

---

# NSGlyphStorage Protocol Reference

[Cocoa](#) > [Text & Fonts](#)



2009-01-06



Apple Inc.  
© 2009 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, Cocoa, 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

## **NSGlyphStorage Protocol Reference 5**

---

Overview 5

Tasks 5

    Returning Text Storage 5

    Returning Glyph Display Options 5

    Modifying the Glyph Cache 5

Instance Methods 6

    attributedString 6

    insertGlyphs:length:forStartingGlyphAtIndex:characterIndex: 6

    layoutOptions 7

    setIntAttribute:value:forGlyphAtIndex: 7

Constants 7

    Layout Options 7

## **Document Revision History 9**

---

## **Index 11**

---



# NSGlyphStorage Protocol Reference

---

<b>Adopted by</b>	NSLayoutManager
<b>Framework</b>	/System/Library/Frameworks/AppKit.framework
<b>Availability</b>	Available in Mac OS X v10.3 and later.
<b>Declared in</b>	NSGlyphGenerator.h
<b>Companion guides</b>	Text System Overview Text Layout Programming Guide for Cocoa

## Overview

The `NSGlyphStorage` protocol defines the methods that a glyph storage object must implement in order to interact properly with `NSGlyphGenerator`.

An example of a Cocoa class conforming to the `NSGlyphStorage` protocol is `NSLayoutManager`.

## Tasks

### Returning Text Storage

- [attributedString](#) (page 6)  
Returns the text storage object from which the `NSGlyphGenerator` object procures characters for glyph generation.

### Returning Glyph Display Options

- [layoutOptions](#) (page 7)  
Returns the current layout options.

### Modifying the Glyph Cache

- [insertGlyphs:length:forStartingGlyphAtIndex:characterIndex:](#) (page 6)  
Inserts the given glyphs into the glyph cache and maps them to the specified characters.

- [setIntAttribute:value:forGlyphAtIndex:](#) (page 7)  
Sets a custom attribute value for a given glyph.

## Instance Methods

### attributedString

Returns the text storage object from which the `NSGlyphGenerator` object procures characters for glyph generation.

```
- (NSAttributedString *)attributedString
```

#### Return Value

The receiver's text storage object.

#### Availability

Available in Mac OS X v10.3 and later.

#### Declared In

`NSGlyphGenerator.h`

### insertGlyphs:length:forStartingGlyphAtIndex:characterIndex:

Inserts the given glyphs into the glyph cache and maps them to the specified characters.

```
- (void)insertGlyphs:(const NSGlyph *)glyphs length:(NSUInteger)length
  forStartingGlyphAtIndex:(NSUInteger)glyphIndex
  characterIndex:(NSUInteger)charIndex
```

#### Parameters

*glyphs*

The glyphs to insert.

*glyphIndex*

Location in the glyph cache to begin inserting glyphs.

*length*

Number of glyphs to insert.

*charIndex*

Index of first character to be mapped.

#### Discussion

This is a bulk insert method for the glyph cache.

#### Availability

Available in Mac OS X v10.3 and later.

#### Declared In

`NSGlyphGenerator.h`

## layoutOptions

Returns the current layout options.

- (NSUInteger)layoutOptions

### Return Value

The layout options as a bit mask, as defined in “Constants” (page 7).

### Availability

Available in Mac OS X v10.3 and later.

### Declared In

NSGlyphGenerator.h

## setIntAttribute:value:forGlyphAtIndex:

Sets a custom attribute value for a given glyph.

```
- (void)setIntAttribute:(NSInteger)attributeTag value:(NSInteger)val
forGlyphAtIndex:(NSUInteger)glyphIndex
```

### Parameters

*attributeTag*

The custom attribute.

*val*

The new attribute value.

*glyphIndex*

Index of the glyph whose attribute is set.

### Discussion

Custom attributes are glyph attributes such as `NSGlyphInscription` or attributes defined by subclasses. Subclasses that define their own custom attributes must override this method and provide their own storage for the attribute values. Nonnegative tags are reserved; you can define your own attributes with negative tags and set values using this method.

### Availability

Available in Mac OS X v10.3 and later.

### Declared In

NSGlyphGenerator.h

## Constants

### Layout Options

These constants describe layout options returned as a bit mask by the `layoutOptions` (page 7) method.

```
enum {  
    NSShowControlGlyphs = (1 << 0),  
    NSShowInvisibleGlyphs = (1 << 1),  
    NSWantsBidiLevels = (1 << 2)  
};
```

### Constants

`NSShowControlGlyphs`

Generates displayable glyphs for control characters.

Available in Mac OS X v10.3 and later.

Declared in `NSGlyphGenerator.h`.

`NSShowInvisibleGlyphs`

Generates displayable glyphs for invisible characters.

Available in Mac OS X v10.3 and later.

Declared in `NSGlyphGenerator.h`.

`NSWantsBidiLevels`

Generates directional formatting codes for bidirectional text.

Available in Mac OS X v10.3 and later.

Declared in `NSGlyphGenerator.h`.

### Declared In

`NSGlyphGenerator.h`



# Document Revision History

---

This table describes the changes to *NSGlyphStorage Protocol Reference*.

Date	Notes
2009-01-06	Reformatted parameter descriptions.
2006-05-23	First publication of this content as a separate document.

## REVISION HISTORY

### Document Revision History

# Index

---

## A

---

attributedString protocol instance method [6](#)

## I

---

insertGlyphs:length:forStartingGlyphAtIndex:  
characterIndex: protocol instance method [6](#)

## L

---

Layout Options [7](#)

layoutOptions protocol instance method [7](#)

## N

---

NSShowControlGlyphs constant [8](#)

NSShowInvisibleGlyphs constant [8](#)

NSWantsBidiLevels constant [8](#)

## S

---

setIntAttribute:value:forGlyphAtIndex: protocol  
instance method [7](#)