CTTypesetter Reference

Carbon > Text & Fonts

2007-05-01

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

CTTypesetter Reference

Derived From:	CFType
Framework:	ApplicationServices/CoreText
Declared in	CTTypesetter.h

Overview

The CTTypesetter opaque type represents a typesetter, which performs line layout.

Line layout includes word wrapping, hyphenation, and line breaking in either vertical or horizontal rectangles. A typesetter object takes as input an attributed string and produces a line of typeset glyphs (composed into glyph runs) in a CTLine object. The typesetter performs character-to-glyph encoding, glyph ordering, and positional operations, such as kerning, tracking, and baseline adjustments. If multiline layout is needed, it is performed by a framesetter object, which calls into the typesetter to generate the typeset lines to fill the frame.

A framesetter encapsulates a typesetter and provides a reference to it as a convenience, but a caller may also choose to create a freestanding typesetter.

Functions by Task

Creating a Typesetter

`CTTypesetterCreateWithAttributedString` (page 6)
: Creates an immutable typesetter object using an attributed string.

`CTTypesetterCreateWithAttributedStringAndOptions` (page 7)
: Creates an immutable typesetter object using an attributed string and a dictionary of options.

Creating Lines

`CTTypesetterCreateLine` (page 6)
: Creates an immutable line from the typesetter.

Breaking Lines

`CTTypesetterSuggestLineBreak` (page 8)
Suggests a contextual line breakpoint based on the width provided.

`CTTypesetterSuggestClusterBreak` (page 8)
Suggests a cluster line breakpoint based on the width provided.

Getting the Type Identifier

`CTTypesetterGetTypeID` (page 7)
Returns the Core Foundation type identifier of the typesetter object.

Functions

CTTypesetterCreateLine

Creates an immutable line from the typesetter.

```
CTLineRef CTTypesetterCreateLine( CTTypesetterRef typesetter, CFRange stringRange
 );
```

Parameters

typesetter
The typesetter that creates the line. This parameter is required and cannot be set to `NULL`.

stringRange
The string range on which the line is based. If the length portion of range is set to `0`, then the typesetter continues to add glyphs to the line until it runs out of characters in the string. The location and length of the range must be within the bounds of the string, or the call will fail.

Return Value
A reference to a CTLine object if the call was successful; otherwise, `NULL`.

Discussion
The resultant line consists of glyphs in the correct visual order, ready to draw.

Availability
Available in Mac OS X v10.5 and later.

Declared In
`CTTypesetter.h`

CTTypesetterCreateWithAttributedString

Creates an immutable typesetter object using an attributed string.

```
CTTypesetterRef CTTypesetterCreateWithAttributedString( CFAttributedStringRef string
 );
```

Parameters

string

The attributed string to typeset. This parameter must be filled in with a valid CFAttributedString object.

Return Value

A reference to a CTTypesetter object if the call was successful; otherwise, `NULL`.

Discussion

The resultant typesetter can be used to create lines, perform line breaking, and do other contextual analysis based on the characters in the string.

Availability

Available in Mac OS X v10.5 and later.

Declared In

`CTTypesetter.h`

CTTypesetterCreateWithAttributedStringAndOptions

Creates an immutable typesetter object using an attributed string and a dictionary of options.

```
CTTypesetterRef CTTypesetterCreateWithAttributedStringAndOptions(
CFAttributedStringRef string, CFDictionaryRef options );
```

Parameters

string

The attributed string to typeset. This parameter must be filled in with a valid CFAttributedString object.

options

A dictionary of typesetter options, or `NULL` if there are none.

Return Value

A reference to a CTTypesetter object if the call was successful; otherwise, `NULL`.

Discussion

The resultant typesetter can be used to create lines, perform line breaking, and do other contextual analysis based on the characters in the string.

Availability

Available in Mac OS X v10.5 and later.

Declared In

`CTTypesetter.h`

CTTypesetterGetTypeID

Returns the Core Foundation type identifier of the typesetter object.

```
CFTypeID CTTypesetterGetTypeID( void );
```

Availability
Available in Mac OS X v10.5 and later.

Declared In
`CTTypesetter.h`

CTTypesetterSuggestClusterBreak

Suggests a cluster line breakpoint based on the width provided.

```
CFIndex CTTypesetterSuggestClusterBreak( CTTypesetterRef typesetter, CFIndex
startIndex, double width );
```

Parameters

typesetter
The typesetter that creates the line. This parameter is required and cannot be set to `NULL`.

startIndex
The starting point for the typographic cluster-break calculations. The break calculations include the character starting at *startIndex*.

width
The requested typographic cluster-break width.

Return Value
A count of the characters from *startIndex* that would cause the cluster break. The value returned can be used to construct a character range for `CTTypesetterCreateLine` (page 6).

Discussion
This cluster break is similar to a character break, except that it does not break apart linguistic clusters. No other contextual analysis is done. This can be used by the caller to implement a different line-breaking scheme, such as hyphenation. A typographic cluster break can also be triggered by a hard-break character in the stream.

Availability
Available in Mac OS X v10.5 and later.

Declared In
`CTTypesetter.h`

CTTypesetterSuggestLineBreak

Suggests a contextual line breakpoint based on the width provided.

```
CFIndex CTTypesetterSuggestLineBreak( CTTypesetterRef typesetter, CFIndex startIndex,
 double width );
```

Parameters

typesetter
The typesetter that creates the line. This parameter is required and cannot be set to `NULL`.

startIndex

The starting point for the line-break calculations. The break calculations include the character starting at *startIndex*.

width

The requested line-break width.

Return Value

A count of the characters from *startIndex* that would cause the line break. The value returned can be used to construct a character range for `CTTypesetterCreateLine` (page 6).

Discussion

The line break can be triggered either by a hard-break character in the stream or by filling the specified width with characters.

Availability

Available in Mac OS X v10.5 and later.

Declared In

`CTTypesetter.h`

Data Types

CTTypesetterRef

A reference to a typesetter object.

```
typedef const struct __CTTypesetter *CTTypesetterRef;
```

Availability

Available in Mac OS X v10.5 and later.

Declared In

`CTTypesetter.h`

Constants

kCTTypesetterOptionDisableBidiProcessing

Disables bidirectional processing.

```
const CFStringRef kCTTypesetterOptionDisableBidiProcessing;
```

Constants

`kCTTypesetterOptionDisableBidiProcessing`

Disables bidirectional processing.

Available in Mac OS X v10.5 and later.

Declared in `CTTypesetter.h`.

Discussion

Value must be a CFBoolean object. Default value is `False`. Normally, typesetting applies the Unicode Bidirectional Algorithm as described in Unicode Standard Annex #9. If a typesetter is created with this option set to `True`, no directional reordering is performed and any directional control characters are ignored.

Declared In

`CTTypesetter.h`

Document Revision History

This table describes the changes to *CTTypesetter Reference*.

Date	Notes
2007-05-01	New document that describes the Core Text opaque type used to represent a typesetter, which performs line layout.

Index