

Welcome

To Advance through Presentation  
Use Page Up and Page Down Keys



99 | Worldwide  
Developers  
Conference



99 | Worldwide  
Developers  
Conference

# Cocoa (Yellow)— An Overview

Becky Willrich

Application Frameworks  
Engineer

# Overview

- What is Cocoa?
- Foundation in detail
- AppKit in detail

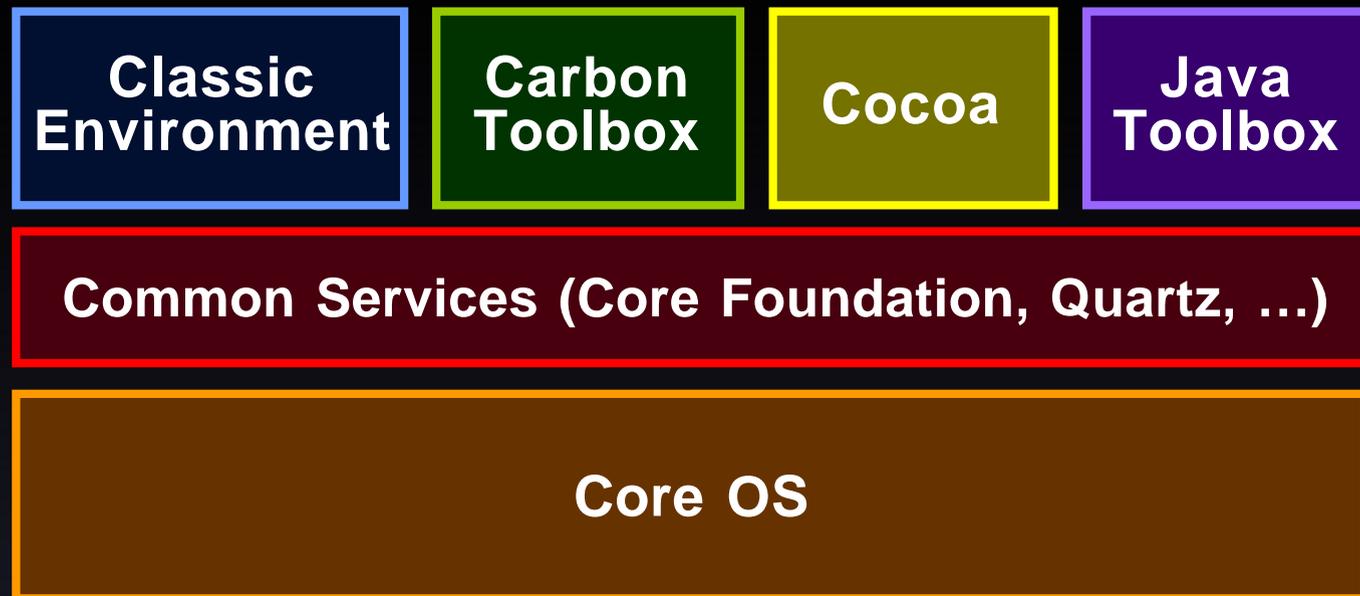


# What Is Cocoa?

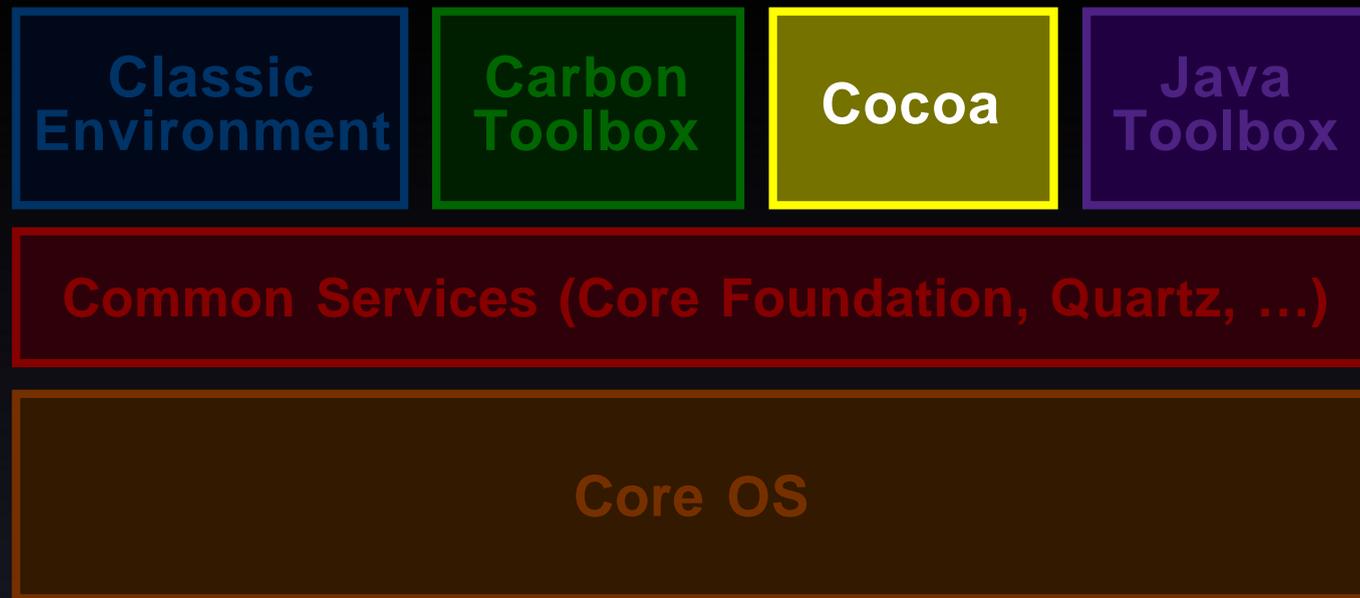
- Full-featured frameworks for application development
- Fully object-oriented
  - APIs in Java and Objective C
- Cross-platform (Mac OS X and Windows)
- Basis for technologies like WebObjects and EOF



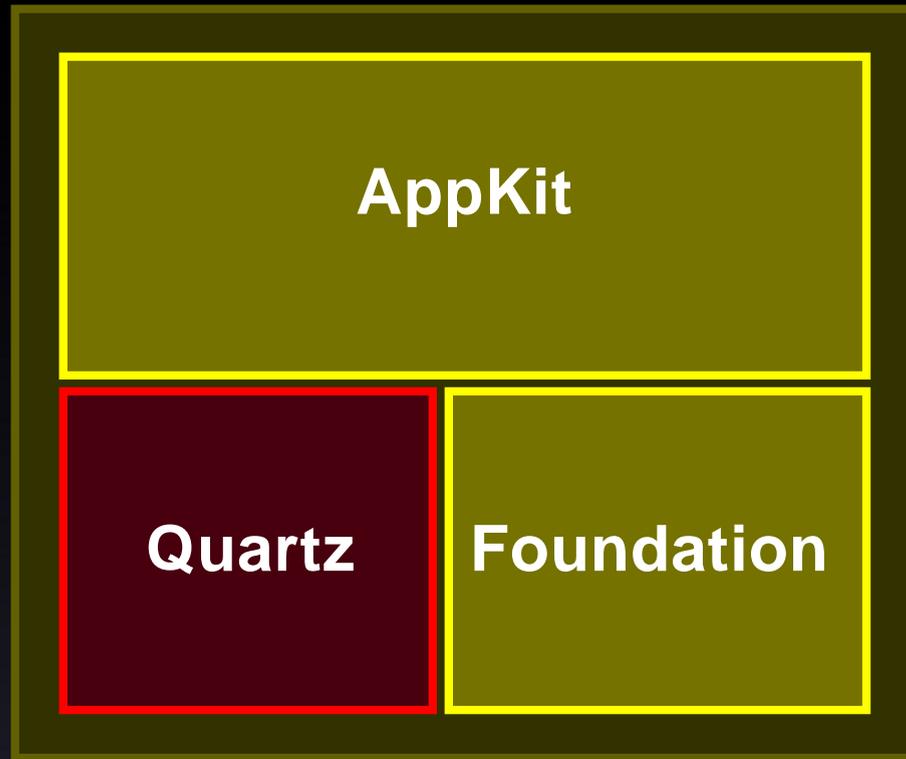
# Where Does Cocoa Fit?



# Where Does Cocoa Fit?



# Where Does Cocoa Fit?



# Foundation in Detail

- Provides basic object abstractions for storage and collections
- Insulates clients from operating system specifics
- Supports internationalization and localization



# Storage Classes

- Most classes have a mutable and immutable variety
- NSValue, NSString, and NSData store values directly
- NSArray, NSDictionary, etc. store collections of objects
- Can be simply written to file as *property lists*



# Property Lists

- Plain text format for simple object graphs
- XML format available via CoreFoundation
- NSString, NSData, NSDictionary, NSArray
- Also NSNumber and NSDate for XML
- Used by NSUserDefaults to store user preferences



# OS Insulation

- File/resource access (NSFileManager, NSURL, NSFileHandle, NSPathUtilities . . .)
- Process management (NSThread, NSProcessInfo)
- Cross-thread and cross-process communication (NSRunLoop, NSDistributedNotificationCenter)
- User preferences (NSUserDefaults)



# Internationalization

- NSString
  - Stores characters as UniChars (i.e. Unicode encoding)
  - Supports conversion to/from different encodings
  - Used throughout the API in place of (char \*)



# Internationalization

- NSBundle
  - Allows resources to be specified and retrieved by language
  - Provides access to localized strings
- NSUserDefaults
  - Defines several localization defaults
  - Stores these on a per-user basis



# AppKit

- Provides the infrastructure for drawing and event handling
- Includes all the standard controls
- Sophisticated text layout, display, and editing
- Standard dialogs (*panels*)
- Relies on Foundation plus Quartz





99 | Worldwide  
Developers  
Conference

# Demo

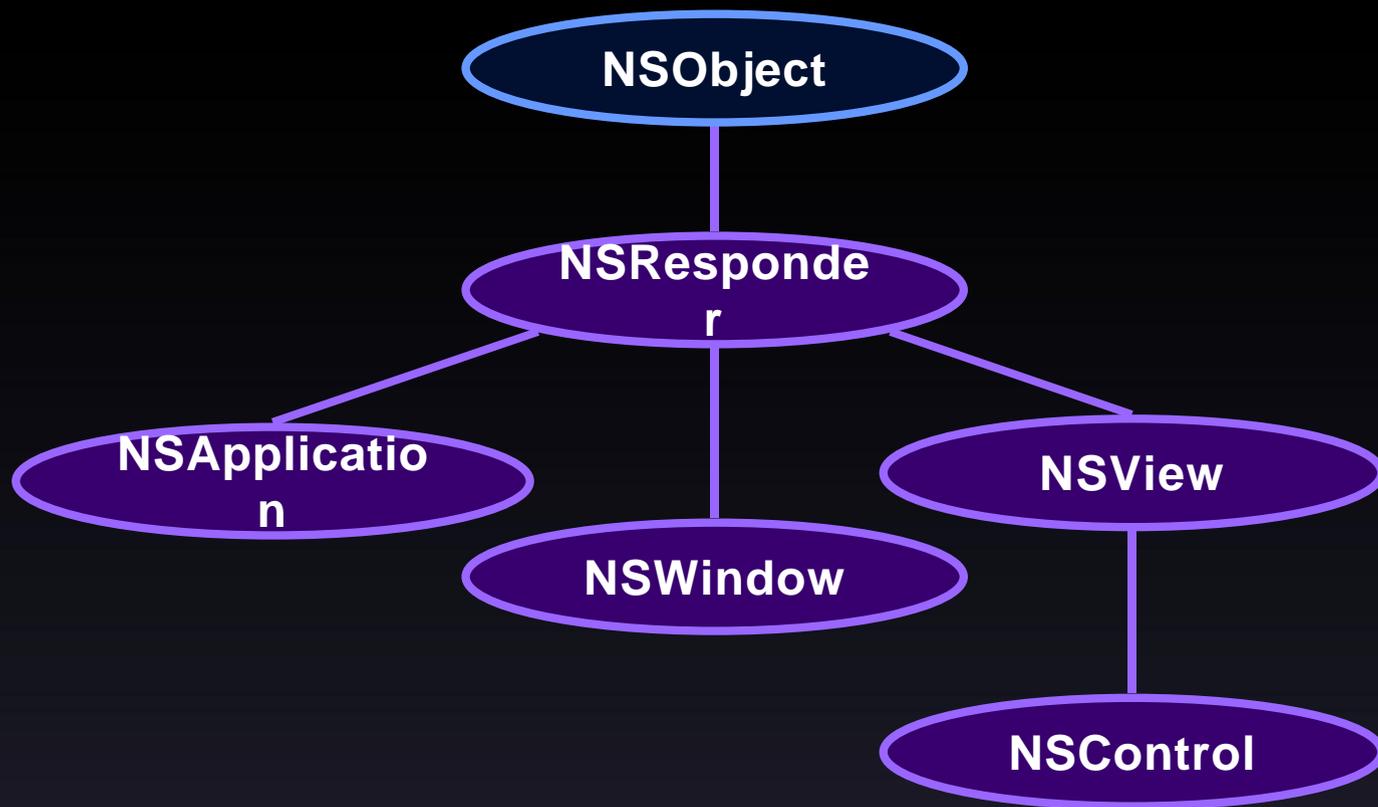
The TextEdit Application

# AppKit in Detail

- Core classes
- Abstractions
- User interface, IB, and NIB files
- Text subsystem



# AppKit—Core Classes



# NSResponder

- NSResponders can respond to events
- Events arrive as methods
  - **keyDown(), mouseMoved(),...**
- Responders have a *next responder*
- Events are passed from responder to responder through the *responder chain*



# NSApplication

- One, shared instance per application
- Manages communications with the rest of the system
- Maintains the list of windows in the app
- Maintains and runs the app *run loop*
  - Receives events from the WindowServer
  - Dispatches those events to its windows



# NSWindow

- NSWindows reserve screen real estate
- Translate raw events in to the appropriate responder method
- Maintains a *first responder*
- Dispatch the events to the “correct” responder
  - Mouse events go to the view under the mouse
  - Key events go to the first responder



# NSView

- NSViews draw to the screen
  - Override **drawRect()** to customize drawing
- Live in a hierarchy within a window
  - One *superview*; many *subviews*
- Provide support for many basic UI notions
  - Drag-and-drop, Keyboard navigation, context-sensitive menus, tool tips...

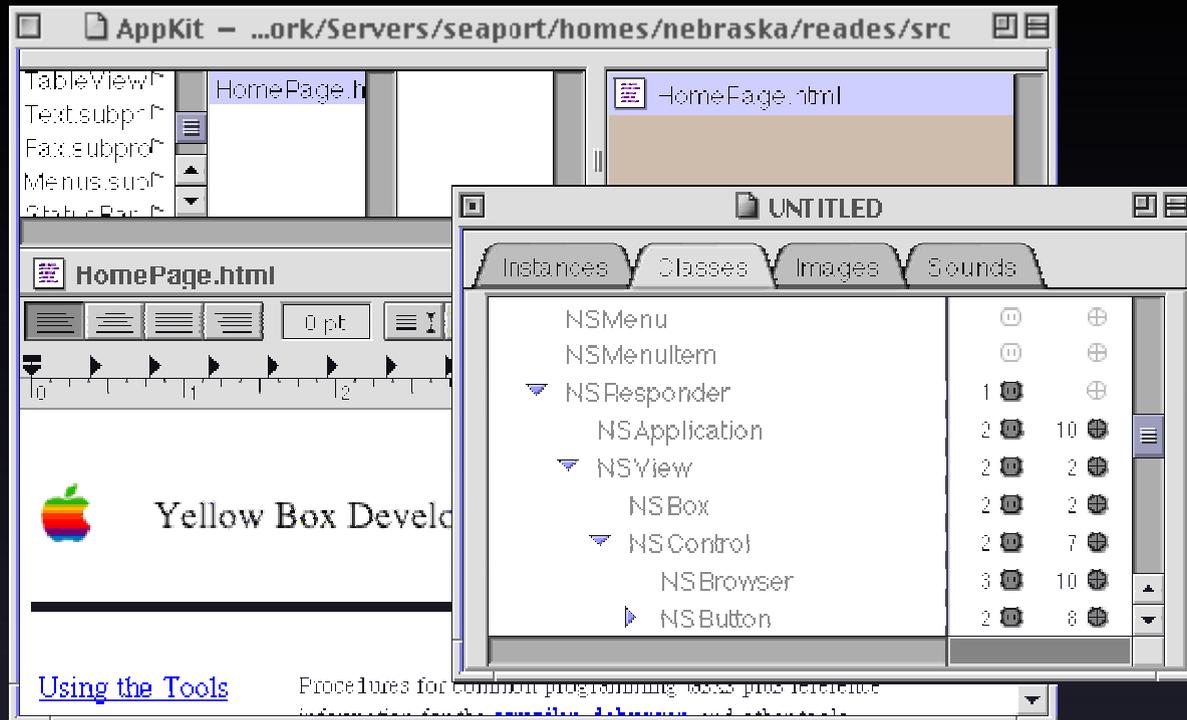


# Drawing

- NSImageRep, NSString, NSAttributedString—**drawAtPoint()**, **drawInRect()**
- NSImage —**compositeToPoint()**
- NSBezierPath—**stroke()**, **fill()**, **clip()**
- NSColor, NSFont—**set()**
- NSCell—**drawWithFrameInView()**



# NSView — Concrete Subclasses



# NSControl

- Triggered or activated by user actions
- Examples: buttons, textfields, sliders
- Have a *target* object and an *action* method



# Target/Action

- When a control is triggered, it sends the action method to its target object
- If its target is NULL,
  - The action method is sent to the first responder
  - Then through the responder chain
- Common examples:
  - **cut(), copy(), ...**



# Controls

- Basic
  - NSButton, NSSlider, NSTextField, NSColorWell
- More sophisticated
  - NSMatrix, NSMenu, NSScroller
- Most sophisticated
  - NSBrowser, NSTableView, NSOutlineView



# AppKit—Abstractions

- Notifications
- Delegation
- Data Abstractions



# Notifications

- Actually comes from Foundation
- Used to broadcast events throughout an app or even to other apps
- Examples:
  - **NSNotificationWillClose**
  - **NSNotificationWillTerminate**
  - **NSNotificationDidBecomeKey**



# Receiving Notifications

- Objects register with the notification center
  - **`NSNotificationCenter defaultCenter.addObserver(myObject, ...)`**
- You provide:
  - The name of the notification you want
  - The posting object (if any) to observe
  - The method to be called when the notification arrives



# Delegation

- Allows one object to act on behalf of another
- Avoids subclassing
- Unlike notifications, the delegate can affect the object
  - **windowShouldClose()**



# Implementing a Delegate

- Choose an object to be the delegate
- Look up the list of available delegate methods
- Implement the ones you're interested in
- Call **theObject.setDelegate()**



# Delegation vs. Notifications

- Notification observers are *passive*; delegates are *active*
- Only one delegate; potentially many observers
- Delegates allowed to change the actions taking place
  - **Should** vs. **Will/Did** method names



# Data Abstractions

- NSDocument
- NSImage and NSImageRep
- NSFont
- NSAttributedString
- NSColor



# User Interface, IB, and Nib Files

- Consistent with the platform
- Stored in nib files
- Created via Interface Builder
- A single project can have multiple nib files
- Typically one nib file per window or panel



# Nib Files

- An archive of the UI elements
- Also connections and other relationships between objects
  - View hierarchy
  - Target/action connections
  - Delegate connections



# Nib files

- Can be customized for different platforms (like all resources)
  - MyApp.nib, MyApp-macintosh.nib, MyApp-windows.nib . . .
- At runtime, load the resource appropriate for the platform
  - `NSApplication–loadNibNamed()`
  - `NSBundle–pathForResource()`





99 | Worldwide  
Developers  
Conference

# Demo

## InterfaceBuilder

# AppKit— The Text Subsystem

- International
- Flexible
- Powerful



# International

- Unicode characters in the backing store
- Character to glyph conversion for display
- Glyph reordering, required for displaying Indic scripts
- Inline input methods, for Asian language input



# Flexible

- Multiple replaceable classes with public APIs
- Uses of the system:
  - Basic string drawing (like UI widgets)
  - Simple word processing
  - Specialized text editing
  - High-end publishing



# Powerful

- Multiple fonts, graphics, paragraph styles with ruler support
- Ligatures, kerning, diacritical marks, baseline adjustment
- Spelling checker, hyphenation
- Asynchronous layout
- Undo
- Copy/paste, read/write rich text



# Documentation

- On the web
  - <http://developer.apple.com/techpubs/macosxserver>
  - Also in  
`/System/Documentation/Developer`
- Contains programming topics and class reference sheets
- Example code in  
`/System/Developer/Examples`



# Roadmap

---

**CoreFoundation  
Overview**

Hall 2  
**Wed., 1:00pm**

---

**Using ProjectBuilder for  
Mac OS X**

Hall A1  
**Thur., 2:30pm**

---

**Apple Development  
Tools for Mac OS X**

Hall A1  
**Thur., 4:00pm**

---

**What's New in the Cocoa  
(Yellow) Framework**

Hall B  
**Fri., 10:00am**





Think different.<sup>TM</sup>





99 | Worldwide  
Developers  
Conference

# Posting Notifications

- Notifications are posted to the notification center
- Posting objects provide:
  - The notification's name
  - The desired posting behavior
  - The desired coalescing behavior
  - A dictionary of extra information



# UI Elements— Standard Panels

- NSOpenPanel
- NSColorPanel
- NSFontPanel
- NSPrintPanel
- NSSavePanel



Welcome

To Advance through Presentation  
Use Page Up and Page Down Keys



99 | Worldwide  
Developers  
Conference