

Designing for Performance

[On This Page](#)

Key-value coding is efficient, especially when you rely on the default implementation to do most of the work, but it does add a level of indirection that is slightly slower than direct method invocations. Use key-value coding only when you can benefit from the flexibility that it provides, or to allow your objects to participate in the Cocoa technologies that depend on it.

Overriding Key-Value Coding Methods

Typically, you make your objects key-value coding compliant by ensuring that they inherit from `NSObject`, and then providing the property-specific accessors and related methods as described throughout this book. You rarely need to override the default implementations of the key-value coded accessors, such as `valueForKey:` and `setValue:forKey:`, or the key-based validation methods like `validateValue:forKey:`. Because these implementations cache information about the runtime environment to increase efficiency, if you do override them to introduce custom logic, make sure that you invoke the default implementation in the superclass before returning.

Optimizing To-Many Relationships

When you implement to-many relationships, the indexed form of the accessors provides significant performance gains in many cases, especially for mutable collections. See [Accessing Collection Properties](#) and [Defining Collection Methods](#) for further information.

Copyright © 2016 Apple Inc. All rights reserved. [Terms of Use](#) | [Privacy Policy](#) | Updated: 2016-10-24