

Welcome

To Advance through Presentation
Use Page Up and Page Down Keys



99 | Worldwide
Developers
Conference



99 | Worldwide
Developers
Conference

What's New in Cocoa (Yellow)

Ali Ozer and Chris Kane
Application Frameworks

Today's Topics

- What's new in Mac OS X Server
- What's new in Mac OS X
- Case study



Mac OS X Server



- Threading
- URL Resource Loading
- Sound, MovieView
- Typesetter
- TextAttachmentCell
- Standard About Box
- Exception Handling
- ActiveX



Mac OS X Server

- Scripting
 - Allow Yellow applications to be scriptable
 - Script the model of the application
 - Especially easy if your model is clearly defined
 - Driven from the Blue Box
 - Document and Text objects scriptable out of the box



Mac OS X Server

- Yellow/Java
 - Robust, complete APIs
 - Many Yellow/Java applications on the system

SetupAssistant

Sketch

NetworkManagerCalculator

TextEdit

IconBuilder

PrintManager

BlastApp



Mac OS X

egcs



Mac OS X

- Lots of changes around ~~Yellow~~
 - Core OS
 - Compiler
 - CoreFoundation
 - Carbon
 - Graphics



Core OS

- Pthreads
- Untyped IPC
- No nmserver
 - Currently no cross-host distributed objects



Compiler

- egcs
 - Better C++ support
- Changed macros

ppc

__ppc__

NeXT

__APPLE__



Core Foundation

- Common substrate beneath Carbon, Classic, and Cocoa
- Cross-platform, portable
- C APIs and implementation
- Some object-oriented paradigms



CF Types

- Many concepts from Cocoa
 - String, collections, property lists
 - Preferences
 - Notifications
 - Bundle
 - RunLoop
- And new stuff
 - XML
 - Plug-in



CF to Cocoa Bridging

- Equivalence of NS objects and CF types
 - Can send NSArray methods to CFArray
 - All methods, even categories
 - Can call CFArray functions on NSArray
 - To any NSArray, even subclasses
 - Efficient (no conversions or lookups)

“Toll-free bridging”



CF Bridged Types

- Toll-free bridged classes

NSArray NSObject

NSCharacterSet NSMutableSet

NSData NSString

NSDate NSTimer

NSDictionary NSURL

- Coming soon: NSNumber



CF Cocoa Impact

- Most toll-free bridged classes implemented in terms of their counterparts
- NSUserDefaults implemented in terms of CFPREFERENCES
 - Defaults are now saved as XML files, in `~/Library/Preferences`
 - Auto-conversion on first run



CF Cocoa Impact

- Property List Handling
 - CF property lists
 - XML format
 - Support number, boolean, date
 - Existing Foundation functions do not call through to the CF functions
 - But the functionality will be exposed





99 | Worldwide
Developers
Conference

Demo

Carbon

- Need to assure the two toolkits work well and efficiently together
 - Common services
 - CoreFoundation
 - CarbonCore
 - AppleEvents
 - Quartz



Carbon

- Need to enable developers to use functionality from Carbon
 - Can access Carbon frameworks
 - Can render QuickDraw in NSViews
 - Toll-free bridging helps with CF types
- Other direction also possible
 - Mach-O Carbon applications can use Cocoa frameworks



Carbon

- Need seamless user experience between Carbon and Cocoa applications
 - Copy/paste
 - Drag & drop
 - AppleEvents, AppleScript
 - Same application packaging model



Carbon

- Need seamless user experience between Carbon and Cocoa applications
 - Same behavior of controls
 - Same windowing and activation behavior
 - Shared common panels
 - Shared input managers





99 | Worldwide
Developers
Conference

Demo

Graphics

- No Display PostScript
- Quartz: New client side library for 2D graphics
 - PostScript imaging model
 - Compositing, transparency, effects
 - Anti-aliased rendering
- Lightweight Windowserver (LWWS): Server for window management



Graphics: What's Out

- PSxxx() functions no longer work
 - Some are still emulated
 - But not for long!
- Custom wraps (.psw) no longer work
 - Not emulated
- NSDPSTexture, NSDPSTextureServerContext
 - Most operations not necessary anymore



Graphics: What's In

- NSView drawing mechanisms the same
 - lockFocus, drawRect, display
- NSBezierPath
 - Create, stroke, and fill paths
- NSAffineTransform
 - Manipulate transformation matrices



Graphics: What's In

- NSImage, NSImageRep, and subclasses
- NSColor
- NSEPSImageRep
 - No on-screen PostScript rendering
 - Will be able to print to PostScript
- NSGraphics.h
 - NSRectFill, NSRectClipList, etc.



Graphics: What's In

- `NSTextView`
 - `NSStringText` finally removed...
- `NSString`, `NSAttributedString` drawing
- `NSFont`
 - TrueType fonts
 - No PostScript fonts for now
 - Coming in a future release



Graphics: What's In

- NSGraphicsContext
 - Replaces NSDPSContext
 - One per NSWindow
 - One per each additional drawing thread
 - Methods for saving/restoring graphics state
 - saveGraphicsState
 - restoreGraphicsState



Graphics: What's In

- Possible to use Quartz directly

CGMoveTo(CGSContextRef, float, float)

CGSetLineWidth(CGSContextRef, float)

CGStroke(CGSContextRef)

CGRotateCTM(CGSContextRef, float)



Graphics: What's New

- PDF support
 - PDF generation
 - From print panel
 - From NSView
 - PDF import
 - NSPDFImageRep
 - Imaging only
- Scalable Vector Graphics (SVG)
coming soon



Graphics: What's New

- New Printing Architecture
 - New print and page layout panels
 - Ability to query printer for features
 - Job ticket





99 | Worldwide
Developers
Conference

Demo



99 | Worldwide
Developers
Conference

Case Study: Create

Andrew Stone

andrew@stone.com

www.stone.com

Background

- Create was written over last 10 years
- Intensely used Display PostScript
- Was ported through 13 versions of NextStep, OpenStep, Rhapsody, and OS X
- Was crufty and showing age



The Problem

- Display PostScript is going away
- Many new features in Cocoa
- Create needed basic overhaul



The Solution

- Do a re-architecture using:
 - NSBezierPath and NSAffineTransform
 - NSDocument
 - NSImage
 - Use built-in JPEG, GIF, and PNG features
- Reuse algorithms and data structures



The Results

- NSBezierPath and NSAffineTransform
 - Completely unified imaging model
- NSDocument
 - Could ditch homerolled undo
 - Replaced bunches of other code
 - AppleScript for almost free
- NSImage
 - Ditch JPEG and GIF libraries

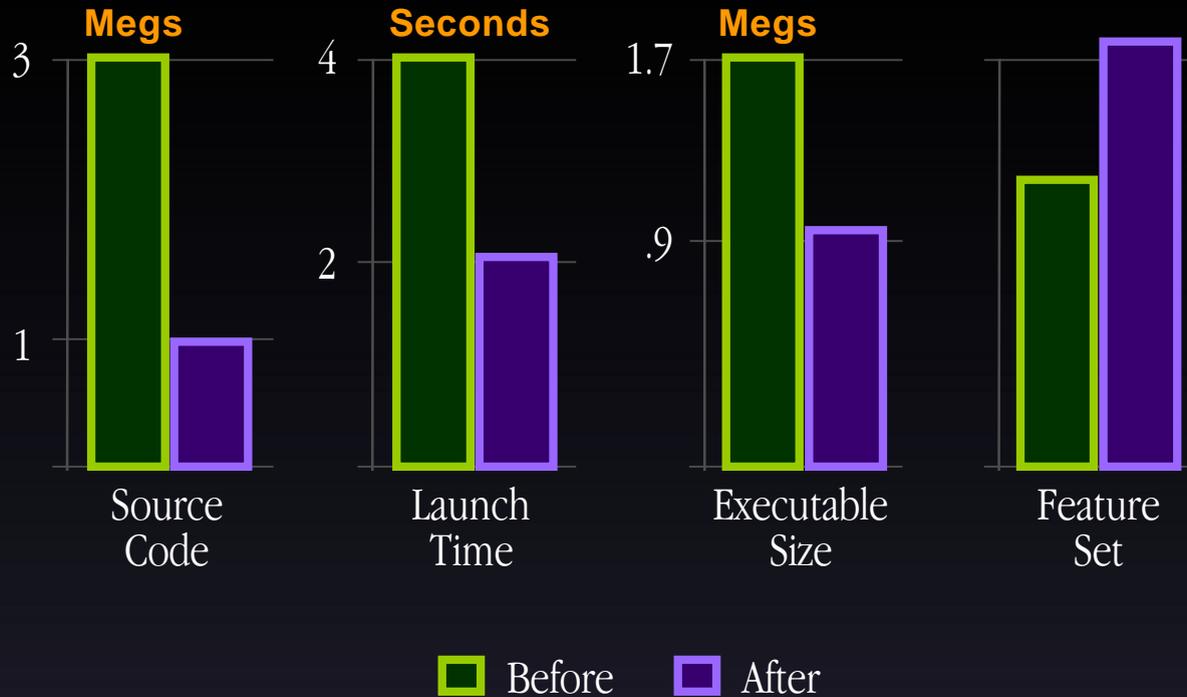


The Results

- A much smaller, faster application with much less code to maintain and a solid architecture to easily add new features
- Work done in about 60 days



The Results



Conclusion

- Although it seems like a daunting task, re-architecting can greatly improve your application and provide many benefits, as well as a solid basis for future development





Demo

99 | Worldwide
Developers
Conference

Stone ♦ Design



Objective-C APIs

- At some point, the center of gravity will shift over the Java APIs
 - Some new functionality might be available in Java only
- We intend to continue to make the Objective-C APIs available in Mac OS X
- We encourage you to use the Java APIs for Cocoa for new development



Windows

- Currently looking at possible business models
- WebObjects development and deployment on Windows still ongoing



Items of Note

- Not functional in Developer Preview 1
 - AWT
 - NSSound
 - NSMovieView
 - AppleScript



Other Items of Note

- NSPasteboard
 - Uses XML serialization
- Applications launched from telnet
 - ...can't find pasteboard server
- Miscellaneous UI nits
 - Dragging
 - Window manipulations
- Read the release notes!



Summary

- Many changes from Mac OS X Server to Mac OS X
 - Cocoa isolates you from most
- Cocoa and Carbon are the two technologies to develop great Mac OS X applications
- Cocoa applications enjoy proven object-oriented benefits





99 | Worldwide
Developers
Conference

Q&A



Think different.TM



Welcome

To Advance through Presentation
Use Page Up and Page Down Keys



99 | Worldwide
Developers
Conference