

The background features a dark, textured surface with a glowing blue and purple sphere in the center. The sphere has a white Apple logo on its top. A magnifying glass is positioned over the sphere, and a pen is visible on the right side. The text "Worldwide" is written in a golden, serif font across the top.

Worldwide

The background features a dark, textured surface with a glowing blue and purple sphere in the center. The sphere has a white Apple logo on its top. A magnifying glass is positioned over the sphere, and a pen is visible on the right side. The text "Developers" is written in a white, serif font across the middle, enclosed in a white rectangular border.

Developers

The background features a dark, textured surface with a glowing blue and purple sphere in the center. The sphere has a white Apple logo on its top. A magnifying glass is positioned over the sphere, and a pen is visible on the right side. The text "Conference" is written in a golden, serif font across the bottom.

Conference



Introduction to OpenStep's Foundation Framework

Jordan Dea-Mattson

**Senior Evangelist
Rhapsody Evangelism**

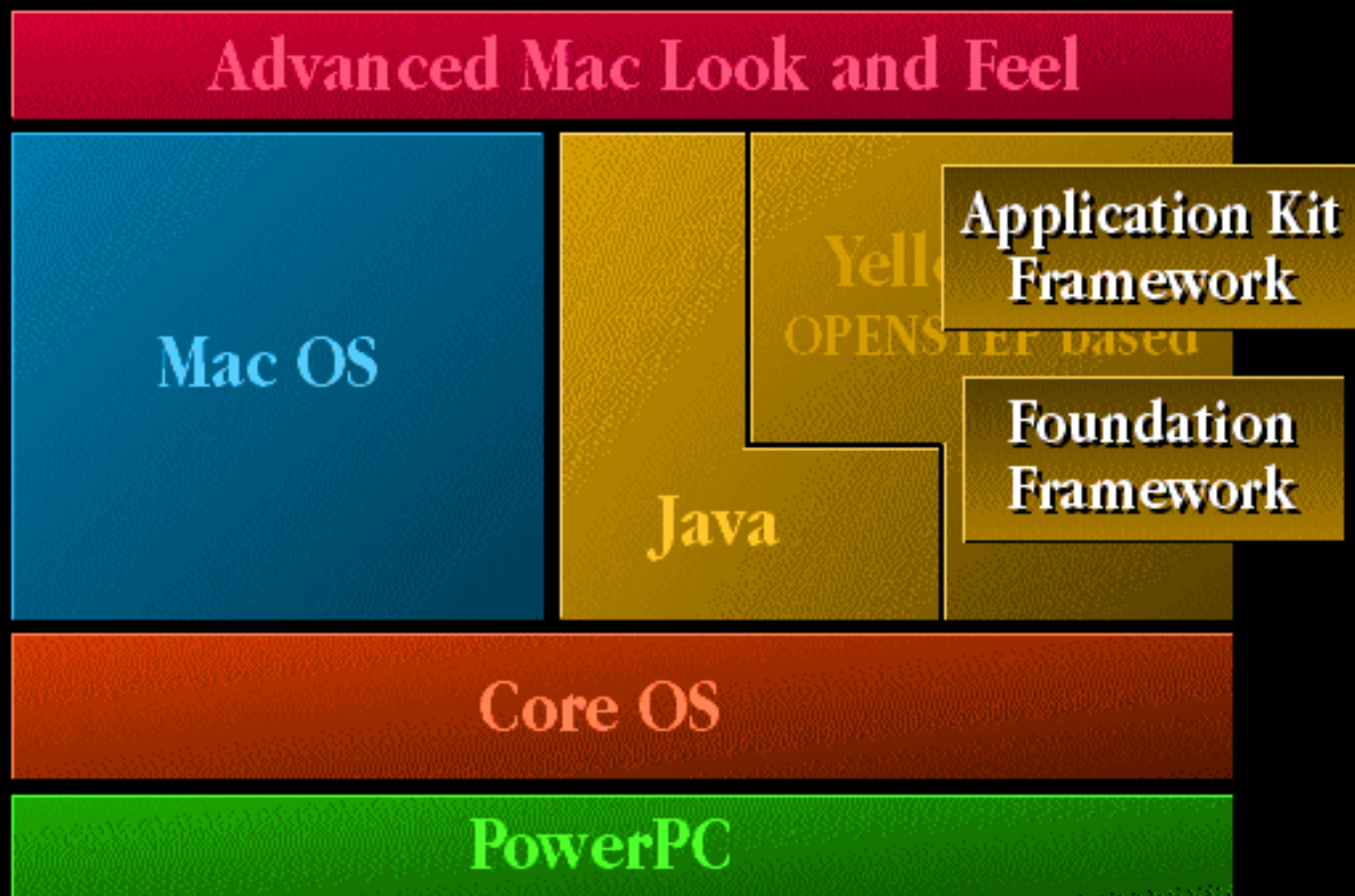


Introduction to Yellow Box Foundation Framework

Blaine Garst

Chris Kane

Where Does Foundation Fit?



Foundation Framework

- **Overview of the classes**
- **Major Design Patterns**
 - Class Clusters
 - Object Reclamation
- **Deep Diving**
 - Notifications
 - Run Loop
 - File Access
 - Distribution—see Session 220





Core Classes

Foundation Framework: Core

- **NSObject**—the root of (almost) everything
- **Basic value objects**
 - Strings, Dates, Numbers, Datas
- **Collection classes**
 - Arrays, Dictionaries, Sets
- **Mutability**
 - Small objects costly
 - Immutability mitigates cost



Foundation Framework: Application Programming

- **Notifications**
- **User Defaults**
- **Object Persistency**
- **Operating System Concepts**
 - Files, tasks, threads
 - Most classes not thread-safe
- **Event Loops**
- **Distributed Objects**





Design Patterns

Design Pattern: Class Clusters

- **Abstract class—defines the API, but no data storage**
 - Few abstract (primitive) methods
 - Many generic method implementations
- **Subclasses implement the primitive methods and define the data storage**
- **Foundation provides default (private) implementations for its clusters**
- **Constructors create an instance from an appropriate implementation**



Design Pattern: Class Clusters

Example: NSString

- **Several private subclasses**
- **API is in terms of Unicode**
- **Basic Unicode string subclass**
- **Other subclasses for 8-bit storage, when full Unicode not needed**
- **Constructors choose an appropriate subclass**



Design Pattern: Object Reclamation

- **Reference counting used to track references on objects**
 - Not for char *s, structs, or malloc() memory
- **Default implementation in NSObject**
- **References formalized with -retain/-release**
- **Weak references allowed, but do not guarantee validity over time**



Design Pattern: Object Reclamation

The ownership non-problem

- `NSAutoreleasePools` establish a working zone
- If you alloc/init, you must release
- If you retain, you must release
- Methods get/return usable objects
- Callers retain results for long term usage
- Pattern extends across a network



Design Pattern: Object Reclamation

```
- getTitle {  
    return title;  
    // and if we change it?  
    return [title copy];  
    // who reclaims it?  
    return [[title copy]  
    autorelease];  
}  
  
aTitle = [[x getTitle] retain];  
// we keep it
```



Design Pattern: Object Reclamation

```
NSAutoreleasePool pool =  
[NSAutoreleasePool new];  
... foo = [[x getTitle]  
retain];  
... bar = [y getTitle];  
[pool.release];  
[foo description]; //okay!  
[bar description]; // invalid!!
```





Deep Diving

*Don't hold your
breath!*

Deep Diving: Notifications

- **Notifications**
 - Sender, subject, target, user info
- **Notification centers**
 - Registrar for notification observers
 - Propagates notifications to registrants
- **Notification queues**
 - Event loop based propagation
- **Example from AppKit:**
NSNotification



Deep Diving: Event Loops

- **An NSRunLoop per thread**
- **Coordinates waiting upon inputs**
 - Timers
 - Ports
- **Inputs are grouped into modes**
 - Port delegate handles input
 - Modes allow broad filtering of inputs
 - Asynchronous file operation implementation



Deep Diving: File Access

- **Indirect access via other classes**
 - `x = [NSString stringWithContentsOfFile:@"f"];`
 - `y = [NSUnarchiver unarchiveObjectWithFile:@"g"];`
- **Direct access with file handles**
 - `z = [fh readDataOfLength:100];`
- **File system access API**
 - Moving, removing, and copying of files, getting and setting of attributes, etc.



Deep Diving: Distributed Objects (a.k.a. PDO)

- **Extensible**
 - Substitute underlying transport
 - Supply security layer
 - Substitute naming layer
- **Integrated support in ObjC language**
 - oneway, in, out, inout keyword support
 - byref, bycopy object copying directives
- **Transparent programming**
 - Exceptions, objects, etc., flow unimpeded
 - Garbage collection



Deep Diving: Distributed Objects

Client (and Server!!) coding example!

- `id client = [NSDistantObject proxyWithName:@“ideas” host:@“apple.com”];`
- `[client postSuggestion:@“buy NeXT”];`
- `...meanwhile`
- `id conn = [NSConnection connectionWithRoot:[Ideas new] name:@“ideas”];`
- `[[NSRunLoop currentRunLoop] run];`



Foundation Summary

- **Underpinnings for AppKit programming**
- **Design Patterns pervasive in Yellow Box**
- **Provides lots of rich cross platform services**





Q&A

The background features a dark, textured surface with a glowing blue and purple sphere in the center. A white Apple logo is positioned at the top of the sphere. The text 'Worldwide Developers Conference' is overlaid on the image. 'Worldwide' and 'Conference' are in a gold, serif font, while 'Developers' is in a white, serif font inside a white rectangular box. The overall aesthetic is futuristic and tech-oriented.

Worldwide

Developers

Conference