



Java on Mac OS X

Insanely Great or Just Insane?

A presentation to the Los Angeles Java Users Group

April 2, 2002

Craig E. Ward, cew@acm.org

1

Presented to the Los Angeles Java Users Group, LAJUG, on Tuesday, April 2, 2002.

What is different, cool, or special about Java on Apple's Mac OS X?

(FYI: The "X" is a Roman numeral and is pronounced "ten".)

Not intended as a complete introduction to Java or Mac OS X. See Resource slide for more information.

Introduction

- *“I’m a back-end guy...”*
- *Adventures in GOO-EE programming*
- *Looking for what’s cool with Java on Mac OS X*

April 2, 2002

Craig E. Ward, cew@acm.org

2

- Where I’m coming from...
- My professional experience is almost exclusively on the back-end. I once wrote a utility that used curses. This means I have no particular expectations of how these GUI programs are supposed to work or look. Other than some simple applets, this is my first exploration of GUI programming and development.
- Looking for what’s cool was part of the request for this presentation.
- This is an adventure!

Agenda

- *Architecture of Mac OS X*
- *Integration of Java 2*
- *Development Tools*
- *Cocoa Framework*
- *Anatomy of a Cocoa Application*
- *Tool Demonstrations*
- *Sources for the talk and other Resources*
- *Conclusions by the presenter*
- *Discussion*

April 2, 2002

Craig E. Ward, cew@acm.org

3

These are the major topics that constitute this presentation.

See “Conclusions and More Questions” slide for some of what’s not included.

Architecture of Mac OS X



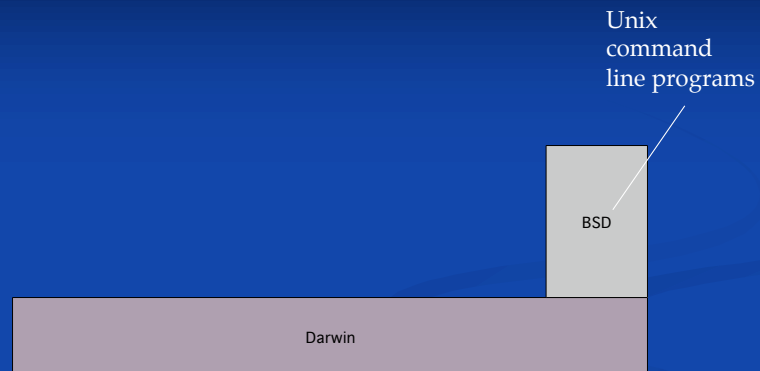
April 2, 2002

Craig E. Ward, cew@acm.org

4

Mac OS X is a modern operating system with (1) Protected Memory; (2) Preemptive Multitasking; (3) Virtual Memory; (4) Symmetric Multiprocessing; (6) BSD Network Stack. It is built on the Mach 3.0 kernel and FreeBSD Unix. Darwin is an “open source” product.

Architecture of Mac OS X



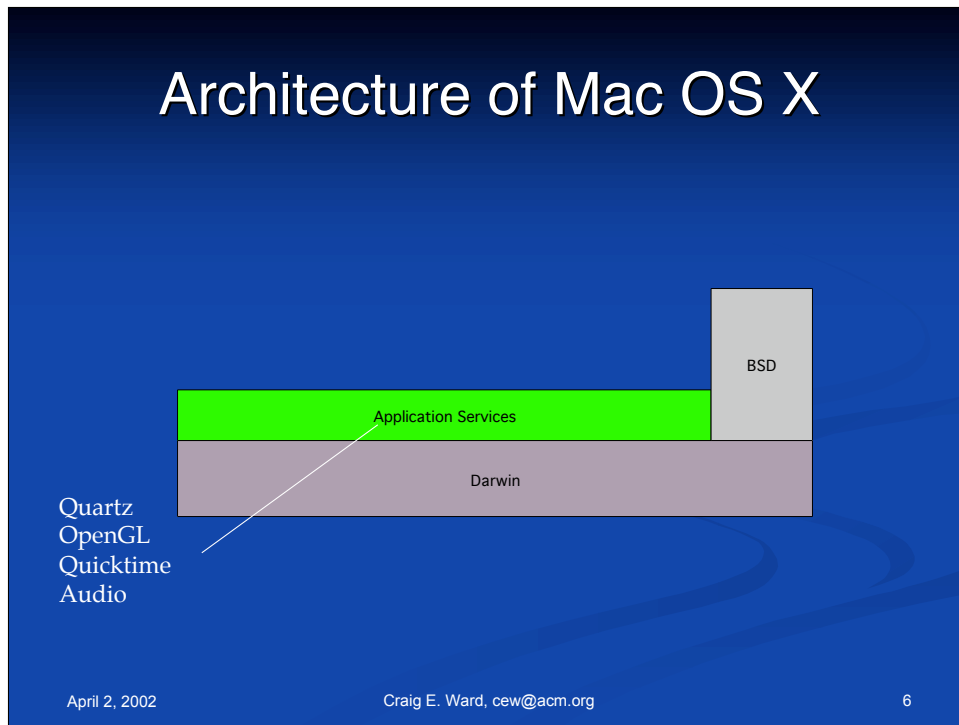
April 2, 2002

Craig E. Ward, cew@acm.org

5

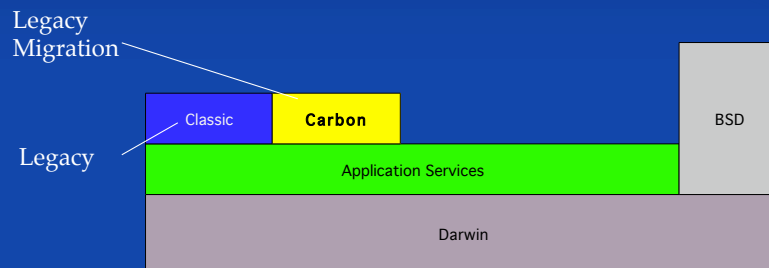
- **Programs running in the BSD subsystem have direct access to Darwin.**
- **This is where “traditional” Unix applications will run.**
- **The BSD environment is considered a secondary environment.**
- **The GNU C/C++ development tools are bundled with Darwin**

Architecture of Mac OS X



- Quartz is based on the Adobe Portable Document Format and is used to draw most of the user interface widgets.
- OpenGL is an open source graphics package.
- QuickTime is an Apple video technology.
- These technologies will not be addressed in any detail here.

Architecture of Mac OS X



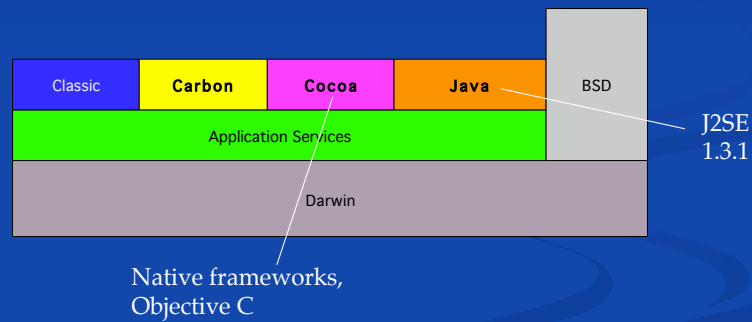
April 2, 2002

Craig E. Ward, cew@acm.org

7

- Classic is the old-style Mac OS. Most legacy Mac programs that do not attempt to access the hardware directly will still work.
- Carbon is the migration environment for classic programs. “Carbonized” programs will run natively on Mac OS X as well as Mac OS 9 and 8 with the Carbon.lib extension.
- Carbon is one of the three primary environments. C++ is the language of Carbon.

Architecture of Mac OS X



April 2, 2002

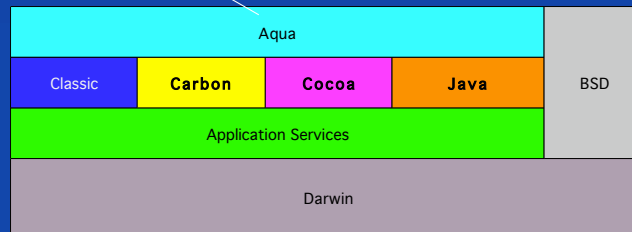
Craig E. Ward, cew@acm.org

8

- Cocoa and Java round out the three primary environments of Mac OS X.
- Cocoa is the native frameworks and is based on OpenStep, formerly known as NextStep. The native language for Cocoa is Objective C.
- Java is Java 2 Standard Edition (J2SE)

Architecture of Mac OS X

User Interface Look & Feel



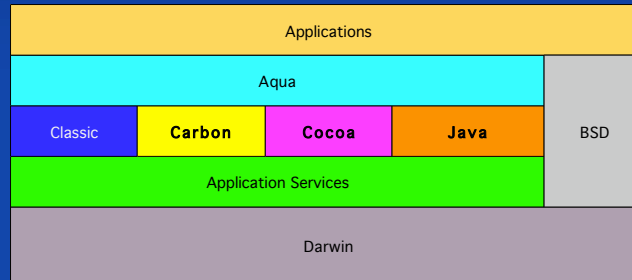
April 2, 2002

Craig E. Ward, cew@acm.org

9

Aqua provides the “look and feel” of Mac OS X. This part is not open source.

Architecture of Mac OS X



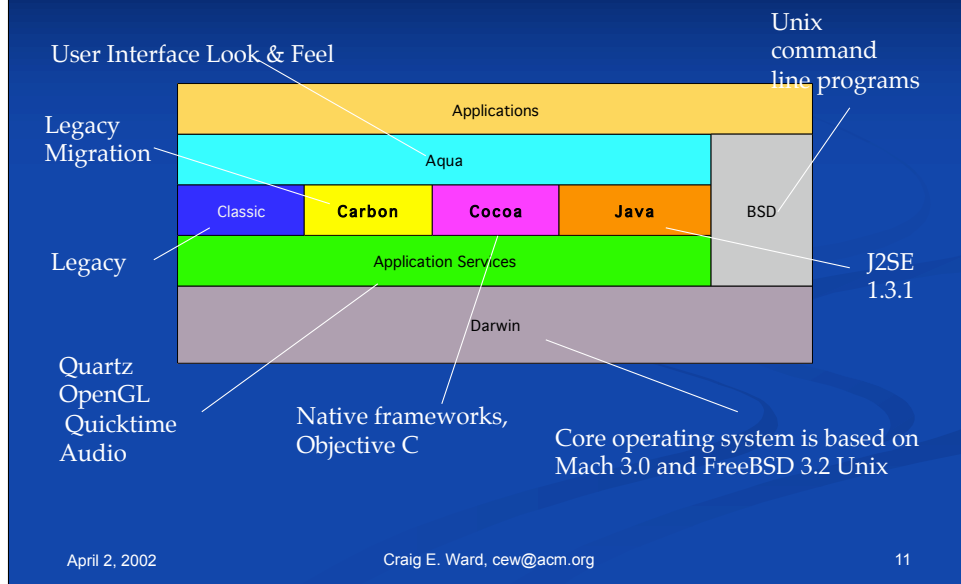
April 2, 2002

Craig E. Ward, cew@acm.org

10

Mac OS X is a modern operating system with (1) Protected Memory; (2) Preemptive Multitasking; (3) Virtual Memory; (4) Symmetric Multiprocessing; (6) BSD Network Stack.

Architecture of Mac OS X



Mac OS X is a modern operating system with (1) Protected Memory; (2) Preemptive Multitasking; (3) Virtual Memory; (4) Symmetric Multiprocessing; (6) BSD Network Stack.

The three primary programming environments are Carbon, Cocoa, and Java. Carbon is intended for porting legacy programs; these will run under either Mac OS. Cocoa is the native framework and is based on OpenStep. Programs running in the BSD subsystem have direct access to Darwin.

Integration of Java 2

- *Pre-installed on all new Macintosh computers*
- *Peer core environment to Cocoa and Carbon*
 - *Mix Java code with Cocoa native code*
- *Automatic Aqua PLAF*
- *Optimized HotSpot VM*
- *Almost latest releases (1.3.1)*

April 2, 2002

Craig E. Ward, cew@acm.org

12

- Java is placed as an equal peer to the native Cocoa environment.
- Cocoa and Java code may be mixed
- Apple has provided wrapper Java classes for most of the Cocoa classes allowing all (if not “pure”) Java coded programs to access Aqua objects.
- The Java configuration allows Java programs to have an Aqua-like look and feel without user or programmer effort. (Pluggable Look and Feel)
- Apple plans to keep Java updated as Sun releases new versions of the platform. Java 1.4 should be out real soon now.

Development Tools

- *Standard J2SE command line tools*
- *Many third-party tools*
 - JBuilder
 - CodeWarrior
 - JDEE
- *Apple*
 - MRJAppBuilder
 - Project Builder and Interface Builder
 - JavaBrowser, AppletLauncher, Java Web Start

April 2, 2002

Craig E. Ward, cew@acm.org

13

- The Developer package includes the expected command line tools and they work the same as on other Unix systems.
- Borland's JBuilder 6 is a popular, commercial product. A personal edition is available at no dollar cost. Product does not include all of the features of the Windows version.
- Net beans is another choice.
- CodeWarrior is an IDE that provides C, C++, and Pascal as well as Java. It is still primarily for Classic or Carbon development. The next release is supposed to include Cocoa.
- Darwin includes GNU Emacs and you can use the Java Development Environment for Emacs with it. I rebuilt it with X Windows enabled so it almost looks like the Unix/Windows versions. (Can't get Aqua browsers to read argv...)
- MRJAppBuilder is a tool to wrap class and jar files into an Aqua wrapper to make them resemble native applications.
- Project Builder and Interface Builder are the primary Apple tools for developing Cocoa and Java programs.
- JavaBrowser is a class and JavaDoc browser. Java Web Start is for Java Network Launch Protocol, JNLP.

Cocoa Frameworks

- **Foundation**
 - *Low-level classes representing data types, files, threads, collections, etc.*
- **AppKit**
 - *User interface components*
- **Java Bridge**
 - *The wrapper classes*

April 2, 2002

Craig E. Ward, cew@acm.org

14

- A framework is basically a library of shared code.
- The Foundation framework is similar in content to java.lang, java.util, java.io, and java.text. The overlap is not exact or one-to-one. Some classes were not bridged like NSString and NSThread were not bridged.
- The Foundation framework also includes classes that allow for application scripting (AppleScript, Open Scripting Architecture (OSA))
- One of the more interesting AppKit objects is NSDrawer
- The “NS” part is found on all class names. Legacy of “NextStep.”

Anatomy of a Cocoa Application

- *Hidden file structure*
- *Property Lists in XML*
- *Other Resources*
 - Graphics
 - NIB files

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist SYSTEM "file://localhost/System/Library/DTDs/PropertyList.dtd">
<plist version="0.9">
<dict>
  <key>CFBundleDevelopmentRegion</key>
  <string>English</string>
```

April 2, 2002

Craig E. Ward, cew@acm.org

15

The primary “shell” is the Finder, modeled after the traditional Mac OS UI. It is configured to hide file structure of applications. These configuration parameters are stored in .plist, “property list” files in an XML format. These can be edited by any text editor or with the PropertyListEditor in the developer package.

The last text box illustrates a property list.

More about the hidden file structure will be covered during the demonstrations.

NIB Files

- Serialized instances of interface objects
- Important Concepts
 - Action
 - Outlet

April 2, 2002

Craig E. Ward, cew@acm.org

16

- Action: A method that takes a single argument of type Object.
- Outlet: A place to hold a reference to another entity
- These are used in Interface Builder to link actions to objects on the screen, message passing
- We will see the creation and manipulation of NIB files in the demonstration

Tool Demonstrations

- *MRJAppBuilder*
- *Project Builder and Interface Builder*
- *JavaBrowser*

April 2, 2002

Craig E. Ward, cew@acm.org

17

- MRJAppBuilder wraps Java class and jar files to make them appear more like standard Mac OS X programs.
- Project Builder and Interface Builder are the primary IDE tools.
- JavaBrowser can be used to view JavaDoc and source files.

Sources & Resources

- Apple Developer Connection
 - Free web membership
 - Inside Mac OS X: Java Development on Mac OS X
 - Inside Cocoa: Developing Cocoa Java Applications: A Tutorial
 - List java-dev@lists.apple.com
- Early Adopter Mac OS X Java
- Mac OS X: The Complete Reference

April 2, 2002

Craig E. Ward, cew@acm.org

18

The URL for the Apple Developer Connection is <http://developer.apple.com/>.

The two documents listed are PDF files on the ADC site.

Williams, Murry Todd, et.al. *Early Adopter Mac OS X Java*, Wrox Press Limited, 2001. ISBN 1-861006-11-X

Feiler, Jesse. *Mac OS X: The Complete Reference*. Osborne McGraw-Hill, 2001.

ISBN 0-07-212663-9

Conclusions

- First class Java development platform
- User-friendly runtime
- Cocoa applications have access to Java 2 features and vice-versa
- More than just an educational toy

April 2, 2002

Craig E. Ward, cew@acm.org

19

- Java on Mac OS X compares favorably with the implementations I've used on Windows NT/2000 and Linux (Red Hat 6.0).
- It is easier to deploy Java programs to Mac OS X than to either Windows or Linux.
- The intermixing of Cocoa and Java 2 presents interesting possibilities, but it is also a danger if one is writing "pure" Java.
- Before this investigation, I viewed Java on Mac as just a place where I could run examples from the books I was reading, especially Eckel's Thinking in Java). Now I believe it could be a platform for real work.

Discussion and More Questions?

- What wasn't covered
- Questions and Comments
- More?



April 2, 2002

Craig E. Ward, cew@acm.org

20

This presentation only touches the surface of the Cocoa environment and how it is similar and different from Java 2. Also, much of what comprises J2EE can run on Mac OS X. This presentation didn't address such issues as

- Legacy integration with the old Mac OS; the MRJ only supports Java 1.1.8
- More in depth integration into Mac OS X, e.g. custom icons, binding documents, etc.
- JNI for Mac OS X

Looking into these and Apple's WebObjects technology would be yet another interesting adventure.