
CTFrame Reference

[Carbon > Text & Fonts](#)



2007-05-01



Apple Inc.
© 2007 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

CTFrame Reference 5

Overview	5
Functions by Task	5
Getting Frame Data	5
Getting Lines	5
Drawing the Frame	6
Getting the Type Identifier	6
Functions	6
CTFrameDraw	6
CTFrameGetFrameAttributes	6
CTFrameGetLineOrigins	7
CTFrameGetLines	7
CTFrameGetPath	8
CTFrameGetStringRange	8
CTFrameGetTypeID	8
CTFrameGetVisibleStringRange	9
Data Types	9
CTFrameRef	9
Constants	9
CTFrameProgression	9
kCTFrameProgressionAttributeName	10

Document Revision History 11

Index 13

CTFrame Reference

Derived From:	CType
Framework:	ApplicationServices/CoreText
Declared in	CTFrame.h

Overview

The CTFrame opaque type represents a frame containing multiple lines of text. The frame object is the output resulting from the text-framing process performed by a framesetter object.

You can draw the entire text frame directly into the current graphic context. The frame object contains an array of line objects that can be retrieved for individual rendering or to get glyph information.

Functions by Task

Getting Frame Data

[CTFrameGetStringRange](#) (page 8)

Returns the range of characters originally requested to fill the frame.

[CTFrameGetVisibleStringRange](#) (page 9)

Returns the range of characters that actually fit in the frame.

[CTFrameGetPath](#) (page 8)

Returns the path used to create the frame.

[CTFrameGetFrameAttributes](#) (page 6)

Returns the frame attributes used to create the frame.

Getting Lines

[CTFrameGetLines](#) (page 7)

Returns an array of lines stored in the frame.

[CTFrameGetLineOrigins](#) (page 7)

Copies a range of line origins for a frame.

Drawing the Frame

[CTFrameDraw](#) (page 6)

Draws an entire frame into a context.

Getting the Type Identifier

[CTFrameGetTypeID](#) (page 8)

Returns the type identifier for the CTFrame opaque type.

Functions

CTFrameDraw

Draws an entire frame into a context.

```
void CTFrameDraw( CTFrameRef frame, CGContextRef context );
```

Parameters

frame

The frame to draw.

context

The context in which to draw the frame.

Discussion

If both the frame and the context are valid, the frame is drawn in the context. This call can leave the context in any state and does not flush it after the draw operation.

Availability

Available in Mac OS X v10.5 and later.

Declared In

CTFrame.h

CTFrameGetFrameAttributes

Returns the frame attributes used to create the frame.

```
CFDictionaryRef CTFrameGetFrameAttributes( CTFrameRef frame );
```

Parameters

frame

The frame whose attributes are returned.

Return Value

A reference to a CFDictionary object containing the frame attributes that were used to create the frame, or, if the frame was created without any frame attributes, NULL.

Discussion

You can create a frame with an attributes dictionary to control various aspects of the framing process. These attributes are different from the ones used to create an attributed string.

Availability

Available in Mac OS X v10.5 and later.

Declared In

CTFrame.h

CTFrameGetLineOrigins

Copies a range of line origins for a frame.

```
void CTFrameGetLineOrigins( CTFrameRef frame, CFRange range, CGPoint origins[] );
```

Parameters

frame

The frame whose line origin array is copied.

range

The range of line origins you wish to copy. If the length of the range is 0, then the copy operation continues from the start index of the range to the last line origin.

origins

The buffer to which the origins are copied. The buffer must have at least as many elements as specified by range's length.

Discussion

This function copies a range of `CGPoint` structures. Each `CGPoint` is the origin of the corresponding line in the array of lines returned by [CTFrameGetLines](#) (page 7), relative to the origin of the frame's path. The maximum number of line origins returned by this function is the count of the array of lines.

Availability

Available in Mac OS X v10.5 and later.

Declared In

CTFrame.h

CTFrameGetLines

Returns an array of lines stored in the frame.

```
CFArrayRef CTFrameGetLines( CTFrameRef frame );
```

Parameters

frame

The frame whose line array is returned.

Return Value

A `CFArray` object containing the `CTLine` objects that make up the frame, or, if there are no lines in the frame, an array with no elements.

Availability

Available in Mac OS X v10.5 and later.

Declared In

CTFrame.h

CTFrameGetPath

Returns the path used to create the frame.

```
CGPathRef CTFrameGetPath( CTFrameRef frame );
```

Parameters*frame*

The frame whose path is returned.

Availability

Available in Mac OS X v10.5 and later.

Declared In

CTFrame.h

CTFrameGetStringRange

Returns the range of characters originally requested to fill the frame.

```
CFRange CTFrameGetStringRange( CTFrameRef frame );
```

Parameters*frame*

The frame whose character range is returned.

Return ValueA `CFRange` structure containing the backing store range of characters that were originally requested to fill the frame, or, if the function call is not successful, an empty range.**Availability**

Available in Mac OS X v10.5 and later.

Declared In

CTFrame.h

CTFrameGetTypeID

Returns the type identifier for the CTFrame opaque type.

```
CFTypeID CTFrameGetTypeID( void );
```

Return Value

The type identifier for the CTFrame opaque type.

Availability

Available in Mac OS X v10.5 and later.

Declared In

CTFrame.h

CTFrameGetVisibleStringRange

Returns the range of characters that actually fit in the frame.

```
CFRange CTFrameGetVisibleStringRange( CTFrameRef frame );
```

Parameters

frame

The frame whose visible character range is returned.

Return Value

A `CFRange` structure containing the backing store range of characters that fit into the frame, or if the function call is not successful or no characters fit in the frame, an empty range.

Discussion

This function can be used to cascade frames, because it returns the range of characters that can be seen in the frame. The next frame would start where this frame ends.

Availability

Available in Mac OS X v10.5 and later.

Declared In

`CTFrame.h`

Data Types

CTFrameRef

A reference to a Core text frame object.

```
typedef const struct __CTFrame *CTFrameRef;
```

Availability

Available in Mac OS X v10.5 and later.

Declared In

`CTFrame.h`

Constants

CTFrameProgression

These constants specify frame progression types.

```
enum{
    kCTFrameProgressionTopToBottom = 0,
    kCTFrameProgressionRightToLeft = 1
};
typedef uint32_t CTFrameProgression;
```

Constants

kCTFrameProgressionTopToBottom

Lines are stacked top to bottom for horizontal text.

Available in Mac OS X v10.5 and later.

Declared in CTFrame.h.

kCTFrameProgressionRightToLeft

Lines are stacked right to left for vertical text.

Available in Mac OS X v10.5 and later.

Declared in CTFrame.h.

Discussion

The lines of text within a frame may be stacked for either horizontal or vertical text. Values are enumerated for each stacking type supported by CTFrame. Frames created with a progression type specifying vertical text rotate lines 90 degrees counterclockwise when drawing.

Declared In

CTFrame.h

kCTFrameProgressionAttributeName

Specifies progression for a frame.

```
const CFStringRef kCTFrameProgressionAttributeName;
```

Constants

kCTFrameProgressionAttributeName

A CFNumberRef object containing a “CTFrameProgression” (page 9) constant. The default is kCTFrameProgressionTopToBottom.

Available in Mac OS X v10.5 and later.

Declared in CTFrame.h.

Discussion

This value determines the line-stacking behavior for a frame and does not affect the appearance of the glyphs within that frame.

Declared In

CTFrame.h

Document Revision History

This table describes the changes to *CTFrame Reference*.

Date	Notes
2007-05-01	New document that describes the Core Text opaque type used to represent a frame containing multiple lines of text.

REVISION HISTORY

Document Revision History

Index

C

CTFrameDraw **function** [6](#)
CTFrameGetFrameAttributes **function** [6](#)
CTFrameGetLineOrigins **function** [7](#)
CTFrameGetLines **function** [7](#)
CTFrameGetPath **function** [8](#)
CTFrameGetStringRange **function** [8](#)
CTFrameGetTypeID **function** [8](#)
CTFrameGetVisibleStringRange **function** [9](#)
CTFrameProgression [9](#)
CTFrameRef **structure** [9](#)

K

kCTFrameProgressionAttributeName [10](#)
kCTFrameProgressionAttributeName **constant** [10](#)
kCTFrameProgressionRightToLeft **constant** [10](#)
kCTFrameProgressionTopToBottom **constant** [10](#)