

Introduction

Important: This documentation contains preliminary information about an API or technology in development. This information is subject to change, and software implemented according to this documentation should be tested with final operating system software.

Block objects are a C-level syntactic and runtime feature. They are similar to standard C functions, but in addition to executable code they may also contain variable bindings to automatic (stack) or managed (heap) memory. A block can therefore maintain a set of state (data) that it can use to impact behavior when executed.

You can use blocks to compose function expressions that can be passed to API, optionally stored, and used by multiple threads. Blocks are particularly useful as a callback because the block carries both the code to be executed on callback and the data needed during that execution.

Blocks are available in GCC and Clang as shipped with the OS X v10.6 Xcode developer tools. You can use blocks with OS X v10.6 and later, and iOS 4.0 and later. The blocks runtime is open source and can be found in LLVM's compiler-rt subproject repository. Blocks have also been presented to the C standards working group as N1370: Apple's Extensions to C. As Objective-C and C++ are both derived from C, blocks are designed to work with all three languages (as well as Objective-C++). The syntax reflects this goal.

You should read this document to learn what block objects are and how you can use them from C, C++, or Objective-C.

Organization of This Document

This document contains the following chapters:

- Getting Started with Blocks provides a quick, practical, introduction to blocks.
- Conceptual Overview provides a conceptual introduction to blocks.
- Declaring and Creating Blocks shows you how to declare block variables and how to implement blocks.
- Blocks and Variables describes the interaction between blocks and variables, and defines the `__block` storage type modifier.
- Using Blocks illustrates various usage patterns.