NSPropertyListSerialization Class Reference

Cocoa > Data Management

2008-06-09

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

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

NSPropertyListSerialization Class Reference 5

Overview 5
Tasks 5
  Serializing a Property List 5
  Deserializing a Property List 6
  Validating a Property List 6
Class Methods 6
  dataFromPropertyList:format:errorDescription: 6
  propertyList:isValidForFormat: 7
  propertyListFromData:mutabilityOption:format:errorDescription: 7
Constants 8
  NSPropertyListMutabilityOptions 8
  NSPropertyListFormat 9

Document Revision History 11

Index 13

NSPropertyListSerialization Class Reference

Inherits from	NSObject
Conforms to	NSObject (NSObject)
Framework	/System/Library/Frameworks/Foundation.framework
Availability	Available in Mac OS X v10.2 and later.
Declared in	NSPropertyList.h
Companion guides	Archives and Serializations Programming Guide for Cocoa Property List Programming Guide
Related sample code	EnhancedAudioBurn iSpend People Sketch-112 Squiggles

Overview

The `NSPropertyListSerialization` class provides methods that convert property list objects to and from several serialized formats. Property list objects include `NSData`, `NSString`, `NSArray`, `NSDictionary`, `NSDate`, and `NSNumber` objects. These objects are toll-free bridged with their respective Core Foundation types (`CFData`, `CFString`, and so on). For more about toll-free bridging, see Interchangeable Data Types.

Property list serialization automatically takes account of endianness on different platforms—for example, you can correctly read on an Intel-based Macintosh a binary property list created on a PowerPC-based Macintosh.

Tasks

Serializing a Property List

`+ dataFromPropertyList:format:errorDescription:` (page 6)
Returns an `NSData` object containing a given property list in a specified format.

Deserializing a Property List

`+ propertyListFromData:mutabilityOption:format:errorDescription:` (page 7)
Returns a property list object corresponding to the representation in a given `NSData` object.

Validating a Property List

`+ propertyList:isValidForFormat:` (page 7)
Returns a Boolean value that indicates whether a given property list is valid for a given format.

Class Methods

dataFromPropertyList:format:errorDescription:

Returns an `NSData` object containing a given property list in a specified format.

```
+ (NSData *)dataFromPropertyList:(id)plist format:(NSPropertyListFormat)format
    errorDescription:(NSString **)errorString
```

Parameters

plist

A property list object. *plist* must be a kind of `NSData`, `NSString`, `NSNumber`, `NSDate`, `NSArray`, or `NSDictionary` object. Container objects must also contain only these kinds of objects.

format

A property list format. Possible values for *format* are described in NSPropertyListFormat (page 9).

errorString

Upon return, if the conversion is successful, *errorString* is `nil`. If the conversion fails, upon return contains a string describing the nature of the error. If you receive a string, you must release it.

Return Value
An `NSData` object containing *plist* in the format specified by *format*.

Special Considerations

Unlike the normal memory management rules for Cocoa, strings returned in *errorString* need to be released by the caller.

Availability
Available in Mac OS X v10.2 and later.

See Also
`+ propertyListFromData:mutabilityOption:format:errorDescription:` (page 7)

Related Sample Code
EnhancedAudioBurn

iSpend

People

SpotlightFortunes

Squiggles

Declared In

`NSPropertyList.h`

propertyList:isValidForFormat:

Returns a Boolean value that indicates whether a given property list is valid for a given format.

```
+ (BOOL)propertyList:(id)plist isValidForFormat:(NSPropertyListFormat)format
```

Parameters

plist

A property list object.

format

A property list format. Possible values for *format* are listed in NSPropertyListFormat (page 9).

Return Value

YES if *plist* is a valid property list in format *format*, otherwise NO.

Availability

Available in Mac OS X v10.2 and later.

Declared In

`NSPropertyList.h`

propertyListFromData:mutabilityOption:format:errorDescription:

Returns a property list object corresponding to the representation in a given NSData object.

```
+ (id)propertyListFromData:(NSData *)data
    mutabilityOption:(NSPropertyListMutabilityOptions)opt
    format:(NSPropertyListFormat *)format errorDescription:(NSString **)errorString
```

Parameters

data

A data object containing a serialized property list.

opt

Determines whether the property list's contents are created as mutable objects, where possible. Possible values are described in NSPropertyListMutabilityOptions (page 8).

format

If the property list is valid, upon return contains the format. *format* can be NULL, in which case the property list format is not returned. Possible values are described in NSPropertyListFormat (page 9).

errorString

Upon return, if the conversion is successful, *errorString* is nil. If the conversion fails, upon return contains a string describing the nature of the error. If you receive a string, you must release it.

Return Value

A property list object corresponding to the representation in *data*. If data is not in a supported format, returns nil.

Special Considerations

Unlike the normal memory management rules for Cocoa, strings returned in *errorString* need to be released by the caller.

Availability

Available in Mac OS X v10.2 and later.

See Also

`+ dataFromPropertyList:format:errorDescription:` (page 6)

Related Sample Code

AlbumToSlideshow

iSpend

People

Sketch-112

Squiggles

Declared In

`NSPropertyList.h`

Constants

NSPropertyListMutabilityOptions

These constants specify mutability options in property lists.

```
typedef enum {
    NSPropertyListImmutable = kCFPropertyListImmutable,
    NSPropertyListMutableContainers = kCFPropertyListMutableContainers,
    NSPropertyListMutableContainersAndLeaves =
kCFPropertyListMutableContainersAndLeaves
} NSPropertyListMutabilityOptions;
```

Constants

`NSPropertyListImmutable`

Causes the returned property list to contain immutable objects.

Available in Mac OS X v10.2 and later.

Declared in `NSPropertyList.h`.

`NSPropertyListMutableContainers`

Causes the returned property list to have mutable containers but immutable leaves.

Available in Mac OS X v10.2 and later.

Declared in `NSPropertyList.h`.

`NSPropertyListMutableContainersAndLeaves`

Causes the returned property list to have mutable containers and leaves.

Available in Mac OS X v10.2 and later.

Declared in `NSPropertyList.h`.

Availability

Available in Mac OS X v10.2 and later.

Declared In
`NSPropertyList.h`

NSPropertyListFormat

These constants are used to specify a property list serialization format.

```
typedef enum {
    NSPropertyListOpenStepFormat = kCFPropertyListOpenStepFormat,
    NSPropertyListXMLFormat_v1_0 = kCFPropertyListXMLFormat_v1_0,
    NSPropertyListBinaryFormat_v1_0 = kCFPropertyListBinaryFormat_v1_0
} NSPropertyListFormat;
```

Constants

`NSPropertyListOpenStepFormat`
: Specifies the old-style ASCII property list format inherited from the OpenStep APIs.

> **Important:** The `NSPropertyListOpenStepFormat` constant is not supported for writing. It can be used only for reading old-style property lists.

Available in Mac OS X v10.2 and later.

Declared in `NSPropertyList.h`.

`NSPropertyListXMLFormat_v1_0`
: Specifies the XML property list format.

Available in Mac OS X v10.2 and later.

Declared in `NSPropertyList.h`.

`NSPropertyListBinaryFormat_v1_0`
: Specifies the binary property list format.

Available in Mac OS X v10.2 and later.

Declared in `NSPropertyList.h`.

Availability
Available in Mac OS X v10.2 and later.

Declared In
`NSPropertyList.h`

Document Revision History

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

Date	Notes
2008-06-09	Updated dataFromPropertyList: and propertyListFromData: to remove note about Mac OS X v10.5 behavior.
2007-12-11	Updated to fix minor typos.
2007-03-06	Described update to dataFromPropertyList:format:errorDescription: in Mac OS X v10.5. Noted that NSPropertyListOpenStepFormat is not supported for writing.
2006-05-23	First publication of this content as a separate document.

Index

D

dataFromPropertyList:format:errorDescription: class method 6

N

NSPropertyListBinaryFormat_v1_0 constant 9
NSPropertyListFormat data type 9
NSPropertyListImmutable constant 8
NSPropertyListMutabilityOptions data type 8
NSPropertyListMutableContainers constant 8
NSPropertyListMutableContainersAndLeaves constant 8
NSPropertyListOpenStepFormat constant 9
NSPropertyListXMLFormat_v1_0 constant 9

P

propertyList:isValidForFormat: class method 7
propertyListFromData:mutabilityOption:format:errorDescription: class method 7