CFPropertyList Reference

Core Foundation

2006-02-07

Contents

CFPropertyList Reference

Derived From:	CFType
Framework:	CoreFoundation/CoreFoundation.h
Declared in	CFBase.h CFPropertyList.h
Companion guides	Property List Programming Topics for Core Foundation XML Programming Topics for Core Foundation

Overview

CFPropertyList provides functions that convert property list objects to and from several serialized formats such as XML. The `CFPropertyListRef` (page 11) type that denotes CFPropertyList objects is an abstract type for property list objects. Depending on the contents of the XML data used to create the property list, `CFPropertyListRef` can be any of the property list objects: CFData, CFString, CFArray, CFDictionary, CFDate, CFBoolean, and CFNumber. Note that if you use a property list to generate XML, the keys of any dictionaries in the property list must be CFString objects.

It is important to understand that CFPropertyList provides an abstraction for all the property list types—you can think of CFPropertyList in object-oriented terms as being the superclass of CFString, CFNumber, CFDictionary, and so on. When a Core Foundation function returns a `CFPropertyListRef`, it means that the value may be any of the property list types. For example, `CFPreferencesCopyAppValue` returns a `CFPropertyListRef`. This means that the value returned can be a CFString object, a CFNumber object, a CFDictionary object, and so on again. You can use `CFGetTypeID` to determine what type of object a property list value is.

You use one of the `CFPropertyListCreate...` functions to create a property list object given an existing property list object, raw XML data (as in a file), or a stream. You can also convert a property list object to XML using the `CFPropertyListCreateXMLData` (page 9) function. You use the `CFPropertyListWriteToStream` (page 10) function to write a property list to an output stream, and validate a property list object using the `CFPropertyListIsValid` (page 9) function. CFPropertyList properly takes care of endian issues—a property list (whether represented by a stream, XML, or a CFData object) created on a PowerPC-based Macintosh is correctly interpreted on an Intel-based Macintosh, and vice versa.

For code examples illustrating how to read and write property list files, see *Property List Programming Topics for Core Foundation* and in particular Saving and Restoring Property Lists.

Functions by Task

Creating a Property List

`CFPropertyListCreateDeepCopy` (page 6)
: Recursively creates a copy of a given property list.

`CFPropertyListCreateFromXMLData` (page 8)
: Creates a property list using the specified XML or binary property list data.

`CFPropertyListCreateFromStream` (page 7)
: Creates a property list using data from a stream.

Exporting a Property List

`CFPropertyListCreateXMLData` (page 9)
: Creates an XML representation of the specified property list.

`CFPropertyListWriteToStream` (page 10)
: Writes the bytes of a property list serialization out to a stream.

Validating a Property List

`CFPropertyListIsValid` (page 9)
: Determines if a property list is valid.

Functions

CFPropertyListCreateDeepCopy

Recursively creates a copy of a given property list.

```
CFPropertyListRef CFPropertyListCreateDeepCopy (
   CFAllocatorRef allocator,
   CFPropertyListRef propertyList,
   CFOptionFlags mutabilityOption
);
```

Parameters

allocator
: The allocator to use to allocate memory for the new property list. Pass `NULL` or `kCFAllocatorDefault` to use the current default allocator.

propertyList
: The property list to copy. This may be any of the standard property list objects, for example a CFArray or a CFDictionary object.

mutabilityOption
: A constant that specifies the degree of mutability of the returned property list. See Property List Mutability Options (page 12) for descriptions of possible values.

Return Value
A new property list that is a copy of *propertyList*. Ownership follows the Create Rule.

Discussion
Recursively creates a copy of the given property list so nested arrays and dictionaries are copied as well as the top-most container.

Availability
Available in CarbonLib v1.1 and later.

Available in Mac OS X v10.0 and later.

Related Sample Code
MoreIsBetter

MoreSCF

QISA

Declared In
`CFPropertyList.h`

CFPropertyListCreateFromStream

Creates a property list using data from a stream.

```
CFPropertyListRef CFPropertyListCreateFromStream (
   CFAllocatorRef allocator,
   CFReadStreamRef stream,
   CFIndex streamLength,
   CFOptionFlags mutabilityOption,
   CFPropertyListFormat *format,
   CFStringRef *errorString
);
```

Parameters

allocator
: The allocator to use to allocate memory for the new property list. Pass `NULL` or `kCFAllocatorDefault` to use the current default allocator.

stream
: The stream whose data contains the content. The stream must be opened and configured—this function simply reads bytes from the stream. The stream may contain any supported property list type (see Property List Formats (page 11)).

streamLength
: The number of bytes to read. If `0`, this function will read to the end of the stream.

mutabilityOption
: A constant that specifies the degree of mutability for the returned property list. See Property List Mutability Options (page 12) for descriptions of possible values.

format
: A constant that specifies the format of the property list. See Property List Formats (page 11) for possible values.

errorString

On return, `NULL` if the conversion is successful, otherwise a string that describes the nature of the error. Error messages are not localized, but may be in the future, so they are not suitable for comparison.

Pass `NULL` if you do not wish to receive an error string. Ownership follows the Create Rule.

Return Value

A new property list initialized with the data contained in *stream*. Ownership follows the Create Rule.

Discussion

This function simply reads bytes from *stream* starting at the current location to the end, which is expected to be the end of the property list, or up to the number of bytes specified by *streamLength* if it is not `0`.

Availability

Available in Mac OS X v10.2 and later.

Declared In

`CFPropertyList.h`

CFPropertyListCreateFromXMLData

Creates a property list using the specified XML or binary property list data.

```
CFPropertyListRef CFPropertyListCreateFromXMLData (
   CFAllocatorRef allocator,
   CFDataRef xmlData,
   CFOptionFlags mutabilityOption,
   CFStringRef *errorString
);
```

Parameters

allocator

The allocator to use to allocate memory for the new property list. Pass `NULL` or `kCFAllocatorDefault` to use the current default allocator.

data

The raw bytes to convert into a property list. The bytes may be the content of an XML file or of a binary property list (see Property List Formats (page 11)).

mutabilityOption

A constant that specifies the degree of mutability for the returned property list. See Property List Mutability Options (page 12) for descriptions of possible values.

errorString

On return, `NULL` if the conversion is successful, otherwise a string that describes the nature of the error. Error messages are not localized, but may be in the future, so they are not currently suitable for comparison.

Pass `NULL` if you do not wish to receive an error string. Ownership follows the Create Rule.

Return Value

A new property list if the conversion is successful, otherwise `NULL`. Ownership follows the Create Rule.

Availability

Available in CarbonLib v1.0 and later.

Available in Mac OS X v10.0 and later.

Related Sample Code

BSDLLCTest

HID Utilities Source

MoreIsBetter

QISA

StickiesExample

Declared In

`CFPropertyList.h`

CFPropertyListCreateXMLData

Creates an XML representation of the specified property list.

```
CFDataRef CFPropertyListCreateXMLData (
   CFAllocatorRef allocator,
   CFPropertyListRef propertyList
);
```

Parameters

allocator

The allocator to use to allocate memory for the new data object. Pass `NULL` or `kCFAllocatorDefault` to use the current default allocator.

propertyList

The property list to convert. This may be any of the standard property list objects, for example a CFArray or a CFDictionary object.

Return Value

A CFData object containing the XML data. Ownership follows the Create Rule.

Availability

Available in CarbonLib v1.0 and later.

Available in Mac OS X v10.0 and later.

Related Sample Code

BSDLLCTest

CFPrefTopScores

MoreIsBetter

QISA

StickiesExample

Declared In

`CFPropertyList.h`

CFPropertyListIsValid

Determines if a property list is valid.

```
Boolean CFPropertyListIsValid (
   CFPropertyListRef plist,
   CFPropertyListFormat format
);
```

Parameters

plist

The property list to validate.

format

A constant that specifies the allowable format of *plist*. See Property List Formats (page 11) for possible values.

Return Value

`true` if the object graph rooted at *plist* is a valid property list graph—that is, the property list contains no cycles, only contains property list objects, and all dictionary keys are strings; otherwise `false`.

Discussion

The debugging library version of this function prints out some useful messages.

Availability

Available in Mac OS X v10.2 and later.

Declared In

`CFPropertyList.h`

CFPropertyListWriteToStream

Writes the bytes of a property list serialization out to a stream.

```
CFIndex CFPropertyListWriteToStream (
   CFPropertyListRef propertyList,
   CFWriteStreamRef stream,
   CFPropertyListFormat format,
   CFStringRef *errorString
);
```

Parameters

propertyList

The property list to write out.

stream

The stream to write to. The stream must be opened and configured—this function simply writes bytes to the stream.

format

A constant that specifies the format used to write *propertyList*. See Property List Formats (page 11) for possible values.

errorString

On return, `NULL` if the conversion is successful, otherwise a string that describes the nature of the errors. Error messages are not localized, but may be in the future, so they are not currently suitable for comparison.

Pass `NULL` if you do not wish to receive an error string. Ownership follows the Create Rule.

Return Value
The number of bytes written, or `0` if an error occurred. If `0` is returned, *`errorString`* will contain an error message.

Discussion
This function leaves the stream open after reading the content. When reading a property list, this function expects the reading stream to end wherever the writing ended, so that the end of the property list data can be identified.

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

Declared In
`CFPropertyList.h`

Data Types

CFPropertyListRef

A reference to a CFPropertyList object.

```
typedef CFTypeRef CFPropertyListRef;
```

Discussion
This is an abstract type for property list objects. The return value of the `CFPropertyListCreateFromXMLData` function depends on the contents of the given XML data. `CFPropertyListRef` can be a reference to any of the property list objects: CFData, CFString, CFArray, CFDictionary, CFDate, CFBoolean, and CFNumber.

Availability
Available in Mac OS X v10.0 and later.

Declared In
`CFBase.h`

Constants

Property List Formats

Specifies the format of a property list.

```
enum CFPropertyListFormat {
    kCFPropertyListOpenStepFormat = 1,
    kCFPropertyListXMLFormat_v1_0 = 100,
    kCFPropertyListBinaryFormat_v1_0 = 200
};
typedef enum CFPropertyListFormat CFPropertyListFormat;
```

Constants

`kCFPropertyListOpenStepFormat`

OpenStep format (use of this format is discouraged).

Available in Mac OS X v10.2 and later.

Declared in `CFPropertyList.h`.

`kCFPropertyListXMLFormat_v1_0`

XML format version 1.0.

Available in Mac OS X v10.2 and later.

Declared in `CFPropertyList.h`.

`kCFPropertyListBinaryFormat_v1_0`

Binary format version 1.0.

Available in Mac OS X v10.2 and later.

Declared in `CFPropertyList.h`.

Property List Mutability Options

Option flags that determine the degree of mutability of newly created property lists.

```
enum CFPropertyListMutabilityOptions {
    kCFPropertyListImmutable = 0,
    kCFPropertyListMutableContainers = 1,
    kCFPropertyListMutableContainersAndLeaves = 2
};
typedef enum CFPropertyListMutabilityOptions CFPropertyListMutabilityOptions;
```

Constants

`kCFPropertyListImmutable`

Specifies that the property list should be immutable.

Available in Mac OS X v10.0 and later.

Declared in `CFPropertyList.h`.

`kCFPropertyListMutableContainers`

Specifies that the property list should have mutable containers but immutable leaves.

Available in Mac OS X v10.0 and later.

Declared in `CFPropertyList.h`.

`kCFPropertyListMutableContainersAndLeaves`

Specifies that the property list should have mutable containers and mutable leaves.

Available in Mac OS X v10.0 and later.

Declared in `CFPropertyList.h`.

Document Revision History

This table describes the changes to *CFPropertyList Reference*.

Date	Notes
2006-02-07	Clarified endian safeness for property lists.
2005-12-06	Made minor changes to conform to reference consistency guidelines.
2005-11-09	Added further links to "Property Lists" document, which contains code samples showing how to read and write property lists.
2005-08-11	Corrected descriptions of CFPropertyListCreateFromStream and CFPropertyListCreateFromXMLData, and minor typographical errors.
2005-04-29	Moved Introduction to new Introduction page.
2004-04-01	Noted where error string parameters may be `NULL` in `CFPropertyListCreateFromXMLData`, `CFPropertyListCreateFromStream`, and `CFPropertyListWriteToStream`.
2003-01-01	First version of this document.

Index