

---

# NSCoding Protocol Reference

Data Management: File Management



2009-11-17



Apple Inc.  
© 2009 Apple Inc.  
All rights reserved.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, mechanical, electronic, photocopying, recording, or otherwise, without prior written permission of Apple Inc., with the following exceptions: Any person is hereby authorized to store documentation on a single computer for personal use only and to print copies of documentation for personal use provided that the documentation contains Apple's copyright notice.

The Apple logo is a trademark of Apple Inc.

Use of the "keyboard" Apple logo (Option-Shift-K) for commercial purposes without the prior written consent of Apple may constitute trademark infringement and unfair competition in violation of federal and state laws.

No licenses, express or implied, are granted with respect to any of the technology described in this document. Apple retains all intellectual property rights associated with the technology described in this document. This document is intended to assist application developers to develop applications only for Apple-labeled computers.

Every effort has been made to ensure that the information in this document is accurate. Apple is not responsible for typographical errors.

Apple Inc.  
1 Infinite Loop  
Cupertino, CA 95014  
408-996-1010

Apple, the Apple logo, Cocoa, Mac, Mac OS, and Quartz are trademarks of Apple Inc., registered in the United States and other countries.

Simultaneously published in the United States and Canada.

**Even though Apple has reviewed this document, APPLE MAKES NO WARRANTY OR REPRESENTATION, EITHER EXPRESS OR IMPLIED, WITH RESPECT TO THIS DOCUMENT, ITS QUALITY, ACCURACY, MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE. AS A RESULT, THIS DOCUMENT IS PROVIDED "AS IS," AND YOU, THE READER, ARE ASSUMING THE ENTIRE RISK AS TO ITS QUALITY AND ACCURACY.**

**IN NO EVENT WILL APPLE BE LIABLE FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES RESULTING FROM ANY**

**DEFECT OR INACCURACY IN THIS DOCUMENT, even if advised of the possibility of such damages.**

**THE WARRANTY AND REMEDIES SET FORTH ABOVE ARE EXCLUSIVE AND IN LIEU OF ALL OTHERS, ORAL OR WRITTEN, EXPRESS OR IMPLIED. No Apple dealer, agent, or employee is authorized to make any modification, extension, or addition to this warranty.**

**Some states do not allow the exclusion or limitation of implied warranties or liability for incidental or consequential damages, so the above limitation or exclusion may not apply to you. This warranty gives you specific legal rights, and you may also have other rights which vary from state to state.**

# Contents

---

## **NSCoding Protocol Reference 5**

---

Overview 5

Tasks 6

    Initializing with a Coder 6

    Encoding with a Coder 6

Instance Methods 6

    encodeWithCoder: 6

    initWithCoder: 6

---

## **Document Revision History 9**

---

**Index 11**

---



# NSCoding Protocol Reference

---

<b>Adopted by</b>	Various Cocoa classes
<b>Framework</b>	/System/Library/Frameworks/Foundation.framework
<b>Availability</b>	Available in Mac OS X v10.0 and later.
<b>Companion guide</b>	Archives and Serializations Programming Guide for Cocoa
<b>Declared in</b>	NSObject.h
<b>Related sample code</b>	iSpend QTQuartzPlayer SourceView Squiggles With and Without Bindings

## Overview

The `NSCoding` protocol declares the two methods that a class must implement so that instances of that class can be encoded and decoded. This capability provides the basis for archiving (where objects and other structures are stored on disk) and distribution (where objects are copied to different address spaces).

In keeping with object-oriented design principles, an object being encoded or decoded is responsible for encoding and decoding its instance variables. A coder instructs the object to do so by invoking `encodeWithCoder:` (page 6) or `initWithCoder:` (page 6). `encodeWithCoder:` (page 6) instructs the object to encode its instance variables to the coder provided; an object can receive this method any number of times. `initWithCoder:` (page 6) instructs the object to initialize itself from data in the coder provided; as such, it replaces any other initialization method and is sent only once per object. Any object class that should be codable must adopt the `NSCoding` protocol and implement its methods.

It is important to consider the possible types of archiving that a coder supports. On Mac OS X version 10.2 and later, keyed archiving is preferred. You may, however, need to support classic archiving. For details, see *Archives and Serializations Programming Guide for Cocoa*.

## Tasks

### Initializing with a Coder

- `initWithCoder:` (page 6) *required method*  
Returns an object initialized from data in a given unarchiver. (required)

### Encoding with a Coder

- `encodeWithCoder:` (page 6)  
Encodes the receiver using a given archiver.

## Instance Methods

### **encodeWithCoder:**

Encodes the receiver using a given archiver.

```
- (void)encodeWithCoder:(NSCoder *)encoder
```

#### **Parameters**

*encoder*

An archiver object.

#### **Availability**

Available in Mac OS X v10.0 and later.

#### **Declared In**

NSObject.h

### **initWithCoder:**

Returns an object initialized from data in a given unarchiver. (required)

```
- (id)initWithCoder:(NSCoder *)decoder
```

#### **Parameters**

*decoder*

An unarchiver object.

#### **Return Value**

*self*, initialized using the data in *decoder*.

#### **Availability**

Available in Mac OS X v10.0 and later.

**Related Sample Code**  
Movie Overlay

**Declared In**  
NSObject.h



# Document Revision History

---

This table describes the changes to *NSCoding Protocol Reference*.

Date	Notes
2009-11-17	Corrected typo.
2006-05-23	First publication of this content as a separate document.

**REVISION HISTORY**

Document Revision History

# Index

---

## E

---

`encodeWithCoder:` protocol instance method [6](#)

## I

---

`initWithCoder:` protocol instance method [6](#)