
CFXMLTree Reference

Data Management



2008-10-15



Apple Inc.
© 2003, 2008 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, Carbon, Mac, and Mac OS 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

CFXMLTree Reference 5

Overview	5
Functions	5
CFXMLCreateStringByEscapingEntities	5
CFXMLCreateStringByUnescapingEntities	6
CFXMLTreeCreateFromData	7
CFXMLTreeCreateFromDataWithError	8
CFXMLTreeCreateWithDataFromURL	9
CFXMLTreeCreateWithNode	9
CFXMLTreeCreateXMLData	10
CFXMLTreeGetNode	11
Data Types	11
CFXMLTreeRef	11
Constants	11
Error Dictionary Keys	11

Document Revision History 13

Index 15

CFXMLTree Reference

Derived From:	<i>CFTree Reference</i> : <i>CType Reference</i>
Framework:	CoreFoundation/CoreFoundation.h
Companion guide	XML Programming Topics for Core Foundation
Declared in	CFXMLNode.h CFXMLParser.h

Overview

A CFXMLTree object is simply a CFTree object whose context data is known to be a CFXMLNode object. CFXMLTree is derived from CFTree—you can pass CFXMLTree objects in all the CFTree functions. As such, a CFXMLTree object can be used to represent an entire XML document; the CFTree object provides the tree structure of the document, while the CFXMLNode objects identify and describe the nodes of the tree. An XML document can be parsed to a CFXMLTree object, and a CFXMLTree object can generate the data for the equivalent XML document. This opaque type is expected to be used in conjunction with CFXMLParser and CFXMLNode objects.

Functions

CFXMLCreateStringByEscapingEntities

Given a CFString object containing XML source with unescaped entities, returns a string with specified XML entities escaped.

```
CFStringRef CFXMLCreateStringByEscapingEntities(  
    CFAllocatorRef allocator,  
    CFStringRef string,  
    CFDictionaryRef entitiesDictionary,  
);
```

Parameters

allocator

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

string

Any CFString object that may contain XML source. This function translates any substring that is mapped to an entity in *entitiesDictionary* to the specified entity.

entitiesDictionary

Specifies the entities to be replaced. Dictionary keys should be the entity names (for example, “para” for ¶), and the values should be CFString objects containing the expansion. Pass NULL to indicate no entities other than the standard five.

Return Value

A CFString object derived from *string* with substrings identified in *entitiesDictionary* escaped to their corresponding entities. Ownership follows the Create Rule.

Discussion

The standard five predefined entities are automatically supported.

As an example of using this function, say you apply this function to string “Refer to ¶ 5 of the contract” with a key of “para” mapped to “¶” in *entitiesDictionary*. The resulting string is “Refer to ¶ 5 of the contract”.

Note: Currently, only the standard predefined entities are supported; passing NULL for *entitiesDictionary* is sufficient.

Availability

Available in Mac OS X v10.3 and later.

Declared In

CFXMLParser.h

CFXMLCreateStringByUnescapingEntities

Given a CFString object containing XML source with escaped entities, returns a string with specified XML entities unescaped.

```
CFStringRef CFXMLCreateStringByUnescapingEntities(
    CFAllocatorRef allocator,
    CFStringRef string,
    CFDictionaryRef entitiesDictionary,
);
```

Parameters*allocator*

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

string

Any CFString object that may contain XML source. This function translates any entity that is mapped to a substring in *entitiesDictionary* to the specified substring.

entitiesDictionary

Specifies the entities to be replaced. Dictionary keys should be the entity names (for example, “para” for ¶), and the values should be CFString objects containing the expansion. Pass NULL to indicate no entities other than the standard five.

Return Value

A CFString object derived from *string* with entities identified in *entitiesDictionary* unescaped to their corresponding substrings. Ownership follows the Create Rule.

Discussion

The standard five predefined entities are automatically supported.

As an example of using this function, say you apply this function to string “Refer to ¶ 5 of the contract” with a key of “para” mapped to “¶” in *entitiesDictionary*. The resulting string is “Refer to ¶ 5 of the contract”.

Note: Currently, only the standard predefined entities are supported; passing `NULL` for *entitiesDictionary* is sufficient.

Availability

Available in Mac OS X v10.3 and later.

Declared In

CFXMLParser.h

CFXMLTreeCreateFromData

Parses the given XML data and returns the resulting CFXMLTree object.

```
CFXMLTreeRef CFXMLTreeCreateFromData (
    CFAllocatorRef allocator,
    CFDataRef xmlData,
    CFURLRef dataSource,
    CFOptionFlags parseOptions,
    CFIndex versionOfNodes
);
```

Parameters

allocator

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

xmlData

The XML data you wish to parse.

dataSource

The URL from which the XML data was obtained. The URL is used to resolve any relative references found in *xmlData*. Pass `NULL` if a valid URL is unavailable.

parseOptions

Flags which control how the XML data will be parsed. See [Parsing Options](#) for the list of available options.

versionOfNodes

Determines which version of CFXMLNode objects are produced by the parser.

Return Value

A new CFXMLTree object containing the data from the specified XML document. Ownership follows the Create Rule.

Discussion

This function represents the high-level interface to the XML parser. This single function creates a parser for the specified XML data using the specified options. The parser creates and returns a CFXMLTree object that you can examine and modify with the CFTree functions or obtain the node using the [CFXMLTreeGetNode](#) (page 11) function and examine its attributes using CFXMLNode functions.

Availability

Available in CarbonLib v1.1 and later.

Available in Mac OS X v10.0 and later.

Declared In

CFXMLParser.h

CFXMLTreeCreateFromDataWithError

Parses the given XML data and returns the resulting CFXMLTree object and any error information.

```
CFXMLTreeRef CFXMLTreeCreateFromDataWithError (
    CFAllocatorRef allocator,
    CFDataRef xmlData,
    CFURLRef dataSource,
    CFOptionFlags parseOptions,
    CFIndex versionOfNodes
    CFDictionaryRef *errorDict
);
```

Parameters

allocator

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

xmlData

The XML data you wish to parse.

dataSource

The URL from which the XML data was obtained. The URL is used to resolve any relative references found in *xmlData*. Pass NULL if a valid URL is unavailable.

parseOptions

Flags which control how the XML data will be parsed. See [Parsing Options](#) for the list of available options.

versionOfNodes

Determines which version of CFXMLNode objects are produced by the parser. The current version is 1.

errorDict

Upon return, if an error occurs contains a CFDictionary object that describes the error. If no errors occur, this parameter is not changed. Pass NULL if you don't want error information. See ["Error Dictionary Keys"](#) (page 11) for a description of the key-value pairs in this dictionary. Ownership follows the Create Rule.

Return Value

A new CFXMLTree object containing the data from the specified XML document. Ownership follows the Create Rule.

Discussion

Use this function instead of [CFXMLTreeCreateFromData](#) (page 7) if you need access to XML parsing errors.

Availability

Available in Mac OS X v10.3 and later.

Declared In

CFXMLParser.h

CFXMLTreeCreateWithDataFromURL

Creates a new CFXMLTree object by loading the data to be parsed directly from a data source.

```
CFXMLTreeRef CFXMLTreeCreateWithDataFromURL (
    CFAllocatorRef allocator,
    CFURLRef dataSource,
    CFOptionFlags parseOptions,
    CFIndex versionOfNodes
);
```

Parameters

allocator

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

dataSource

The URL from which the XML data is obtained. The URL is used to resolve any relative references found in XML Data. Pass NULL if a valid URL is unavailable.

parseOptions

Flags which control how the XML data will be parsed. See [Parsing Options](#) for the list of available options.

versionOfNodes

Determines which version of CFXMLNode objects are produced by the parser.

Return Value

A new CFXMLTree object containing the data from the specified XML data source. Ownership follows the Create Rule.

Availability

Available in CarbonLib v1.1 and later.

Available in Mac OS X v10.0 and later.

Declared In

CFXMLParser.h

CFXMLTreeCreateWithNode

Creates a childless, parentless CFXMLTree object node for a CFXMLNode object.

```
CFXMLTreeRef CFXMLTreeCreateWithNode (
    CFAllocatorRef allocator,
    CFXMLNodeRef node
);
```

Parameters*allocator*

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

node

The `CFXMLNode` object to use when creating the new `CFXMLTree` object.

Return Value

A `CFXMLTree` object. Ownership follows the Create Rule.

Availability

Available in CarbonLib v1.1 and later.

Available in Mac OS X v10.0 and later.

Declared In

`CFXMLNode.h`

CFXMLTreeCreateXMLData

Generates an XML document from a `CFXMLTree` object which is ready to be written to permanent storage.

```
CFDataRef CFXMLTreeCreateXMLData (
    CFAllocatorRef allocator,
    CFXMLTreeRef xmlTree
);
```

Parameters*allocator*

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

xmlTree

The `CFXMLTree` object you wish to convert to an XML document.

Return Value

The XML data. Ownership follows the Create Rule.

Discussion

This function will *not* regenerate entity references replaced at the parse time (except those required for syntactic correctness). If you need this you must manually walk the tree and re-insert any entity references that should appear in the final output file.

Availability

Available in CarbonLib v1.1 and later.

Available in Mac OS X v10.0 and later.

Declared In

`CFXMLParser.h`

CFXMLTreeGetNode

Returns the node of a CFXMLTree object.

```
CFXMLNodeRef CFXMLTreeGetNode (
    CFXMLTreeRef xmlTree
);
```

Parameters

xmlTree

The CFXMLTree object whose node you wish to obtain.

Return Value

The node of *xmlTree*. Ownership follows the Get Rule.

Availability

Available in CarbonLib v1.1 and later.

Available in Mac OS X v10.0 and later.

Declared In

CFXMLNode.h

Data Types

CFXMLTreeRef

A reference to a CFXMLTree object.

```
typedef CFTreeRef CFXMLTreeRef;
```

Discussion

When using the high-level parser API, XML data is parsed to a special CFTree object which is simply a CFXMLTree object with known contexts and callbacks. The nodes of a CFXMLTree may be queried using the basic CFTree functions (to report on the structure of the tree itself), or via the functions here (to report on the XML contents of the nodes).

Availability

Available in Mac OS X v10.0 and later.

Declared In

CFXMLNode.h

Constants

Error Dictionary Keys

The keys used in an error dictionary returned by some functions to provide more information about XML parse errors.

```
const CFStringRef kCFXMLTreeErrorDescription;
const CFStringRef kCFXMLTreeErrorLineNumber;
const CFStringRef kCFXMLTreeErrorLocation;
const CFStringRef kCFXMLTreeErrorStatusCode;
```

Constants

`kCFXMLTreeErrorDescription`

Dictionary key whose value is a `CFString` containing a readable description of the error.

Available in Mac OS X v10.3 and later.

Declared in `CFXMLParser.h`.

`kCFXMLTreeErrorLineNumber`

Dictionary key whose value is a `CFNumber` containing the line number where the error was detected. This may not be the line number where the actual XML error is located.

Available in Mac OS X v10.3 and later.

Declared in `CFXMLParser.h`.

`kCFXMLTreeErrorLocation`

Dictionary key whose value is a `CFNumber` containing the byte location where the error was detected.

Available in Mac OS X v10.3 and later.

Declared in `CFXMLParser.h`.

`kCFXMLTreeErrorStatusCode`

Dictionary key whose value is a `CFNumber` containing the error status code. See *CFXMLParser Reference* for possible status code values.

Available in Mac OS X v10.3 and later.

Declared in `CFXMLParser.h`.

Discussion

These keys are used in the error dictionary returned by the [CFXMLTreeCreateFromDataWithError](#) (page 8) function.

Availability

Available in Mac OS X v10.3 and later.

Document Revision History

This table describes the changes to *CFXMLTree Reference*.

Date	Notes
2008-10-15	Clarified descriptions of CFXMLCreateStringBy(Un)EscapingEntities functions and made minor corrections.
2006-02-07	Made formatting changes.
2003-08-01	Added descriptions of new Mac OS X v10.3 API.
2003-01-01	First version of this document.

REVISION HISTORY

Document Revision History

Index

C

CFXMLCreateStringByEscapingEntities **function** [5](#)
CFXMLCreateStringByUnescapingEntities **function** [6](#)
CFXMLTreeCreateFromData **function** [7](#)
CFXMLTreeCreateFromDataWithError **function** [8](#)
CFXMLTreeCreateWithDataFromURL **function** [9](#)
CFXMLTreeCreateWithNode **function** [9](#)
CFXMLTreeCreateXMLData **function** [10](#)
CFXMLTreeGetNode **function** [11](#)
CFXMLTreeRef **data type** [11](#)

E

Error Dictionary Keys [11](#)

K

kCFXMLTreeErrorDescription **constant** [12](#)
kCFXMLTreeErrorLineNumber **constant** [12](#)
kCFXMLTreeErrorLocation **constant** [12](#)
kCFXMLTreeErrorStatusCode **constant** [12](#)