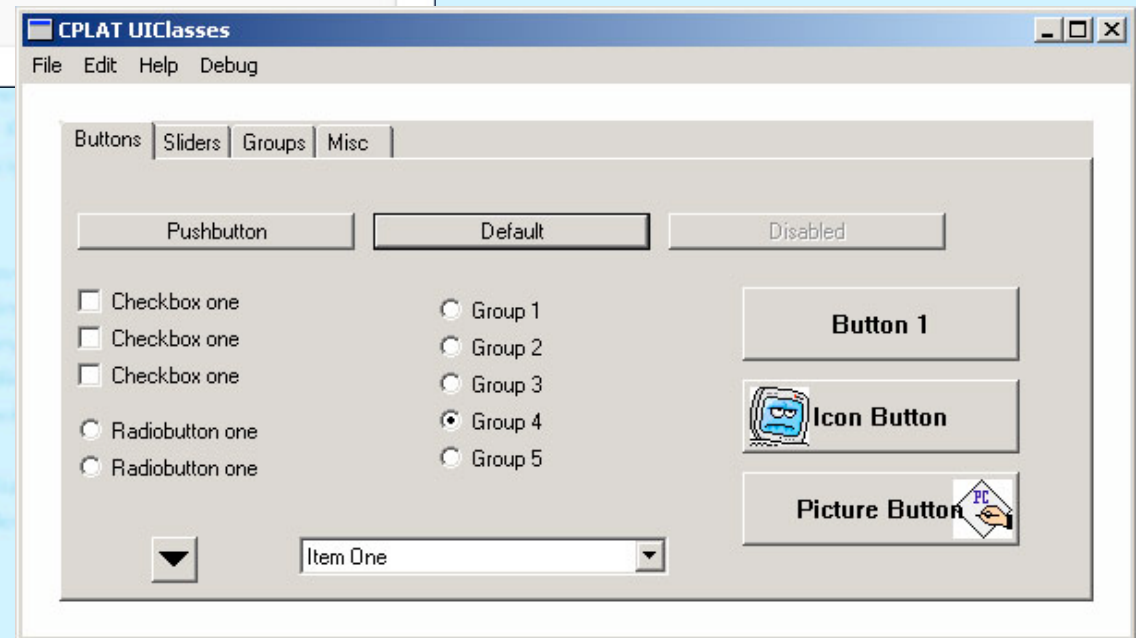
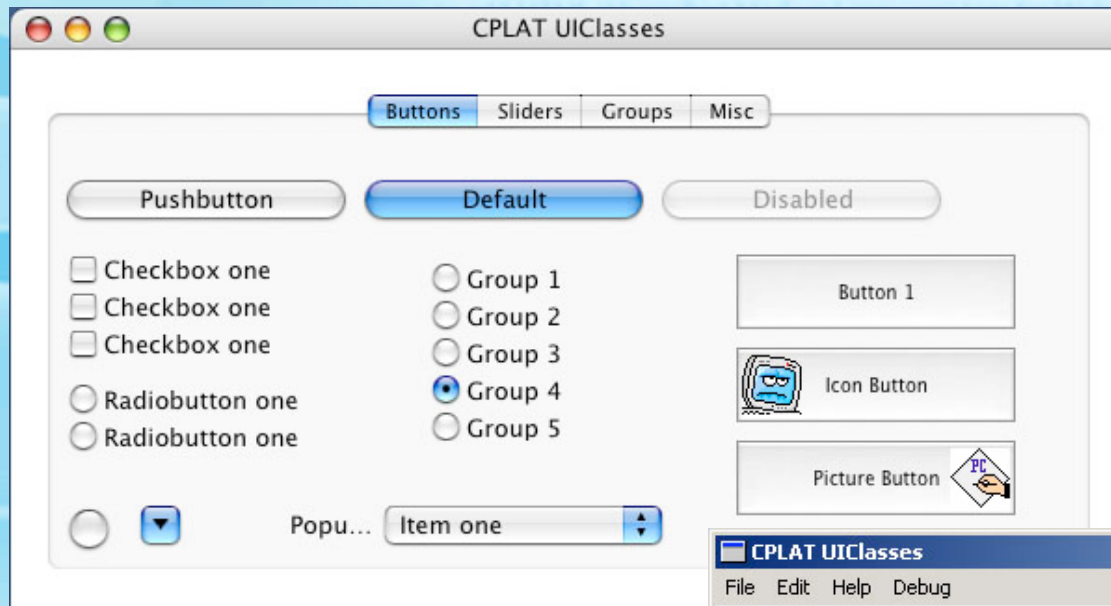
5. Modern X-Platform Frameworks

- a. CPLAT
- b. wxWidgets (formerly wxWindows)
- c. Qt
- d. Other Cross-Platform Frameworks

5a. CPLAT by kSoft

- Price: \$50 per developer (no royalties)
- Mac OS X, Classic, Windows (Linux soon)
- CodeWarrior (Mac & Win), Xcode, Visual C++
- Mac target is a first class citizen
- Amazing work by one Ken Stahlman
- Reminiscent of PowerPlant
- Can convert .nib files into XML for GUI
 - Very comfortable, most Mac-like feel of the frameworks

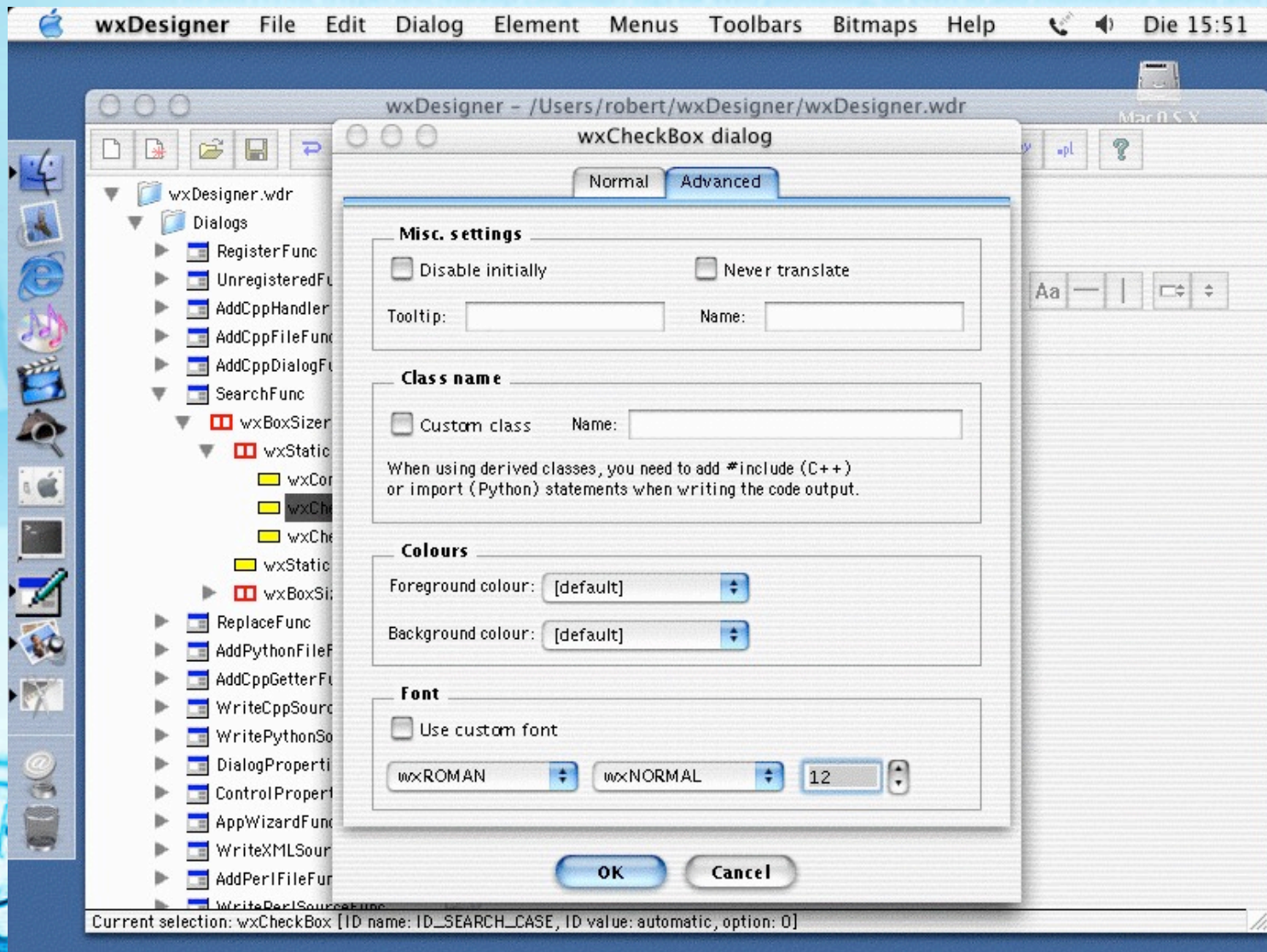
5a. CPLAT screenshots



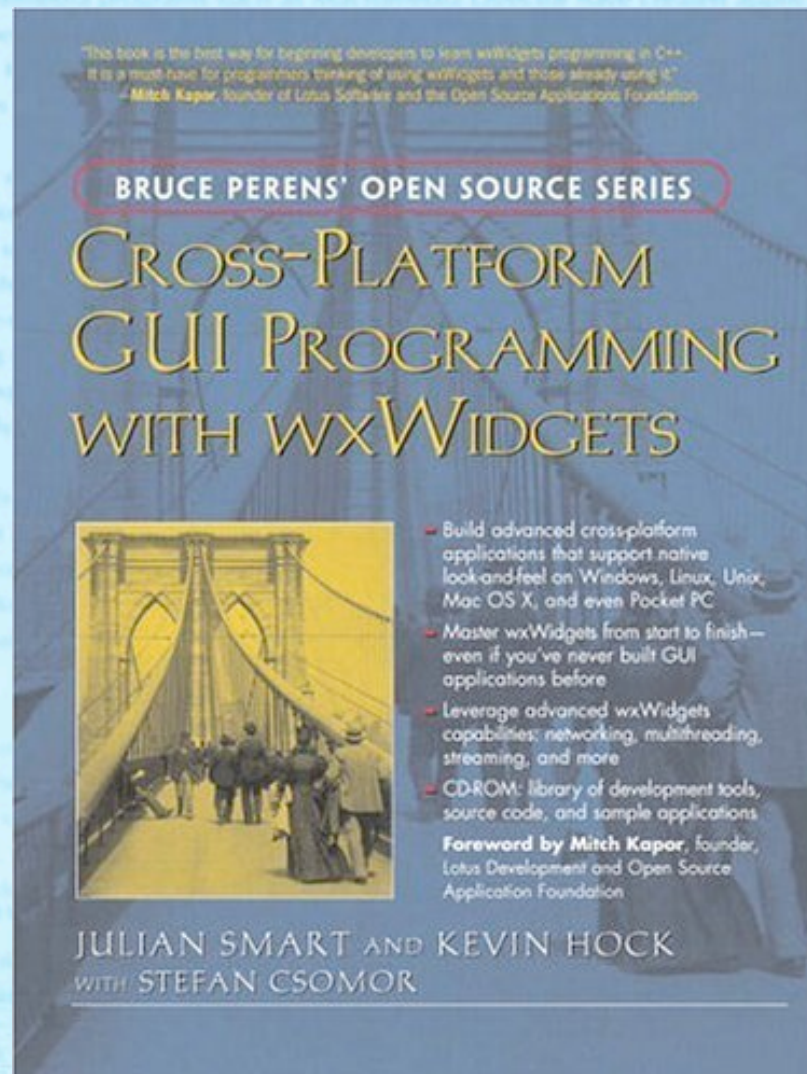
5b. wxWidgets (formerly wxWindows)

- Open Source, free no royalties, no restrictions
- Mac OS X, Classic, Windows, Linux, others
- CodeWarrior (Mac only), Xcode, VC++, others
- Reminiscent of MFC (awkward Mac feel)
- Improving with Open Source community
- Many apps, including *AOL Communicator*
- GUI design tools: *wxDesigner* & *DialogBlocks*
- Does not integrate with InterfaceBuilder
 - Bounties for bugs needing fixing
 - Best free framework for general dev

5b. wxDesigner



5b. wxWidgets Further Reading

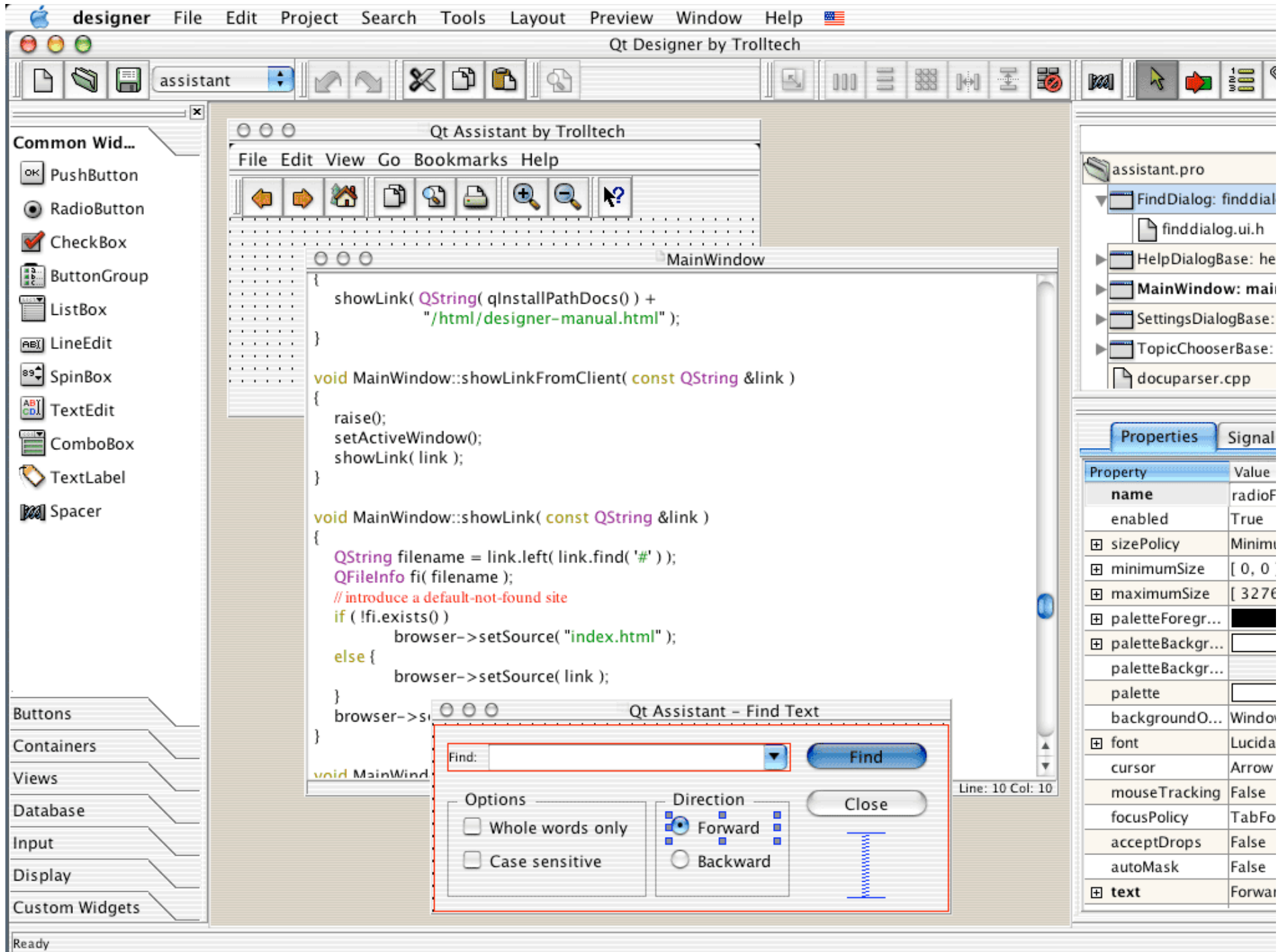


5c. Qt by Trolltech

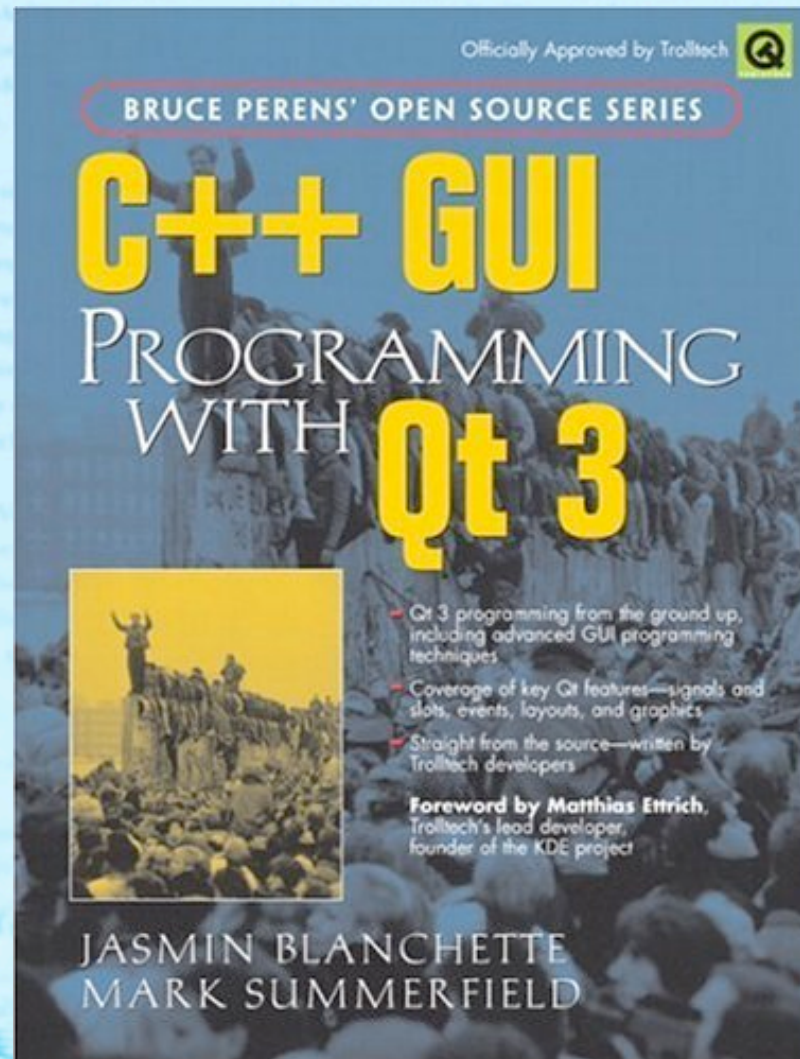
- Tiered pricing: \$1790 Pro license, \$2880 Enterprise, free for Open Source
- Mac OS X, Windows, Linux
- Xcode, Visual C++, gcc
- Very sophisticated, 400 C++ classes
- Many Mac apps: *KOffice* & *PostgreSQL*
- RAD tools: *QtDesigner* and others

Cross-Platform Development with Qt

Scott Collins, Friday 2PM Venice



5c. Qt Further Reading



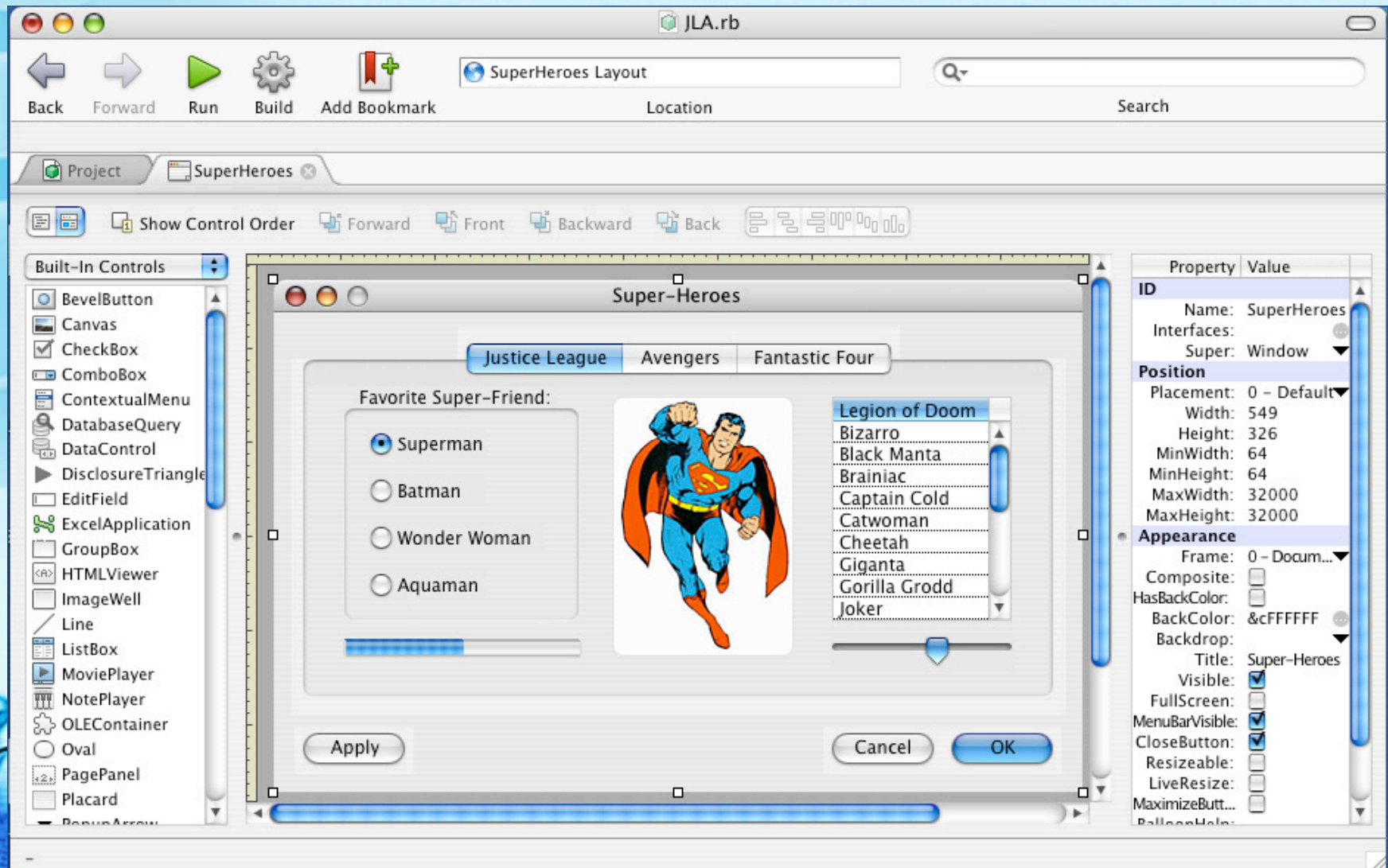
5d. Other X-Platform Frameworks

- **CroPL II** (*Cross-Platform Library*)
- **YAAF** (*Yet Another Application Framework*)
- **FLTK** (*Fast Light Toolkit*)
- **Whisper**
- **ZooLib**

6. REALbasic with C/C++

- A cross-platform “Interface Builder”
- Similar to Visual Basic
- Integrates with C++ code
- Using MVC architecture, RB can generate the GUI and C++ the core:
 - a. Creating the REALbasic GUI
 - b. Creating the C++ Library
 - c. Example: C++ Code
 - d. Example: REALbasic Code

6a. Creating REALbasic GUI



6a. REALbasic GUI (continued)

For more information, download the demo available at: <http://www.realbasic.com>

Also see:

Write a Cross-Platform Game in Two Hours

by Jonathan Johnson

Saturday 2-4PM

Pompeii 2

6b. Creating the C++ Library

- Dynamic Library types:
 - Mac OS 9/X CFM: *Carbon Shared Library*
 - Mac OS X Mach-O: *dylib*
 - Windows: *DLL*
- Use C wrappers for flexibility:
 - `extern "c"` around functions
 - Standardize, eg: `ClassName_MethodName`
 - CFM & Windows DLL functions need to be `__declspec(dllexport)`

6c. Example: C++ code (1)

```
// Model C++ class  
class MyModel  
{  
  public:  
    MyModel();  
    virtual ~MyModel();  
    void foo(int parm1, double parm2);  
    int bar(const char *parm);  
  protected:  
    :  
};
```


6c. Example: C++ code (2)

```
// Exported C Wrapper declarations
extern "C"
{
    export int MyModel_Create();
    export void MyModel_Destroy(int modelHdl);
    export void MyModel_Foo(int modelHdl, int parm1,
                            double parm2);
    export int MyModel_Bar(int modelHdl,
                            const char *parm);
}

// Export macro
#ifdef __MACH__
    #define export
#else
    #define export __declspec(dllexport)
#endif
```

6c. Example: C++ code (3)

```
// Wrapper function implementations  
int MyModel_Create()  
{ return (int) new MyModel; }  
  
void MyModel_Destroy(int modelHdl)  
{ delete ((MyModel *) modelHdl); }  
  
void MyModel_Foo(int modelHdl, int parm1,  
                double parm2)  
{ ((MyModel *) modelHdl)->foo(parm1, parm2); }  
  
int MyModel_Bar(int modelHdl, const char *parm)  
{ return ((MyModel *) modelHdl)->bar(parm); }
```


6d. Example: REALbasic (1)

```
// Define the model library name
#if TargetCarbon
    const ModelLib = "MyModel Library"
#endif

#if TargetMachO
    const ModelLib =
        "@executable_path/../../../../libMyModel.dylib"
#endif

#if TargetWin32
    const ModelLib = "MyModel.dll"
#endif

#if TargetLinux
    const ModelLib = "libMyModel.so"
#endif
```

6d. Example: REALbasic (2)

```
// Define the model library name
```

```
Declare Function MyModel_Create lib ModelLib
```

```
( ) as integer
```

```
Declare Sub MyModel_Destroy lib ModelLib
```

```
(modelHdl as integer)
```

```
Declare Sub MyModel_Foo lib ModelLib(modelHdl
```

```
as integer, parm1 as integer, parm2 as double)
```

```
Declare Function MyModel_Bar lib ModelLib
```

```
(modelHdl as integer, parm as Cstring) as integer
```

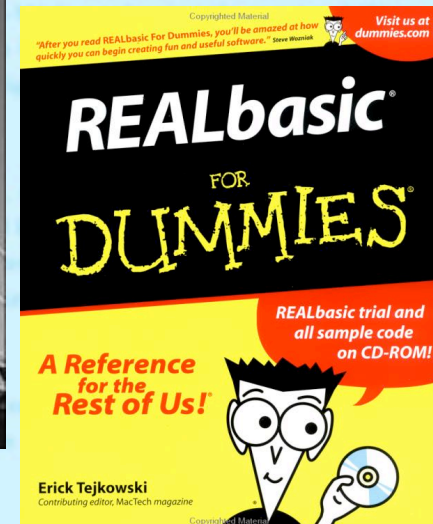
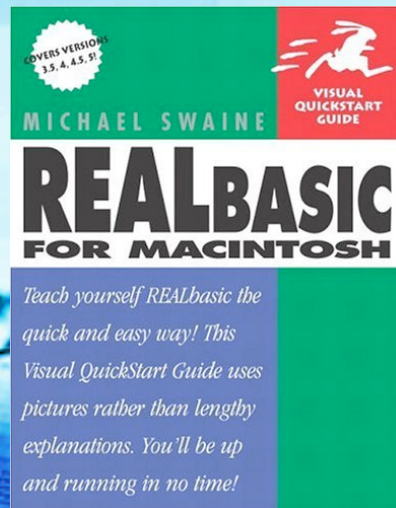
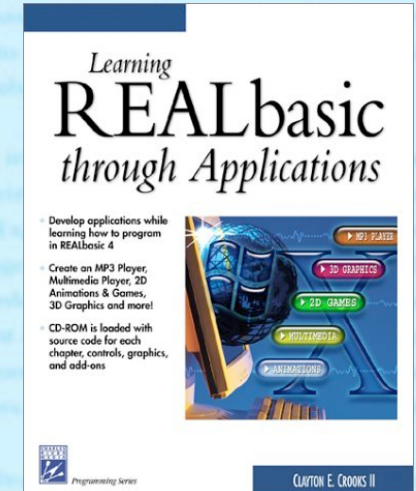
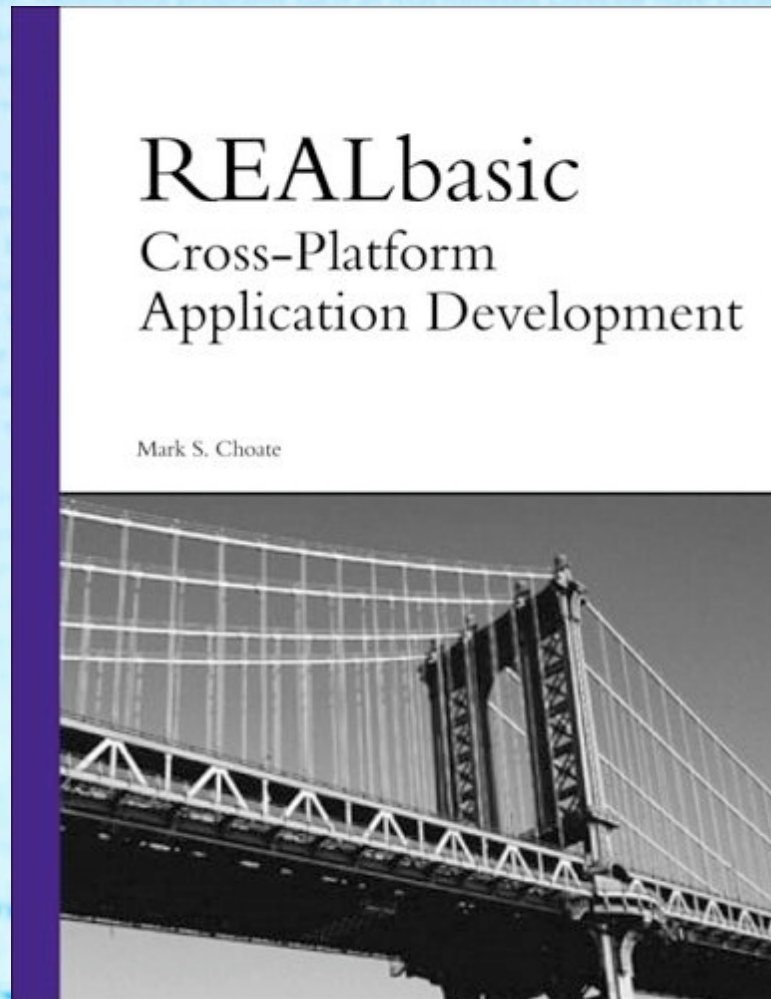
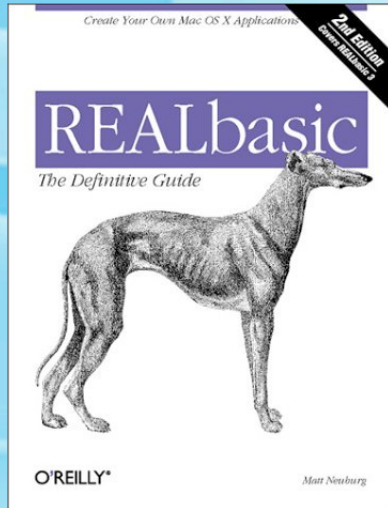

6d. Example: REALbasic (3)

```
// Call the library code
Dim modelHandle as integer
Dim barValue as integer

modelHandle = MyModel_Create()
MyModel_Foo(modelHandle, 12, 3.0)
barValue = MyModel_Bar(modelHandle, "MacHack!")
MyModel_Destroy(modelHandle)

return barValue
```

6. REALbasic Further Reading



7. 5 Rules for a X-Platform Project

1. Design using MVC architecture
2. Have Mac & Windows developers working together from the start
3. Single shared code branch, using `#ifdef`'s if necessary
4. Be ANSI compliant. Use standardized tools and code, such as STL.
5. Place both a Mac & PC on each developer's desk

8. Summary

- C++ cross-platform frameworks:
 - CPLAT: Mac-friendly, great value for \$50
 - wxWidgets: clumsy, MFC-like, but best for a free, non-restriction development
 - Qt: most powerful, also most expensive, but free for Open Source development
- C++ with REALbasic
 - Architect using MVC
 - Place model code into a C++ DLL
 - View app written in REALbasic



For more information...

Copies of the paper, slides &
sample code:

<http://www.jonhoyle.com/MacHack>