

Run-Time System

Library: libsys_s.a

Header File Directory: /NextDeveloper/Headers/objc

Introduction

The ObjectiveC language pushes many decisions from compile time to run time. For example, objects are dynamically allocated and initialized at run time, messages are dynamically bound to method implementations, and objects assigned to **id** variables are dynamically typed. To correctly and efficiently carry out these tasks as a program executes, the language depends on a *run-time system*—a body of code to operate the object-oriented machinery.

The run-time system consists mainly of:

- Data structures that the compiler develops from class and category definitions and from protocol declarations, and
- The functions that operate on those structures and that are called by compiled ObjectiveC code to produce the desired results.

For example, the compiler translates an ObjectiveC message into a call on a messaging function, usually **objc_msgSend()**, which locates the method implementation that should be invoked in response to the message. The messaging function is what makes dynamic binding work. It and the data structures it requires are part of the run-time system.

For the most part, the run-time system operates behind the scenes. It has a public interface in part to declare common elements that every ObjectiveC program uses—such as the **id** data type. But mainly the interface exists to provide access to run-time code from outside ObjectiveC. Therefore, most of the elements documented here will rarely if ever appear in ObjectiveC programs.

Note: The principal interface to the run-time system is contained in the Object class. Because this class is fundamental to all NEXTSTEP software kits, it's presented at the very beginning of this manual, in Chapter 1, "Root Class."