

# The Cocoa Document Architecture

Rob Keniger

[rob@menumachine.com](mailto:rob@menumachine.com)

twitter: rkeniger

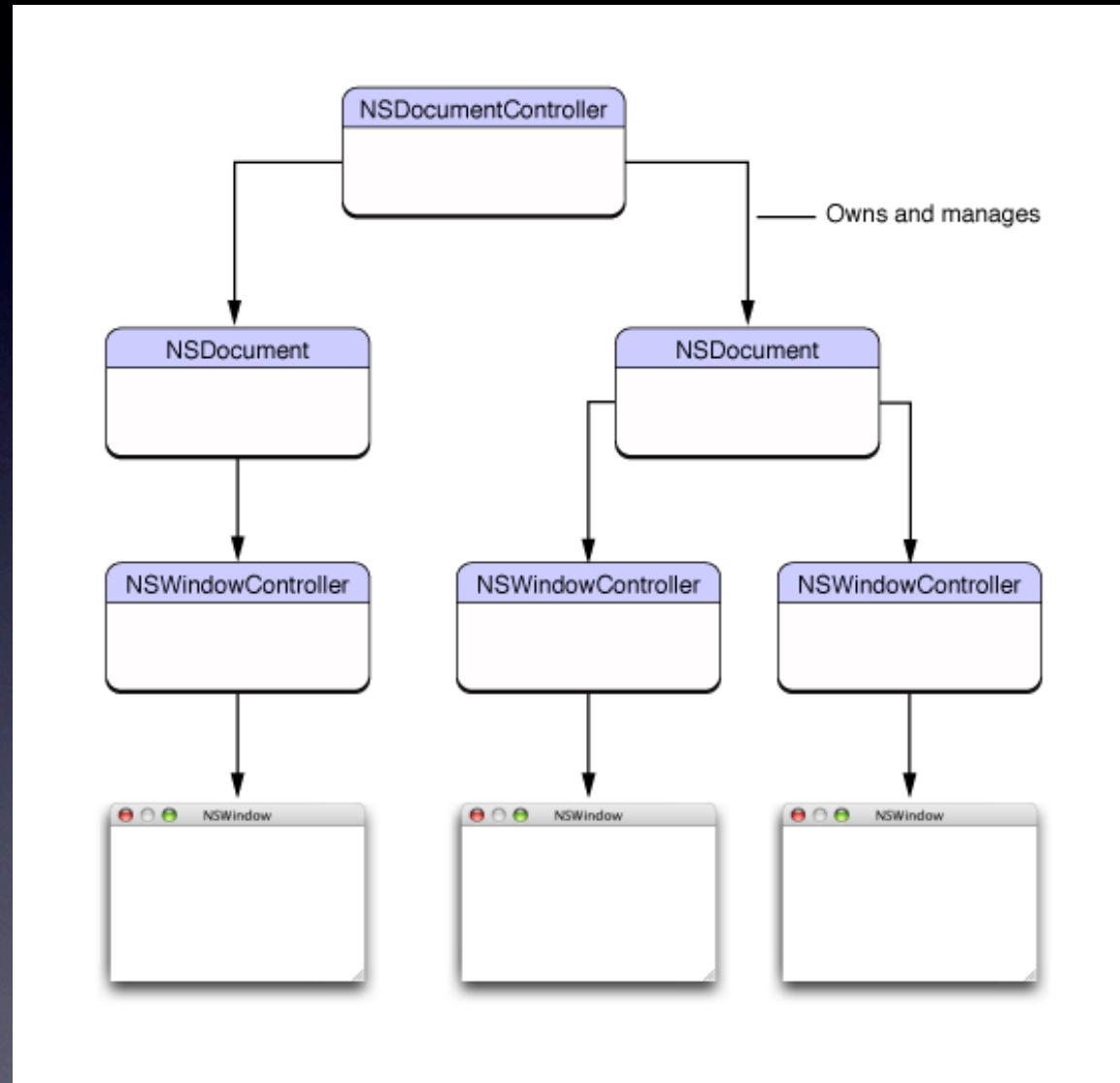


# What is it?

- A standard way of managing multiple documents
- Lots of stuff “for free”
- Simple to use, only three classes, two of which are optional



# The document class hierarchy



# NSWindowController

- Manages each window in a document
- One NSWindowController per nib
- Optional if your document has only one window
- Removes window management code from your NSDocument object
- Often manages the views in the window
- For complex windows, use in conjunction with NSViewController



# NSDocument

- Manages an individual document
- Represents the document's content
- Handles loading/saving of document data
- Manages NSWindowControllers
- Can do window management
- Implements an undo controller for undo management

# NSDocumentController

- Manages multiple documents
- Creates and opens documents
- Controls the default new document type
- Generally unnecessary to subclass
- Can be used by non-document-based apps for "Open Recent" menu



# Basic Usage

- Subclass NSDocument
- Override reading/writing methods
- Add a document type mapping to the app's Info.plist

# Reading and writing data

- Several possible methods to override depending on the type of data
- You can override one type for reading and another type for writing

## //File URLs

```
- (BOOL)readFromURL:(NSURL *)absoluteURL ofType:(NSString *)typeName error:(NSError **)outError  
- (BOOL)writeToURL:(NSURL *)absoluteURL ofType:(NSString *)typeName error:(NSError **)outError
```

## //Data

```
- (BOOL)readFromData:(NSData *)data ofType:(NSString *)typeName error:(NSError **)outError  
- (NSData *)dataOfType:(NSString *)typeName error:(NSError **)outError
```

## //File Wrappers

```
- (NSFileWrapper *)fileWrapperOfType:(NSString *)typeName error:(NSError **)outError  
- (BOOL)readFromFileWrapper:(NSFileWrapper *)fileWrapper ofType:(NSString *)typeName error:(NSError **)outError
```



# Editing the Info.plist File

- Get Info on the on the app target in Xcode and look at the Properties tab
- For more complex configuration you can edit the Info.plist file directly

# Editing the Info.plist File

Target "DocumentDemo" Info

General Build Rules Properties Comments

Executable: `${EXECUTABLE_NAME}`

Identifier: `com.yourcompany.${PRODUCT_NAME:rfc1034identifier}`

Type: APPL Creator: ????

Icon File: Cat

Version: 1

Principal Class: NSApplication

Main Nib File: MainMenu

Document Types:

Name	UTI	Extensions	MIME Types	OS Types	Class	Icon File	Store Type	Role	Packag
HTMLDocumentType	public.html				HTMLDocument		Binary	Viewer	<input type="checkbox"/>
LayoutDocumentType		ddlayout			LayoutDocument	Layout.icns	Binary	Editor	<input type="checkbox"/>
RichTextDocumentType	public.rtf				TextDocument		XML	Editor	<input type="checkbox"/>
PlainTextDocumentType	public.plain-text				TextDocument		XML	Editor	<input type="checkbox"/>

+ - Open Info.plist as File ?



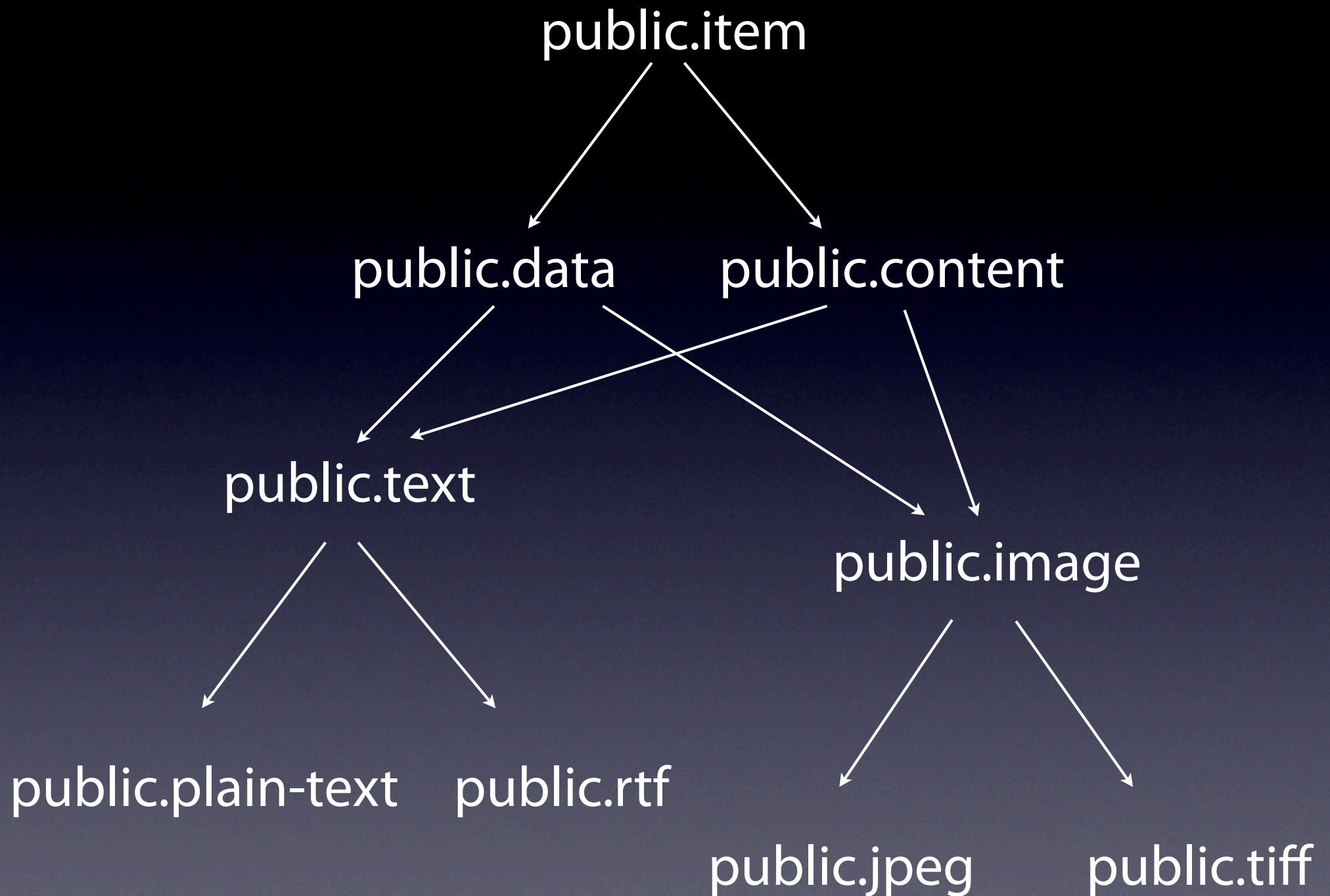
# Document Type Specification

- File extension
- HFS Type Code (Legacy)
- Uniform Type Identifier (UTI)

# Uniform Type Identifiers

- Relatively new, a few glitches
- The recommended way to handle types
- You can export your own types
- There are many system-defined types





DEMO



Rob Keniger  
rob@menumachine.com  
twitter: rkeniger

<http://github.com/rkeniger/CocoaHeadsDocumentDemo>