

---

# NSDeserializer Class Reference

[Cocoa > Objective-C Language](#)



2006-05-23



Apple Inc.  
© 2006 Apple Computer, 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 Objective-C 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

---

## **NSDeserializer Class Reference 5**

---

Overview 5

Tasks 5

    Deserializing a Property List 5

## **Appendix A      Deprecated NSDeserializer Methods 7**

---

Deprecated in Mac OS X v10.2 7

    deserializePropertyListFromData:atCursor:mutableContainers: 7

    deserializePropertyListFromData:mutableContainers: 7

    deserializePropertyListLazilyFromData:atCursor:length:mutableContainers: 8

## **Document Revision History 9**

---

## **Index 11**

---



# NSDeserializer Class Reference

---

<b>Inherits from</b>	NSObject
<b>Conforms to</b>	NSObject (NSObject)
<b>Framework</b>	/System/Library/Frameworks/Foundation.framework
<b>Availability</b>	Deprecated in Mac OS X v10.2.
<b>Companion guide</b>	Archives and Serializations Programming Guide for Cocoa
<b>Declared in</b>	NSSerialization.h

## Overview

**Note:** NSDeserializer is obsolete and has been deprecated. Instead use NSPropertyListSerialization.

The `NSDeserializer` class declares methods that convert a representation of a property list (as contained in an `NSData` object) into a structure of property list objects in memory. The `NSDeserializer` class object itself provides these methods—you don't create instances of `NSDeserializer`. Options to these methods allow you to specify that container objects (arrays or dictionaries) in the resulting graph be mutable or immutable; that deserialization begin at the start of the data or from some position within it; or that deserialization occur lazily, so a property list is deserialized only if it is actually going to be accessed.

## Tasks

### Deserializing a Property List

+ `deserializePropertyListFromData:atCursor:mutableContainers:` (page 7) **Deprecated in Mac OS X v10.2**

Returns a property list object from a given location in a given serialized representation of a property list.

+ `deserializePropertyListFromData:mutableContainers:` (page 7) **Deprecated in Mac OS X v10.2**

Returns a property list object from given serialized data, optionally making the list elements mutable.

+ `deserializePropertyListLazilyFromData:atCursor:length:mutableContainers:` (page 8) **Deprecated in Mac OS X v10.2**

Returns a property list from a given location in a given serialized representation of a property list.



# Deprecated NSDeserializers Methods

---

A method identified as deprecated has been superseded and may become unsupported in the future.

## Deprecated in Mac OS X v10.2

### **deserializePropertyListFromData:atCursor:mutableContainers:**

Returns a property list object from a given location in a given serialized representation of a property list. (Deprecated in Mac OS X v10.2.)

```
+ (id)deserializePropertyListFromData:(NSData *)data atCursor:(unsigned *)cursor
    mutableContainers:(BOOL)mutable
```

#### **Parameters**

*data*

A serialized representation of a property list.

*cursor*

*mutable*

If YES and the property list object is a dictionary or an array, the recomposed object is made mutable.

#### **Return Value**

A property list object corresponding to the representation in *data* at the location *cursor*. Returns nil if the property list object is not valid for property lists.

#### **Availability**

Deprecated in Mac OS X v10.2.

#### **Declared In**

NSSerialization.h

### **deserializePropertyListFromData:mutableContainers:**

Returns a property list object from given serialized data, optionally making the list elements mutable. (Deprecated in Mac OS X v10.2.)

```
+ (id)deserializePropertyListFromData:(NSData *)serialization
    mutableContainers:(BOOL)mutable
```

#### **Parameters**

*serialization*

A serialized representation of a property list.

*mutable*

If YES and the property list object is a dictionary or an array, the recomposed object is made mutable.

## Deprecated NSDeserializer Methods

**Return Value**

A property list object corresponding to the representation in *serialization*, or `nil` if *serialization* does not represent a property list.

**Availability**

Deprecated in Mac OS X v10.2.

**Declared In**

`NSSerialization.h`

**deserializePropertyListLazilyFromData:atCursor:length:mutableContainers:**

Returns a property list from a given location in a given serialized representation of a property list. (Deprecated in Mac OS X v10.2.)

```
+ (id)deserializePropertyListLazilyFromData:(NSData *)data atCursor:(unsigned
    *)cursor length:(unsigned)length mutableContainers:(BOOL)mutable
```

**Parameters**

*data*

A serialized representation of a property list.

*cursor*

The cursor location.

*length*

The number of bytes to read.

*mutable*

If YES and the object is a dictionary or an array, the recomposed object is made mutable.

**Return Value**

A property list from *data* at location *cursor*, or `nil` if *data* does not represent a property list.

**Discussion**

The deserialization proceeds lazily—that is, if the data at *cursor* has a length greater than *length*, a proxy is substituted for the actual property list as long as the constituent objects of that property list are not accessed.

**Availability**

Deprecated in Mac OS X v10.2.

**Declared In**

`NSSerialization.h`



# Document Revision History

---

This table describes the changes to *NSDeserializer Class Reference*.

Date	Notes
2006-05-23	First publication of this content as a separate document.

**REVISION HISTORY**

Document Revision History

# Index

---

## D

---

deserializePropertyListFromData:atCursor:  
mutableContainers: **class method 7**

deserializePropertyListFromData:mutableContainers:  
**class method 7**

deserializePropertyListLazilyFromData:atCursor:  
length:mutableContainers: **class method 8**