



Frameworks

by Theresa Ray of Tensor Information Systems, Inc.

Sponsored by Apple Computer, Inc.
Apple Developer Connection

media

Frameworks



by Theresa Ray of Tensor Information Systems, Inc.

A framework can be defined as a set of classes and other resources commonly available to many applications. Learning to use and create frameworks effectively will have a positive affect on your application development cycle-time and reliability.

When you write an object-oriented program, you rarely do it from scratch. There are almost always class definitions available that you can use. All you need are the class interface files, a library with compiled versions of the class implementations, and some documentation. In OpenStep/Yellow Box, groups of well-defined object-oriented classes with specific behavior are organized in structures known as frameworks.

What do frameworks provide?

A framework implements the basic classes which “glue” your custom objects to an application interface. The WebObjects framework, for example, provides a variety of ready-to-use application features, with classes for everything from collecting and managing session data to controlling the web application’s request/response loop. All of the basic code that is necessary to run a WebObjects application (and there’s a lot of it) is written, provided, and maintained by Apple in this framework, allowing you to focus on providing only the application-specific behavior for your application. The benefit is that the time to design, develop and deploy a complex application is orders of magnitude less that it would be if you tried to implement everything yourself – as well as being more reliable because there is less new code implemented.

Each application will require custom code to do what is unique to the application; however that code needs to fit in with the code provided in the WebObjects framework. For example, much of your application-specific code resides in methods which are invoked by messages sent from the WebObjects framework, such as `awake`, `sleep`, `takeValuesFromRequest:inContext:`, etc. And even though you may define the method name to invoke when submitting a form, the WebObjects framework actually sends the message at the appropriate point in the request-response loop. Much of a WebObjects application’s program is simply implementing methods that respond to system-generated messages.

Adapting a framework

There are many mechanisms you may use to adapt the code in a framework to the needs of your application, including subclasses, delegates, categories, protocols, and posers. Subclasses are discussed in the on-line documentation in the object-oriented techniques section. The topics of categories, protocols, and posers will be the subject of future survival guides.

A delegate is defined as an object that acts on behalf of another – that is an object that shares responsibility with the object it connects to. As stated above, this mechanism is frequently used to allow application-specific options within a framework, or to extend or modify the behavior of the framework’s classes. A delegate typically receives messages notifying it that certain actions are about to occur or have just occurred. The delegate can intervene, implement additional behavior, and even return a result that prevents the action from continuing (in the case of

notification before the action has occurred). Delegation is usually used to coordinate or synchronize a set of objects, and may eliminate the need to define a subclass.



Creating a framework

In addition to the frameworks supplied by Apple, you will likely find yourself developing some code that is not application specific. For example, you may develop a new class or a new WebObjects component that can be reused by many future applications at your company. You need to create your own framework to contain this code. A framework may contain resources related to a set of common code as well as the classes themselves. These related resources typically include images, help files and other documentation, and/or database mapping files (EOModels), but may include any non-executable resource that is logically associated with the class and is not application specific.

When you compile a framework, the appropriate executable and associated framework resources are created based on the platform used for compilation (for example a dll is created for NT and a shlib is created for Solaris). When you include this framework in a full application, the application must be able to find the framework executable at run-time. Therefore, on NT platforms the dll created for the framework must be in the path environment variable. On Solaris, the framework executable must be in the LD_LIBRARY_PATH environment variable. On Rhapsody and Mach, the path is compiled into the binary for the application's executable.

Framework Options

There are several implementation options available to you when creating a new framework. You may choose from a traditional OpenStep/Yellow Box framework, a WebObjects framework, or a Java package. You should carefully consider the opportunities available for reusing your code within your business before deciding what sort of framework should contain your classes.

An OpenStep/Yellow Box framework contains Objective-C classes and is available for inclusion in both web-based and client-server applications. Extensions to Apple's Foundation or Enterprise Object frameworks would likely be placed in this type of framework, but any code that would be used for both types of application interfaces should be placed here. You may want to consider creating a Java wrapper for these classes if your company prefers or is moving toward using Java code.

A WebObjects framework contains components and their associated classes (coded in Objective-C, WebScript, and/or Java), and are available for inclusion in WebObjects applications only. You may have classes without components, such as a subclass of the WebObjects Session class, as well as full components including code or script. You may want to place your reusable WebObjects components on a custom palette for ease of access in WebObjects Builder.

A Java package contains Java classes and is available for inclusion in any Java application, whether web-based or client-server based. This is basically the Java equivalent of an OpenStep/Yellow Box framework.

Which type of framework you chose to implement is purely a business decision based on your company's code practices and projections for future implementation. Do you only develop web-based applications? Do you or will you use Java? Will you ever have the need to implement a client-server or command-line application?

A Word About Bundles

Bundles are similar to frameworks, but are usually not well understood by beginning OpenStep/Yellow Box programmers. Their purpose is different from frameworks, but their basic structure and implementation are similar.

The main difference between frameworks and bundles is that frameworks are loaded automatically at run time, whereas bundles are loaded explicitly when needed by an application.



Bundles typically contain code that is rarely used but necessary for an application such as help files, a filter for a certain editor in the application, or user-interface text for multiple language support. Instead of wasting run-time resources, such code is loaded only when user action requires it. NSBundle methods may be used on frameworks and bundles alike to load resources as needed, or during debug to determine what frameworks or bundles are loaded, or to determine what directory a resource was loaded from (see NSBundle's class methods `allBundles` and `allFrameworks` in the on-line documentation for more information).

Versioning

Versioning may be an issue if you have a framework that is used by production applications that cannot stand a change to the framework's header file (due to certain subclassing implementations, for example). You have updates to make to the framework, but you cannot afford to change the "Current" version. Project Builder provides a useful feature for versioning frameworks on platforms supporting symbolic links. If you click on the Inspector's icon and choose Project Attributes, you can set options for the version name to install with, and whether or not to make it the "Current" version. A separate subdirectory is created for each version name, and a symbolic link is created which points to the "Current" version. Your production applications can continue to reference the "Current" version (or may be changed to reference a specific version), while your new applications can take advantage of the updated code. More information about versioning control of frameworks can be found in the on-line documentation.

Headers

When creating a framework, a decision must be made whether to make the header or Java class public or private. Headers should obviously be made public for common classes that are intended to be subclassed by other applications. But in general it seems that a framework will need nearly all its headers to be public, though you may not be able to foresee the usefulness of subclassing a particular class. Unless you have an overwhelming and specific need to keep the header private, public headers are usually the standard for framework classes.

Conclusion

Frameworks are an essential part of object-oriented programming. Using basic or "root" classes developed by Apple and by others within your company provides a huge jump-start on your programming. Only the application-specific code needs to be developed, lowering the time required to implement the application as well as reducing the opportunity for error. Proper creation and structuring of frameworks (by using public headers and implementing delegate methods, for example) provides maximum opportunity for reuse among applications – even applications which are very different in structure and purpose.

Phone: 817-335-7770

E-mail: theresa@tensor.com

URL: <http://www.tensor.com>

Tips...



- To make a public header, choose the header file in Project Builder's Project Browser window. Click on the Inspector's icon, and choose File Attributes. Select the appropriate checkbox for public or private.
 - To check the install directory for a framework, click on the Inspector's icon in Project Builder and choose Build Attributes. The Install In: directory must be able to be found by applications that will include the framework you are creating.
 - To install a framework, click on the check box in the Project Builder's Build Panel, choose an install target, then compile. Your framework is now installed and ready for use by other applications.
 - If a framework has the potential for an area of application-specific logic, that code can be assigned to a delegate. Each application needing to modify the default logic may register itself as a delegate and modify the result as needed.
- * Frameworks are also referred to as Kits in some of the older documentation.

Resources . . .



<http://gemma.apple.com/techinfo/techdocs/enterprise/WebObjects>

WebObjects Developer's Guide

Enterprise Objects Framework Developer's Guide

<http://www.omnigroup.com/MailArchive/WebObjects>

<http://www.omnigroup.com/MailArchive/eof>

<http://www2.stepwise.com/cgi-bin/WebObjects/Stepwise/Sites>

ftp://dev.apple.com/devworld/Interactive_Media_Resources

<http://www.apple.com/developer>

<http://developer.apple.com/media>

<http://enterprise.apple.com/NeXTanswers>

About the Author . . .

Theresa Ray is a Senior Software Consultant for Tensor Information Systems in Fort Worth, TX (<http://www.tensor.com>). She has worked as a consultant on WebObjects projects for a wide variety of clients including the U.S. Navy, the United States Postal Service, America Online, and Proctor and Gamble. Her experience spans all versions of WebObjects, from 1.0 to 4.0 beta, several versions of EOF, from 1.1 to 3.0 beta, AppKit, NEXTSTEP 3.1 to OPENSTEP 4.2, Rhapsody for Power Macintosh, and yellow-box for NT. In addition, she is an Apple-certified instructor for WebObjects courses.

Tensor Information Systems is a Apple partner providing systems integration and enterprise solutions to its customers. Tensor's employees are experienced in all Apple technologies including OPENSTEP, NEXTSTEP, Rhapsody, EOF and WebObjects. Tensor also provides Apple-certified training in WebObjects, Oracle consulting and training, as well as systems integration consulting on HP-UX.

You may reach Theresa by e-mail: theresa@tensor.com

Interactive Media Resources Include:

Interactive Media Guidebooks

Market Research Reports

Survival Guides—
Technical “How To” Guides

Comarketing Opportunities

Special Discounts

Apple Developer Connection



As Apple technologies such as QuickTime, ColorSync, and AppleScript continue to expand Macintosh as the tool of choice for content creators and interactive media authors, the Apple Developer Connection continues its commitment to provide creative professionals with the latest technical and marketing information and tools.

Interactive Media Resources

Whether looking for technical guides from industry experts or for market and industry research reports to help make critical business decisions, you'll find them on the Interactive Media Resources page.

Apple Developer Connection Programs and Products

ADC programs and products offer easy access to technical and business resources for anyone interested in developing for Apple platforms worldwide. Apple offers three levels of program participation serving developer needs.

Membership Programs

Online Program—Developers gain access to the latest technical documentation for Apple technologies as well as business resources and information through the Apple Developer Connection web site.

Select Program—Offers developers the convenience of technical and business information and resources on monthly CDs, provides access to prerelease software, and bundles two technical support incidents.

Premier Program—Meets the needs of developers who desire the most complete suite of products and services from Apple, including eight technical support incidents and discounts on Apple hardware.

Standalone Products

Apple offers many standalone products that allow developers to choose their own level of support from Apple or enhance their Select or Premier Program membership. Choose from the following products and begin enjoying the benefits today.

Make the Connection.

Join ADC today!

<http://developer.apple.com/programs>



Developer Connection Mailing—Subscribe to the Apple Developer Connection Mailing for the latest in development tools, system software, and more.

Technical Support—Purchase technical support and work directly with Apple's Worldwide Developer Technical Support engineers.

Apple Developer Connection News—Stay connected to Apple and developer-specific news by subscribing to our free weekly e-mail newsletter, Apple Developer Connection News. Each newsletter contains up-to-date information on topics such as Mac OS, Interactive Media, Hardware, Apple News and Comarketing Opportunities.

Macintosh Products Guide

The most complete guide for Macintosh products! Be sure to list your hardware and software products in our *free* online database!

<http://developer.apple.com/media>
The ultimate source for creative professionals.