CFAttributedString Reference

Core Foundation > Data Management

2009-05-06

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

Contents

CFAttributedString Reference

Derived From:	*CFType Reference*
Framework:	CoreFoundation/CoreFoundation.h
Declared in	CFAttributedString.h
Companion guides	Property List Programming Topics for Core Foundation Strings Programming Guide for Core Foundation Data Formatting Guide for Core Foundation

Overview

Instances of CFAttributedString manage character strings and associated sets of attributes (for example, font and kerning information) that apply to individual characters or ranges of characters in the string. CFAttributedString as defined in CoreFoundation provides the basic container functionality, while higher levels provide definitions for standard attributes, their values, and additional behaviors involving these. CFAttributedString represents an immutable string—use CFMutableAttributedString to create and manage an attributed string that can be changed after it has been created.

iPhone OS Note: While Core Foundation on iPhone OS contains CFAttributedString, there are no additions to the APIs in UIKit to add specific attributes such as font, style, or color, and there are no APIs to draw attributed strings.

CFAttributedString is not a "subclass" of CFString; that is, it does not respond to CFString function calls. CFAttributedString conceptually contains a CFString to which it applies attributes. This protects you from ambiguities caused by the semantic differences between simple and attributed string.

Attributes are identified by key/value pairs stored in CFDictionary objects. Keys must be CFString objects, while the corresponding values are CFType objects of an appropriate type. See the attribute constants in *NSAttributedString Application Kit Additions Reference* for standard attribute names.

Important: Attribute dictionaries set for an attributed string must always be created with kCFCopyStringDictionaryKeyCallbacks for their dictionary key callbacks and kCFTypeDictionaryValueCallBacks for their value callbacks; otherwise it's an error.

On Mac OS X, CFAttributedString is "toll-free bridged" with its Cocoa Foundation counterpart, NSAttributedString. This means that the Core Foundation type is interchangeable in function or method calls with the bridged Foundation object. Therefore, in a method where you see an `NSAttributedString *` parameter, you can pass in a `CFAttributedStringRef`, and in a function where you see a

CFAttributedStringRef parameter, you can pass in an NSAttributedString instance. This also applies to concrete subclasses of NSAttributedString. See Interchangeable Data Types for more information on toll-free bridging.

iPhone OS Note: NSAttributedString is not available on iPhone OS.

Functions by Task

Creating a CFAttributedString

CFAttributedStringCreate (page 7)
Creates an attributed string with specified string and attributes.

CFAttributedStringCreateCopy (page 7)
Creates an immutable copy of an attributed string.

CFAttributedStringCreateWithSubstring (page 8)
Creates a sub-attributed string from the specified range.

CFAttributedStringGetLength (page 11)
Returns the length of the attributed string in characters.

CFAttributedStringGetString (page 12)
Returns the string for an attributed string.

Accessing Attributes

CFAttributedStringGetAttribute (page 9)
Returns the value of a given attribute of an attributed string at a specified location.

CFAttributedStringGetAttributes (page 10)
Returns the attributes of an attributed string at a specified location.

CFAttributedStringGetAttributeAndLongestEffectiveRange (page 9)
Returns the value of a given attribute of an attributed string at a specified location.

CFAttributedStringGetAttributesAndLongestEffectiveRange (page 11)
Returns the attributes of an attributed string at a specified location.

Getting Attributed String Properties

CFAttributedStringGetTypeID (page 12)
Returns the type identifier for the CFAttributedString opaque type.

Functions

CFAttributedStringCreate

Creates an attributed string with specified string and attributes.

```
CFAttributedStringRef CFAttributedStringCreate (
   CFAllocatorRef alloc,
   CFStringRef str,
   CFDictionaryRef attributes
);
```

Parameters

alloc

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

str

A string that specifies the characters to use in the new attributed string. This value is copied.

attributes

A dictionary that contains the attributes to apply to the new attributed string. This value is copied.

Return Value

An attributed string that contains the characters from *str* and the attributes specified by *attributes*. The result is `NULL` if there was a problem in creating the attributed string. Ownership follows the Create Rule.

Discussion

Note that both the string and the attributes dictionary are copied. The specified attributes are applied to the whole string. If you want to apply different attributes to different ranges of the string, you should use a mutable attributed string.

Availability

Available in Mac OS X v10.4 and later.

Related Sample Code

CoreTextArc

CoreTextTest

Declared In

`CFAttributedString.h`

CFAttributedStringCreateCopy

Creates an immutable copy of an attributed string.

```
CFAttributedStringRef CFAttributedStringCreateCopy (
   CFAllocatorRef alloc,
   CFAttributedStringRef aStr
);
```

Parameters

alloc

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

aStr

The attributed string to copy.

Return Value

An immutable attributed string with characters and attributes identical to those of *aStr*. Returns `NULL` if there was a problem copying the object. Ownership follows the Create Rule.

Availability

Available in Mac OS X v10.4 and later.

Declared In

`CFAttributedString.h`

CFAttributedStringCreateWithSubstring

Creates a sub-attributed string from the specified range.

```
CFAttributedStringRef CFAttributedStringCreateWithSubstring (
   CFAllocatorRef alloc,
   CFAttributedStringRef aStr,
   CFRange range
);
```

Parameters

alloc

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

theString

The attributed string to copy.

range

The range of the attributed string to copy. *range* must not exceed the bounds of *aStr*.

Return Value

A new attributed string whose string and attributes are copied from from the specified range of the supplied attributed string. Returns `NULL` if there was a problem copying the object. Ownership follows the Create Rule.

Availability

Available in Mac OS X v10.4 and later.

Declared In

`CFAttributedString.h`

CFAttributedStringGetAttribute

Returns the value of a given attribute of an attributed string at a specified location.

```
CFTypeRef CFAttributedStringGetAttribute (
   CFAttributedStringRef aStr,
   CFIndex loc,
   CFStringRef attrName,
   CFRange *effectiveRange
);
```

Parameters

str

The attributed string to examine.

loc

The location in *str* at which to determine the attributes. *loc* must not exceed the bounds of *str*.

attrName

The name of the attribute whose value you want to determine.

effectiveRange

If not NULL, upon return contains a range including *loc* over which exactly the same set of attributes apply as at *loc*.

Return Value

The value of the specified attribute at the specified location in *str*. Ownership follows the Get Rule.

Discussion

For performance reasons, a range returned in *effectiveRange* is not necessarily the maximal range. If you need the maximum range, you should use CFAttributedStringGetAttributeAndLongestEffectiveRange (page 9).

Availability

Available in Mac OS X v10.4 and later.

Declared In

CFAttributedString.h

CFAttributedStringGetAttributeAndLongestEffectiveRange

Returns the value of a given attribute of an attributed string at a specified location.

```
CFTypeRef CFAttributedStringGetAttributeAndLongestEffectiveRange (
   CFAttributedStringRef aStr,
   CFIndex loc,
   CFStringRef attrName,
   CFRange inRange,
   CFRange *longestEffectiveRange
);
```

Parameters

str

The attributed string to examine.

loc

The location in *str* at which to determine the attributes. It is a programming error for *loc* to specify a location outside the bounds of *str*.

attrName

The name of the attribute whose value you want to determine.

inRange

The range in *str* within which you want to find the longest effective range of the attributes at *loc*. *inRange* must not exceed the bounds of *str*.

effectiveRange

If not NULL, upon return contains the maximal range within *inRange* over which the exact same set of attributes apply. The returned range is clipped to *inRange*.

Return Value
A dictionary that contains the attributes of *str* at the specified location. Ownership follows the Get Rule.

Availability
Available in Mac OS X v10.4 and later.

Declared In
CFAttributedString.h

CFAttributedStringGetAttributes

Returns the attributes of an attributed string at a specified location.

```
CFDictionaryRef CFAttributedStringGetAttributes (
   CFAttributedStringRef aStr,
   CFIndex loc,
   CFRange *effectiveRange
);
```

Parameters

str

The attributed string to examine.

loc

The location in *str* at which to determine the attributes. *loc* must not exceed the bounds of *str*.

effectiveRange

If not NULL, upon return contains a range including *loc* over which exactly the same set of attributes apply as at *loc*.

Return Value
A dictionary that contains the attributes of *str* at the specified location. Ownership follows the Get Rule.

Discussion
For performance reasons, a range returned in *effectiveRange* is not necessarily the maximal range. If you need the maximum range, you should use CFAttributedStringGetAttributesAndLongestEffectiveRange (page 11).

Note that the returned attribute dictionary might change in unpredictable ways if the attributed string is edited after this call. If you want to preserve the state of the dictionary, you should make an actual copy of it rather than just retaining it. In addition, you should make no assumptions about the relationship of the actual dictionary returned by this call and the dictionary originally used to set the attributes, other than the fact that the values stored in the dictionaries will be identical (that is, ==) to those originally specified.

Availability
Available in Mac OS X v10.4 and later.

Declared In
`CFAttributedString.h`

CFAttributedStringGetAttributesAndLongestEffectiveRange

Returns the attributes of an attributed string at a specified location.

```
CFDictionaryRef CFAttributedStringGetAttributesAndLongestEffectiveRange (
   CFAttributedStringRef aStr,
   CFIndex loc,
   CFRange inRange,
   CFRange *longestEffectiveRange
);
```

Parameters

str

The attributed string to examine.

loc

The location in *str* at which to determine the attributes. *loc* must not exceed the bounds of *str*.

inRange

The range in *str* within to find the longest effective range of the attributes at *loc*. *inRange* must not exceed the bounds of *str*.

effectiveRange

If not `NULL`, upon return contains the maximal range within *inRange* over which the exact same set of attributes apply. The returned range is clipped to *inRange*.

Return Value
A dictionary that contains the attributes of *str* at the specified location. Ownership follows the Get Rule.

Availability
Available in Mac OS X v10.4 and later.

Declared In
`CFAttributedString.h`

CFAttributedStringGetLength

Returns the length of the attributed string in characters.

```
CFIndex CFAttributedStringGetLength (
   CFAttributedStringRef aStr
);
```

Parameters

str

The attributed string to examine.

Return Value
The length of the attributed string in characters; this is the same as `CFStringGetLength(CFAttributedStringGetString(aStr))`.

Availability
Available in Mac OS X v10.4 and later.

Related Sample Code
CoreTextTest

Declared In
`CFAttributedString.h`

CFAttributedStringGetString

Returns the string for an attributed string.

```
CFStringRef CFAttributedStringGetString (
   CFAttributedStringRef aStr
);
```

Parameters

aStr

The attributed string to examine.

Return Value
An immutable string containing the characters from *aStr*, or `NULL` if there was a problem creating the object. Ownership follows the Get Rule.

Discussion
For performance reasons, the string returned will often be the backing store of the attributed string, and it might therefore change if the attributed string is edited. However, this is an implementation detail, and you should not rely on this behavior.

Availability
Available in Mac OS X v10.4 and later.

Declared In
`CFAttributedString.h`

CFAttributedStringGetTypeID

Returns the type identifier for the CFAttributedString opaque type.

```
CFTypeID CFAttributedStringGetTypeID (
   void
);
```

Return Value
The type identifier for the CFAttributedString opaque type.

Discussion
CFMutableAttributedString objects have the same type identifier as CFAttributedString objects.

Availability
Available in Mac OS X v10.4 and later.

Declared In
`CFAttributedString.h`

Data Types

CFAttributedStringRef

A reference to a CFAttributedString object.

```
typedef const struct __CFAttributedString *CFAttributedStringRef;
```

Discussion
The `CFAttributedStringRef` type refers to an object that combines a CFString object with a collection of attributes that specify how the characters in the string should be displayed. CFAttributedString is an opaque type that defines the characteristics and behavior of CFAttributedString objects.

Values of type `CFAttributedStringRef` may refer to immutable or mutable strings, as CFMutableAttributedString objects respond to all functions intended for immutable CFAttributedString objects. Functions which accept `CFAttributedStringRef` values, and which need to hold on to the values immutably, should call `CFAttributedStringCreateWithSubstring` (page 8) (instead of `CFRetain`) to do so.

Availability
Available in Mac OS X v10.4 and later.

Declared In
`CFAttributedString.h`

Document Revision History

This table describes the changes to *CFAttributedString Reference*.

Date	Notes
2009-05-06	Added noted that CFAttributedString is not toll-free bridged to NSAttributedString on iPhone.
2008-07-01	Added note to introduction about attributed string support in iPhone OS. Enhanced warning about attribute dictionaries requiring key and value callbacks.
2005-12-06	Made minor changes to text to conform to reference consistency guidelines.
2005-11-09	Updated link in Companion Documents.
2005-04-29	First version of this document.

Document Revision History

Index

C