Copyright ©1996 by NeXT Software, Inc. All Rights Reserved.

Glossary

abstract class A class that's defined solely so that other classes can inherit from it. Programs don't use instances of an abstract class, only of its subclasses.

abstract superclass Same as *abstract class*.

action message In the Application Kit, a message sent by an object (such as an NSButton or NSSlider) in response to a user action (such as clicking the button or dragging the slider's knob). The message translates the user's action into a specific instruction for the application. See also *target*.

active application The application associated with keyboard events, the one the user is currently working in. On Mach, menus are visible on-screen only for the active application, and only the active application can have the current key window.

adopt In the Objective-C language, a class is said to adopt a protocol if it declares that it implements all the methods in the protocol. Protocols are adopted by listing their names between angle brackets in a class or category declaration.

anonymous object An object of unknown class. The interface to an anonymous object is published through a protocol declaration.

Application Kit The Objective-C classes and C functions available for implementing the window-based user interface in an application. The Application Kit provides a basic program structure for applications that draw on the screen and respond to events. The Application Kit is packaged as a *framework*.

archiving The process of preserving a data structure, especially an object, for later use. An archived data structure

is usually stored in a file, but it can also be written to memory, copied to the pasteboard, or sent to another application. In OpenStep, archiving involves writing data to an NSData object.

asynchronous message A remote message that returns immediately, without waiting for the application that receives the message to respond. The sending application and the receiving application act independently, and are therefore not ªin sync.º See also *synchronous message*.

category In the Objective-C language, a set of method definitions that is segregated from the rest of the class definition. Categories can be used to split a class definition into parts or to add methods to an existing class.

class In the Objective-C language, a prototype for a particular kind of object. A class definition declares instance variables and defines methods for all members of the class. Objects that have the same types of instance variables and have access to the same methods belong to the same class. See also *class object*.

class method In the Objective-C language, a method that can be used by the class object rather than by instances of the class.

class object In the Objective-C language, an object that represents a class and knows how to create new instances of the class. Class objects are created by the compiler, lack instance variables, and can't be statically typed, but otherwise behave like all other objects. As the receiver in a message expression, a class object is represented by the class name.

compile time The time when source code is compiled. Decisions made at compile time are constrained by the amount and kind of information encoded in source files.

conform In the Objective-C language, a class is said to conform to a protocol if it adopts the protocol or inherits from a class that adopts it. An instance conforms to a protocol if its class does. Thus, an instance that conforms to a protocol can perform any of the instance methods declared in the protocol.

content view In the Application Kit, the NSView object that's associated with the content area of a windowÐall the area in the window excluding the title bar, resize bar, and border. All other NSViews in the window are arranged in a hierarchy beneath the content view.

controls Graphical objectsÐsuch as buttons, sliders, text fields, and scrollersÐthat the user can operate to give instructions to an application.

cursor The small image (usually an arrow) that moves on the screen and is controlled by moving the mouse.

delegate An object that acts on behalf of another object.

designated initializer The **init...** method that has primary responsibility for initializing new instances of a class. Each class defines or inherits its own designated initializer. Through messages to **self**, other **init...** methods in the same class directly or indirectly invoke the designated initializer, and the designated initializer, through a message to **super**, invokes the designated initializer of its superclass.

dynamic binding Binding a method to a messageÐthat is, finding the method implementation to invoke in response to the messageÐat run time, rather than at compile time.

dynamic typing Discovering the class of an object at run time rather than at compile time.

event The direct or indirect report of external activity, especially user activity on the keyboard and mouse.

event message In the Application Kit, a message to perform a method named after an event or sub-event. Event messages are used to dispatch events to the objects that will respond to them. See also *action message*.

factory Same as *class object*.

factory method Same as *class method*.

factory object Same as *class object*.

file package A directory that is presented as a file, allowing the user to manipulate a group of files as if they were one file. A file package for an application executable has the same name as the executable file, plus a ª.appº extension. File packages for documents and bundles bear an extension that's recognized as belonging to a particular application.

formal protocol In the Objective-C language, a protocol that's declared with the **@protocol** directive. Classes can adopt formal protocols, objects can respond at run time when asked if they conform to a formal protocol, and instances can be typed by the formal protocols they conform to.

frameworkA way to package a logically-related set of classes, protocols and functions together with localized strings, on-line documentation, and other pertinent files. OPENSTEP provides the Foundation framework and the

Application Kit framework, among others. Frameworks are sometimes referred to as ªkits.º

id In the Objective-C language, the general type for any kind of object regardless of class. **id** is defined as a pointer to an object data structure. It can be used for both class objects and instances of a class.

informal protocol In the Objective-C language, a protocol declared as a category, usually as a category of the NSObject class. The language gives explicit support to formal protocols, but not to informal ones.

inheritance In object-oriented programming, the ability of a superclass to pass its characteristics (methods and instance variables) on to its subclasses.

inheritance hierarchy In object-oriented programming, the hierarchy of classes that's defined by the arrangement of superclasses and subclasses. Every class (except root classes such as NSObject) has a superclass, and any class may have an unlimited number of subclasses. Through its superclass, each class inherits from those above it in the hierarchy.

instance In the Objective-C language, an object that belongs to (is a member of) a particular class. Instances are created at run time according to the specification in the class definition.

instance method In the Objective-C language, any method that can be used by an instance of a class rather than by the class object.

instance variable In the Objective-C language, any variable that's part of the internal data structure of an instance. Instance variables are declared in a class definition and become part of all objects that are members of or inherit from the class.

Interface Builder A tool that lets you graphically specify your application's user interface. It sets up the corresponding objects for you and makes it easy for you to establish connections between these objects and your own code where needed.

introspection The ability of an object to reveal information about itself as an objectÐsuch as its class and superclass, the messages it can respond to, and the protocols it conforms to.

key window The window in the active application that receives keyboard events and is the focus of user activity.

link time The time when files compiled from different source modules are linked into a single program. Decisions

made by the linker are constrained by the compiled code and ultimately by the information contained in source code.

localize To adapt an application to work under various local conditionsÐespecially to have it use a language selected by the user. Localization entails freeing application code from language-specific and culture-specific references and making it able to import localized resources (such as character strings, images, and sounds). For example, an application localized in Spanish would display ªSalirº as the last item in the main menu. In Italian, it would be ªEsci,º in German ªVerlassen,º and in English ªQuit.º

main event loop The principal control loop for applications that are driven by events. From the time it's launched until the moment it's terminated, an application gets one keyboard or mouse event after another from the Window Server and responds to them, waiting between events if the next event isn't ready. In the Application Kit, the NSApplication object runs the main event loop.

menu A small window that displays a list of commands. Only menus for the active application are visible on-screen.

message In object-oriented programming, the method selector (name) and accompanying arguments that tell the receiving object in a message expression what to do.

message expression In object-oriented programming, an expression that sends a message to an object. In the Objective-C language, message expressions are enclosed within square brackets and consist of a receiver followed by a message (method selector and arguments).

method In object-oriented programming, a procedure that can be executed by an object.

modal event loop A temporary event loop that's set up to get events directly from the event queue, bypassing the main event loop. Typically, a mouse-down event initiates the modal loop and the following mouse-up event ends it. The loop gets mouse-dragged events (or mouse-entered and mouse-exited events) to track the cursor's movement while the user holds the mouse button down.

multiple inheritance In object-oriented programming, the ability of a class to have more than one superclassÐto inherit from different sources and thus combine separately-defined behaviors in a single class. Objective-C doesn't support multiple inheritance.

name space A logical subdivision of a program within which all names must be unique. Symbols in one name space won't conflict with identically named symbols in another name space. For example, in Objective-C, the instance methods of each class are in a separate name space, as are the class methods and instance variables

OPENSTEP A set of frameworks, including Foundation and the Application Kit. NeXT also includes an application development and user environment, consisting of the Workspace Manager, the Window Server, Project Builder and Interface Builder, and other software.

nib file A file (actually a file package) that stores the specifications for all or part of an application's interface. Nib files are created using Interface Builder and can contain archived objects, information about connections between objects, and sound and image data.

nil In the Objective-C language, an object **id** with a value of 0.

object A programming unit that groups together a data structure (instance variables) and the operations (methods) that can use or affect that data. Objects are the principal building blocks of object-oriented programs.

outlet An instance variable that points to another object. Outlet instance variables are a way for an object to keep track of the other objects to which it may need to send messages.

panel A window that holds objects that control what happens in other windows (such as a Font panel) or in the application generally (such as a Preferences panel), or a window that presents information about the application to the user (such as an information panel). See also *attention panel*.

polymorphism In object-oriented programming, the ability of different objects to respond, each in its own way, to the same message.

pop-up list A menu-like list of items that appears over (or next to) an on-screen button when the button is pressed. The user can choose an item by dragging to it and releasing the mouse button. When the mouse button is released, the pop-up list disappears.

procedural programming language A language, like C, that organizes a program as a set of procedures that have definite beginnings and ends.

protocol In the Objective-C language, the declaration of a group of methods not associated with any particular

class. See also *formal protocol* and *informal protocol*.

receiver In object-oriented programming, the object that is sent a message.

remote message A message sent from one application to an object in another application.

remote object An object in another application, one that's a potential receiver for a remote message.

run time The time after a program is launched and while it's running. Decisions made at run time can be influenced by choices the user makes.

selector In the Objective-C language, the name of a method when it's used in a source-code message to an object, or the unique identifier that replaces the name when the source code is compiled. Compiled selectors are of type SEL.

static typing In the Objective-C language, giving the compiler information about what kind of object an instance is, by typing it as a pointer to a class.

subclass In the Objective-C language, any class that's one step below another class in the inheritance hierarchy. Occasionally used more generally to mean any class that inherits from another class, and sometimes also used as a verb to mean the process of defining a subclass of another class.

superclass In the Objective-C language, a class that's one step above another class in the inheritance hierarchy; the class through which a subclass inherits methods and instance variables.

surrogate An object that stands in for and forwards messages to another object.

synchronous message A remote message that doesn't return until the receiving application finishes responding to the message. Because the application that sends the message waits for an acknowledgment or return information from the receiving application, the two applications are kept ªin sync.º See also *asynchronous message*.

target In the Application Kit, the object that receives action messages from an NSControl.

typed stream A specialized data stream used for archiving. When a typed stream is used, the type of the data is archived along with the data and an object's class hierarchy and version are archived with the object. See also **archiving**.

Window Server A process that dispatches user events to applications and renders PostScript code on behalf of applications.

zone A particular region of dynamic memory. Zones are set up in program code and are passed to allocation methods and functions to specify that the allocated memory should come from a particular zone. Allocating related data structures from the same zone can improve locality of reference and overall system performance.