

---

# NSFontManager Class Reference

Data Management: Strings, Text, & Fonts



2007-04-06



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, Cocoa, Mac, and Mac OS are trademarks of Apple Inc., registered in the United States and other countries.

Helvetica and Times are registered trademarks of Heidelberger Druckmaschinen AG, available from Linotype Library GmbH.

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

## **NSFontManager Class Reference 5**

---

Overview	5
Tasks	6
Getting the Shared Font Manager	6
Changing the Default Font Conversion Classes	6
Getting Available Fonts	6
Setting and Examining the Selected Font	6
Sending Action Methods	7
Converting Fonts Automatically	7
Converting Fonts Manually	7
Getting a Particular Font	8
Examining Fonts	8
Managing the Font Panel and Font Menu	8
Setting the Delegate	8
Accessing the Action Method	9
Setting Attributes	9
Working with Font Descriptors	9
Class Methods	10
setFontManagerFactory:	10
setFontPanelFactory:	10
sharedFontManager	11
Instance Methods	11
action	11
addCollection:options:	12
addFontDescriptors:toCollection:	12
addFontTrait:	12
availableFontFamilies	13
availableFontNamesMatchingFontDescriptor:	13
availableFontNamesWithTraits:	14
availableFonts	14
availableMembersOfFontFamily:	15
collectionNames	15
convertAttributes:	16
convertFont:	16
convertFont:toFace:	17
convertFont:toFamily:	18
convertFont:toHaveTrait:	18
convertFont:toNotHaveTrait:	19
convertFont:toSize:	20
convertFontTraits:	20
convertWeight:ofFont:	21

- currentFontAction 22
- delegate 22
- fontDescriptorsInCollection: 23
- fontMenu: 23
- fontNamed:hasTraits: 24
- fontPanel: 24
- fontWithFamily:traits:weight:size: 25
- isEnabled 25
- isMultiple 26
- localizedNameForFamily:face: 26
- modifyFont: 26
- modifyFontViaPanel: 27
- orderFrontFontPanel: 28
- orderFrontStylesPanel: 28
- removeCollection: 29
- removeFontDescriptor:fromCollection: 29
- removeFontTrait: 29
- selectedFont 30
- sendAction 30
- setAction: 31
- setDelegate: 31
- setEnabled: 32
- setFontMenu: 32
- setSelectedAttributes:isMultiple: 32
- setSelectedFont:isMultiple: 33
- setTarget: 34
- target 34
- traitsOfFont: 34
- weightOfFont: 35
- Delegate Methods 35
  - changeFont: 35
  - fontManager:willIncludeFont: 36
- Constants 37
  - Font Collection Mask 37
  - NSFontTraitMask 37
  - Font traits 37
  - NSFontAction 39

---

**Document Revision History 43**

---

**Index 45**

---

# NSFontManager Class Reference

---

<b>Inherits from</b>	NSObject
<b>Conforms to</b>	NSObject (NSObject)
<b>Framework</b>	/System/Library/Frameworks/AppKit.framework
<b>Availability</b>	Available in Mac OS X v10.0 and later.
<b>Companion guide</b>	Font Panel
<b>Declared in</b>	NSFontManager.h
<b>Related sample code</b>	CoreTextArcCocoa NewsReader QTKitTimeCode Quartz Composer WWDC 2005 TextEdit ToolbarSample

## Overview

`NSFontManager` is the center of activity for the font conversion system. It records the currently selected font, updates the Font panel and Font menu to reflect the selected font, initiates font changes, and converts fonts in response to requests from text-bearing objects. In a more prosaic role, `NSFontManager` can be queried for the fonts available to the application and for the particular attributes of a font, such as whether it's condensed or extended.

You normally set up a font manager and the Font menu using Interface Builder. However, you can also do so programmatically by getting the shared font manager instance and having it create the standard Font menu at runtime:

```
NSFontManager *fontManager = [NSFontManager sharedFontManager];  
NSMenu *fontMenu = [fontManager fontMenu:YES];
```

You can then add the Font menu to your application's main menu. Once the Font menu is installed, your application automatically gains the functionality of both the Font menu and the Font panel.

As of Mac OS X version 10.3, font collections are managed by `NSFontManager`.

## Tasks

### Getting the Shared Font Manager

- + [sharedFontManager](#) (page 11)  
Returns the shared instance of the font manager for the application, creating it if necessary.

### Changing the Default Font Conversion Classes

- + [setFontManagerFactory:](#) (page 10)  
Sets the class object used to create the font manager to the given class.
- + [setFontPanelFactory:](#) (page 10)  
Sets the class used to create the Font panel to the given class.

### Getting Available Fonts

- [availableFonts](#) (page 14)  
Returns the names of the fonts available in the system (not the `NSFont` objects themselves).
- [availableFontFamilies](#) (page 13)  
Returns the names of the font families available in the system.
- [availableFontNamesWithTraits:](#) (page 14)  
Returns the names of the fonts available in the system whose traits are described exactly by the given font trait mask (not the `NSFont` objects themselves).
- [availableMembersOfFontFamily:](#) (page 15)  
Returns an array with one entry for each available member of a font family.

### Setting and Examining the Selected Font

- [setSelectedFont:isMultiple:](#) (page 33)  
Records the given font as the currently selected font and updates the Font panel to reflect this.
- [selectedFont](#) (page 30)  
Returns the last font recorded.
- [isMultiple](#) (page 26)  
Indicates whether the last font selection recorded has multiple fonts.
- [sendAction](#) (page 30)  
Sends the receiver's action message up the responder chain, initiating a font change for whatever conversion and trait to change were last requested.
- [localizedNameForFamily:face:](#) (page 26)  
Returns a localized string with the name of the specified font family and face, if one exists.

## Sending Action Methods

- [addFontTrait:](#) (page 12)  
This action method causes the receiver to send its action message up the responder chain.
- [removeFontTrait:](#) (page 29)  
This action method causes the receiver to send its action message up the responder chain.
- [modifyFont:](#) (page 26)  
This action method causes the receiver to send its action message up the responder chain.
- [modifyFontViaPanel:](#) (page 27)  
This action method causes the receiver to send its action message up the responder chain.
- [orderFrontStylesPanel:](#) (page 28)  
This action method opens the Font styles panel.
- [orderFrontFontPanel:](#) (page 28)  
This action method opens the Font panel by sending it an `orderFront:` message, creating the Font panel if necessary.

## Converting Fonts Automatically

- [convertFont:](#) (page 16)  
Converts the given font according to the object that initiated a font change, typically the Font panel or Font menu.
- [changeFont:](#) (page 35) *delegate method*  
Informs responders of a font change.

## Converting Fonts Manually

- [convertFont:toFace:](#) (page 17)  
Returns a font whose traits are as similar as possible to those of the given font except for the typeface, which is changed to the given typeface.
- [convertFont:toFamily:](#) (page 18)  
Returns a font whose traits are as similar as possible to those of the given font except for the font family, which is changed to the given family.
- [convertFont:toHaveTrait:](#) (page 18)  
Returns a font whose traits are the same as those of the given font, except that the traits are changed to include the single specified trait.
- [convertFont:toNotHaveTrait:](#) (page 19)  
Returns an `NSFont` object with the same traits as the given font, except for the traits in the given font trait mask, which are removed.
- [convertFont:toSize:](#) (page 20)  
Returns an `NSFont` object whose traits are the same as those of the given font, except for the size, which is changed to the given size.
- [convertWeight:ofFont:](#) (page 21)  
Returns an `NSFont` object whose weight is greater or lesser than that of the given font, if possible.
- [currentFontAction](#) (page 22)  
Returns the current font conversion action.

- `convertFontTraits:` (page 20)  
Converts font traits to a new traits mask value.

## Getting a Particular Font

- `fontWithFamily:traits:weight:size:` (page 25)  
Attempts to load a font with the specified characteristics.

## Examining Fonts

- `traitsOfFont:` (page 34)  
Returns the traits of the given font.
- `fontNamed:hasTraits:` (page 24)  
Indicates whether the given font has all the specified traits.
- `weightOfFont:` (page 35)  
Returns a rough numeric measure the weight of the given font.

## Managing the Font Panel and Font Menu

- `setEnabled:` (page 32)  
Controls whether the font conversion system's user interface items (the Font panel and Font menu items) are enabled.
- `isEnabled` (page 25)  
Indicates whether the font conversion system's user interface items (the Font panel and Font menu items) are enabled.
- `fontManager:willIncludeFont:` (page 36) *delegate method*  
Requests permission from the Font panel delegate to display the given font name in the Font panel.
- `fontPanel:` (page 24)  
Returns the application's shared Font panel object, optionally creating it if necessary.
- `setFontMenu:` (page 32)  
Records the given menu as the application's Font menu.
- `fontMenu:` (page 23)  
Returns the menu that's hooked up to the font conversion system, optionally creating it if necessary.

## Setting the Delegate

- `setDelegate:` (page 31)  
Sets the receiver's delegate to the given object.
- `delegate` (page 22)  
Returns the receiver's delegate.

## Accessing the Action Method

- [setAction:](#) (page 31)  
Sets the action that's sent to the first responder, when the user selects a new font from the Font panel or chooses a command from the Font menu, to the given selector.
- [action](#) (page 11)  
Returns the action sent to the first responder when the user selects a new font from the Font panel or chooses a command from the Font menu.
- [setTarget:](#) (page 34)  
Sets the target for the [sendAction](#) (page 30) method.
- [target](#) (page 34)  
Returns the target for the [sendAction](#) (page 30) method.

## Setting Attributes

- [setSelectedAttributes:isMultiple:](#) (page 32)  
Informs the paragraph and character formatting panels when text in a selection has changed attributes.
- [convertAttributes:](#) (page 16)  
Converts attributes in response to an object initiating an attribute change, typically the Font panel or Font menu.

## Working with Font Descriptors

- [availableFontNamesMatchingFontDescriptor:](#) (page 13)  
Returns the names of the fonts that match the attributes in the given font descriptor.
- [collectionNames](#) (page 15)  
Returns the names of the currently loaded font collections.
- [fontDescriptorsInCollection:](#) (page 23)  
Returns an array of the font descriptors in the collection specified by the given collection name.
- [addCollection:options:](#) (page 12)  
Adds a specified font collection to the font manager with a given set of options.
- [removeCollection:](#) (page 29)  
Removes the specified font collection.
- [addFontDescriptors:toCollection:](#) (page 12)  
Adds an array of font descriptors to the specified font collection.
- [removeFontDescriptor:fromCollection:](#) (page 29)  
Removes the specified font descriptor from the specified collection.

## Class Methods

### setFontManagerFactory:

Sets the class object used to create the font manager to the given class.

```
+ (void)setFontManagerFactory:(Class)aClass
```

#### Parameters

*aClass*

The new font manager factory class, which should be a subclass of `NSFontManager`.

#### Discussion

When the `NSFontManager` class object receives a [sharedFontManager](#) (page 11) message, it creates an instance of *aClass*, if no instance already exists. Your font manager class should implement `init` as its designated initializer. The default font manager factory is `NSFontManager`.

This method must be invoked before your application's main nib file is loaded, such as in the application delegate's `applicationWillFinishLaunching:` method.

#### Availability

Available in Mac OS X v10.0 and later.

#### See Also

+ [setFontPanelFactory:](#) (page 10)

#### Declared In

`NSFontManager.h`

### setFontPanelFactory:

Sets the class used to create the Font panel to the given class.

```
+ (void)setFontPanelFactory:(Class)factoryId
```

#### Parameters

*factoryId*

The new font panel factory class, which should be a subclass of `NSFontPanel`.

#### Discussion

Invoke this method before accessing the Font panel in any way, such as in the application delegate's `applicationWillFinishLaunching:` method.

#### Availability

Available in Mac OS X v10.0 and later.

#### See Also

+ [setFontManagerFactory:](#) (page 10)

#### Declared In

`NSFontManager.h`

## sharedFontManager

Returns the shared instance of the font manager for the application, creating it if necessary.

+ (NSFontManager \*)sharedFontManager

### Return Value

The shared font manager.

### Availability

Available in Mac OS X v10.0 and later.

### See Also

+ [setFontManagerFactory:](#) (page 10)

### Related Sample Code

CoreTextArcCocoa

NewsReader

QTKitTimeCode

Quartz Composer WWDC 2005 TextEdit

ToolbarSample

### Declared In

NSFontManager.h

## Instance Methods

### action

Returns the action sent to the first responder when the user selects a new font from the Font panel or chooses a command from the Font menu.

- (SEL)action

### Return Value

The selector for the action.

### Discussion

The default action is [changeFont:](#) (page 35).

### Availability

Available in Mac OS X v10.0 and later.

### See Also

- [setAction:](#) (page 31)

### Declared In

NSFontManager.h

## addCollection:options:

Adds a specified font collection to the font manager with a given set of options.

```
- (BOOL)addCollection:(NSString *)collectionName options:(NSInteger)collectionOptions
```

### Parameters

*collectionName*

The collection to add.

*collectionOptions*

The option described in “[Font Collection Mask](#)” (page 37). This option is not yet implemented.

### Return Value

YES if the font collection was successfully added; otherwise, NO.

### Availability

Available in Mac OS X v10.3 and later.

### See Also

- [removeCollection:](#) (page 29)

### Declared In

NSFontManager.h

## addFontDescriptors:toCollection:

Adds an array of font descriptors to the specified font collection.

```
- (void)addFontDescriptors:(NSArray *)descriptors toCollection:(NSString *)collectionName
```

### Parameters

*descriptors*

The font descriptors to add.

*collectionName*

The font collection to which descriptors are added.

### Availability

Available in Mac OS X v10.3 and later.

### See Also

- [removeFontDescriptor:fromCollection:](#) (page 29)

### Declared In

NSFontManager.h

## addFontTrait:

This action method causes the receiver to send its action message up the responder chain.

```
- (void)addFontTrait:(id)sender
```

**Parameters***sender*

The control that sent the message.

**Discussion**

By default, the action message is [changeFont:](#) (page 35).

When a responder replies by providing a font to convert in a [convertFont:](#) (page 16) message, the receiver converts the font by adding the trait specified by *sender*. This trait is determined by sending a [tag](#) message to *sender* and interpreting it as a font trait mask for a [convertFont:toHaveTrait:](#) (page 18) message.

**Availability**

Available in Mac OS X v10.0 and later.

**See Also**

- [removeFontTrait:](#) (page 29)
- [modifyFont:](#) (page 26)
- [modifyFontViaPanel:](#) (page 27)

**Declared In**

NSFontManager.h

**availableFontFamilies**

Returns the names of the font families available in the system.

- (NSArray \*)availableFontFamilies

**Return Value**

The names of the available font families.

**Discussion**

These fonts are in various system font directories.

**Availability**

Available in Mac OS X v10.0 and later.

**See Also**

- [availableFontNamesWithTraits:](#) (page 14)
- [availableFonts](#) (page 14)

**Declared In**

NSFontManager.h

**availableFontNamesMatchingFontDescriptor:**

Returns the names of the fonts that match the attributes in the given font descriptor.

- (NSArray \*)availableFontNamesMatchingFontDescriptor:(NSFontDescriptor \*)descriptor

**Parameters***descriptor*

The font descriptor whose attributes are matched.

**Return Value**

The names of the matching fonts.

**Availability**

Available in Mac OS X v10.3 and later.

**Declared In**

NSFontManager.h

**availableFontNamesWithTraits:**

Returns the names of the fonts available in the system whose traits are described exactly by the given font trait mask (not the `NSFont` objects themselves).

```
- (NSArray *)availableFontNamesWithTraits:(NSFontTraitMask)fontTraitMask
```

**Parameters**

*fontTraitMask*

The font traits for which to return font names. You specify the desired traits by combining the font trait mask values described in “[Constants](#)” (page 37) using the C bitwise OR operator.

**Return Value**

The names of the corresponding fonts.

**Discussion**

These fonts are in various system font directories.

If *fontTraitMask* is 0, this method returns all fonts that are neither italic nor bold. This result is the same one you'd get if *fontTraitMask* were `NSUnitalicFontMask | NSUnboldFontMask`.

**Availability**

Available in Mac OS X v10.0 and later.

**See Also**

- [availableFontFamilies](#) (page 13)
- [availableFonts](#) (page 14)

**Declared In**

NSFontManager.h

**availableFonts**

Returns the names of the fonts available in the system (not the `NSFont` objects themselves).

```
- (NSArray *)availableFonts
```

**Return Value**

The names of the available fonts.

**Discussion**

These fonts are in various system font directories.

**Availability**

Available in Mac OS X v10.0 and later.

**See Also**

- [availableFontFamilies](#) (page 13)
- [availableFontNamesWithTraits:](#) (page 14)

**Related Sample Code**

NSFontAttributeExplorer

**Declared In**

NSFontManager.h

**availableMembersOfFontFamily:**

Returns an array with one entry for each available member of a font family.

```
- (NSArray *)availableMembersOfFontFamily:(NSString *)family
```

**Parameters***family*The name of a font family, like one that [availableFontFamilies](#) (page 13) returns.**Return Value**The available members of *family*. See the following discussion for a specific description.**Discussion**

Each entry of the returned NSArray is another NSArray with four members, as follows:

0. The PostScript font name, as an NSString object.
1. The part of the font name used in the font panel that's not the font name, as an NSString object. This value is not localized—for example, "Roman", "Italic", or "Bold".
2. The font's weight, as an NSNumber.
3. The font's traits, as an NSNumber.

The members of the family are arranged in the font panel order (narrowest to widest, lightest to boldest, plain to italic).

For example, if you call `availableMembersOfFontFamily:@"Times"`, it might return an array like this:

```
(("Times-Roman", "Roman", 5, 4),
 ("Times-Italic", "Italic", 6, 5),
 ("Times-Bold", "Bold", 9, 2),
 ("Times-BoldItalic", "Bold Italic", 9, 3)
)
```

**Availability**

Available in Mac OS X v10.0 and later.

**Declared In**

NSFontManager.h

**collectionNames**

Returns the names of the currently loaded font collections.

- (NSArray \*)collectionNames

**Return Value**

The names of the current font collections.

**Availability**

Available in Mac OS X v10.3 and later.

**See Also**

- [fontDescriptorsInCollection:](#) (page 23)

**Declared In**

NSFontManager.h

**convertAttributes:**

Converts attributes in response to an object initiating an attribute change, typically the Font panel or Font menu.

- (NSDictionary \*)convertAttributes:(NSDictionary \*)*attributes*

**Parameters**

*attributes*

The current attributes.

**Return Value**

The converted attributes, or *attributes* itself if the conversion isn't possible.

**Discussion**

Attributes unused by the sender should not be changed or removed.

This method is usually invoked on the sender of `changeAttributes:`. See `NSTextView` for more information.

**Availability**

Available in Mac OS X v10.3 and later.

**See Also**

- [setSelectedAttributes:isMultiple:](#) (page 32)

**Declared In**

NSFontManager.h

**convertFont:**

Converts the given font according to the object that initiated a font change, typically the Font panel or Font menu.

- (NSFont \*)convertFont:(NSFont \*)*aFont*

**Parameters**

*aFont*

The font to convert.

**Return Value**

The converted font, or *aFont* itself if the conversion isn't possible.

**Discussion**

This method is invoked in response to an action message such as `addFontTrait:` (page 12) or `modifyFontViaPanel:` (page 27). These initiating methods cause the font manager to query the sender for the action to take and the traits to change. See “[Converting Fonts Manually](#)” (page 7) for more information.

**Availability**

Available in Mac OS X v10.0 and later.

**See Also**

- `convertFont:toFace:` (page 17)
- `convertFont:toFamily:` (page 18)
- `convertFont:toHaveTrait:` (page 18)
- `convertFont:toNotHaveTrait:` (page 19)
- `convertFont:toSize:` (page 20)
- `convertWeight:ofFont:` (page 21)

**Declared In**

NSFontManager.h

**convertFont:toFace:**

Returns a font whose traits are as similar as possible to those of the given font except for the typeface, which is changed to the given typeface.

```
-(NSFont *)convertFont:(NSFont *)aFont toFace:(NSString *)typeface
```

**Parameters**

*aFont*

The font whose traits are matched.

*typeface*

The new typeface; a fully specified family-face name, such as Helvetica-BoldOblique or Times-Roman.

**Return Value**

A font with matching traits and the given typeface, or *aFont* if it can't be converted.

**Discussion**

This method attempts to match the weight and posture of *aFont* as closely as possible. Italic is mapped to Oblique, for example. Weights are mapped based on an approximate numeric scale of 0 to 15.

**Availability**

Available in Mac OS X v10.0 and later.

**See Also**

- `convertFont:toFamily:` (page 18)
- `convertFont:toHaveTrait:` (page 18)
- `convertFont:toNotHaveTrait:` (page 19)
- `convertFont:toSize:` (page 20)
- `convertWeight:ofFont:` (page 21)
- `convertFont:` (page 16)

**Declared In**

NSFontManager.h

**convertFont:toFamily:**

Returns a font whose traits are as similar as possible to those of the given font except for the font family, which is changed to the given family.

```
- (NSFont *)convertFont:(NSFont *)aFont toFamily:(NSString *)family
```

**Parameters***aFont*

The font whose traits are matched.

*family*

The new font family; a generic font name, such as Helvetica or Times.

**Return Value**

A font with matching traits and the given family, or *aFont* if it can't be converted.

**Discussion**

This method attempts to match the weight and posture of *aFont* as closely as possible. Italic is mapped to Oblique, for example. Weights are mapped based on an approximate numeric scale of 0 to 15.

**Availability**

Available in Mac OS X v10.0 and later.

**See Also**

- [convertFont:toFace:](#) (page 17)
- [convertFont:toHaveTrait:](#) (page 18)
- [convertFont:toNotHaveTrait:](#) (page 19)
- [convertFont:toSize:](#) (page 20)
- [convertWeight:ofFont:](#) (page 21)
- [convertFont:](#) (page 16)

**Declared In**

NSFontManager.h

**convertFont:toHaveTrait:**

Returns a font whose traits are the same as those of the given font, except that the traits are changed to include the single specified trait.

```
- (NSFont *)convertFont:(NSFont *)aFont toHaveTrait:(NSFontTraitMask)fontTrait
```

**Parameters***aFont*

The font whose traits are matched.

*fontTrait*

The new trait; may be any one of the traits described in [“Constants”](#) (page 37). Using `NSUnboldFontMask` or `NSUnitalicFontMask` removes the bold or italic trait, respectively.

**Return Value**

A font with matching traits including the given trait, or *aFont* if it can't be converted.

**Discussion**

Using `NSUnboldFontMask` or `NSUnitalicFontMask` removes the bold or italic trait, respectively.

**Availability**

Available in Mac OS X v10.0 and later.

**See Also**

- [convertFont:toNotHaveTrait:](#) (page 19)
- [convertFont:toFace:](#) (page 17)
- [convertFont:toFamily:](#) (page 18)
- [convertFont:toSize:](#) (page 20)
- [convertWeight:ofFont:](#) (page 21)
- [convertFont:](#) (page 16)

**Related Sample Code**

CoreTextArcCocoa

NewsReader

ToolbarSample

**Declared In**

NSFontManager.h

**convertFont:toNotHaveTrait:**

Returns an `NSFont` object with the same traits as the given font, except for the traits in the given font trait mask, which are removed.

```
- (NSFont *)convertFont:(NSFont *)aFont toNotHaveTrait:(NSFontTraitMask)fontTraitMask
```

**Parameters**

*aFont*

The font whose traits are matched.

*fontTraitMask*

The mask for the traits to remove, created using the C bitwise OR operator to combine the traits described in “[Constants](#)” (page 37). Using `NSUnboldFontMask` or `NSUnitalicFontMask` removes the bold or italic trait, respectively.

**Return Value**

A font with matching traits minus the given traits, or *aFont* if it can't be converted.

**Availability**

Available in Mac OS X v10.0 and later.

**See Also**

- [convertFont:toHaveTrait:](#) (page 18)
- [convertFont:toFace:](#) (page 17)
- [convertFont:toFamily:](#) (page 18)
- [convertFont:toSize:](#) (page 20)
- [convertWeight:ofFont:](#) (page 21)

- [convertFont:](#) (page 16)

#### Related Sample Code

CoreTextArcCocoa

ToolbarSample

#### Declared In

NSFontManager.h

### convertFont:toSize:

Returns an `NSFont` object whose traits are the same as those of the given font, except for the size, which is changed to the given size.

```
- (NSFont *)convertFont:(NSFont *)aFont toSize:(CGFloat)size
```

#### Parameters

*aFont*

The font whose traits are matched.

*size*

The new font size.

#### Return Value

A font with matching traits except in the new size, or *aFont* if it can't be converted.

#### Availability

Available in Mac OS X v10.0 and later.

#### See Also

- [convertFont:toFace:](#) (page 17)
- [convertFont:toFamily:](#) (page 18)
- [convertFont:toHaveTrait:](#) (page 18)
- [convertFont:toNotHaveTrait:](#) (page 19)
- [convertWeight:ofFont:](#) (page 21)
- [convertFont:](#) (page 16)

#### Related Sample Code

ToolbarSample

#### Declared In

NSFontManager.h

### convertFontTraits:

Converts font traits to a new traits mask value.

```
- (NSFontTraitMask)convertFontTraits:(NSFontTraitMask)traits
```

#### Parameters

*traits*

The current font traits.

**Return Value**

The new traits mask value to be used by `convertFont:` (page 16).

**Discussion**

This method is intended to be invoked to query the font traits while the action message (usually `changeFont:` (page 35)) is being invoked when the current font action is either `NSAddTraitFontAction` (page 40) or `NSRemoveTraitFontAction` (page 40).

**Availability**

Available in Mac OS X v10.5 and later.

**Declared In**

`NSFontManager.h`

**convertWeight:ofFont:**

Returns an `NSFont` object whose weight is greater or lesser than that of the given font, if possible.

```
- (NSFont *)convertWeight:(BOOL)increaseFlag ofFont:(NSFont *)aFont
```

**Parameters**

*increaseFlag*

If YES, a heavier font is returned; if it's NO, a lighter font is returned.

*aFont*

The font whose weight is increased or decreased.

**Return Value**

A font with matching traits except for the new weight, or *aFont* if it can't be converted.

**Discussion**

Weights are graded along the following scale. The list on the left gives Apple's terminology, and the list on the right gives the ISO equivalents. Names on the same line are treated as identical:

Apple Terminology	ISO Equivalent
1. ultralight	
2. thin	W1. ultralight
3. light, extralight	W2. extralight
4. book	W3. light
5. regular, plain, display, roman	W4. semilight
6. medium	W5. medium
7. demi, demibold	
8. semi, semibold	W6. semibold
9. bold	W7. bold
10. extra, extrabold	W8. extrabold

Apple Terminology	ISO Equivalent
11. heavy, heavyface	
12. black, super	W9. ultrabold
13. ultra, ultrablack, fat	
14. extrablack, obese, nord	

The `NSFontManager` implementation of this method refuses to convert a font's weight if it can't maintain all other traits, such as italic and condensed. You might wish to override this method to allow a looser interpretation of weight conversion.

**Availability**

Available in Mac OS X v10.0 and later.

**See Also**

- [convertFont:toFace:](#) (page 17)
- [convertFont:toFamily:](#) (page 18)
- [convertFont:toHaveTrait:](#) (page 18)
- [convertFont:toNotHaveTrait:](#) (page 19)
- [convertFont:toSize:](#) (page 20)
- [convertFont:](#) (page 16)

**Declared In**

`NSFontManager.h`

**currentFontAction**

Returns the current font conversion action.

- (NSFontAction)currentFontAction

**Return Value**

The current font action used by the [convertFont:](#) (page 16) method.

**Discussion**

This method is intended to be invoked to query the font conversion action while the action message (usually [changeFont:](#) (page 35)) is being invoked.

**Availability**

Available in Mac OS X v10.5 and later.

**Declared In**

`NSFontManager.h`

**delegate**

Returns the receiver's delegate.

- (id)delegate

**Return Value**

The receiver's delegate.

**Availability**

Available in Mac OS X v10.0 and later.

**See Also**

- [setDelegate:](#) (page 31)

**Declared In**

NSFontManager.h

## fontDescriptorsInCollection:

Returns an array of the font descriptors in the collection specified by the given collection name.

- (NSArray \*)fontDescriptorsInCollection:(NSString \*)*collectionName*

**Parameters**

*collectionName*

The font collection for which to return descriptors.

**Return Value**

The font descriptors.

**Availability**

Available in Mac OS X v10.3 and later.

**See Also**

- [collectionNames](#) (page 15)

**Declared In**

NSFontManager.h

## fontMenu:

Returns the menu that's hooked up to the font conversion system, optionally creating it if necessary.

- (NSMenu \*)fontMenu:(BOOL)*createFlag*

**Parameters**

*createFlag*

If YES, the menu object is created if necessary; if NO, it is not.

**Return Value**

The font conversion system menu.

**Availability**

Available in Mac OS X v10.0 and later.

**See Also**

- [setFontMenu:](#) (page 32)

**Declared In**

NSFontManager.h

**fontName:hasTraits:**

Indicates whether the given font has all the specified traits.

```
- (BOOL)fontName:(NSString *)typeface hasTraits:(NSFontTraitMask)fontTraitMask
```

**Parameters***typeface*

The name of the font.

*fontTraitMask*The font traits to test, specified by combining the font trait mask values described in [“Constants”](#) (page 37) using the C bitwise OR operator.**Return Value**YES if the font named *typeface* has all the traits specified in *fontTraitMask*; NO if it doesn't.**Discussion**Using `NSUnboldFontMask` returns YES if the font is not bold, NO otherwise. Using `NSUnitalicFontMask` returns YES if the font is not italic, NO otherwise.**Availability**

Available in Mac OS X v10.0 and later.

**Declared In**

NSFontManager.h

**fontPanel:**

Returns the application's shared Font panel object, optionally creating it if necessary.

```
- (NSFontPanel *)fontPanel:(BOOL)createFlag
```

**Parameters***createFlag*

If YES, the Font panel object is created if necessary; if NO, it is not.

**Return Value**

The application's shared Font panel object.

**Availability**

Available in Mac OS X v10.0 and later.

**See Also**

- + `sharedFontPanel` (NSFontPanel)
- + `sharedFontPanelExists` (NSFontPanel)
- + [setFontPanelFactory:](#) (page 10)

**Declared In**

NSFontManager.h

## fontWithFamily:traits:weight:size:

Attempts to load a font with the specified characteristics.

```
- (NSFont *)fontWithFamily:(NSString *)family traits:(NSFontTraitMask)fontTraitMask
  weight:(NSInteger)weight size:(CGFloat)size
```

### Parameters

*family*

The generic name of the desired font, such as Times or Helvetica.

*fontTraitMask*

The font traits, specified by combining the font trait mask values described in “Constants” (page 37) using the C bitwise OR operator. Using `NSUnboldFontMask` or `NSUnitalicFontMask` loads a font that doesn’t have either the bold or italic trait, respectively.

*weight*

A hint for the weight desired, on a scale of 0 to 15: a value of 5 indicates a normal or book weight, and 9 or more a bold or heavier weight. The weight is ignored if *fontTraitMask* includes `NSBoldFontMask`.

*size*

The point size of the desired font.

### Return Value

A font with the specified characteristics if successful, or `nil` if not.

### Availability

Available in Mac OS X v10.0 and later.

### Declared In

`NSFontManager.h`

## isEnabled

Indicates whether the font conversion system’s user interface items (the Font panel and Font menu items) are enabled.

```
- (BOOL)isEnabled
```

### Return Value

YES if the font conversion system’s user interface items (the Font panel and Font menu items) are enabled; NO if they’re not.

### Availability

Available in Mac OS X v10.0 and later.

### See Also

- `isEnabled` (`NSFontPanel`)
- `isEnabled` (`NSMenuItem`)
- `setEnabled:` (page 32)

### Declared In

`NSFontManager.h`

## isMultiple

Indicates whether the last font selection recorded has multiple fonts.

- (BOOL)isMultiple

### Return Value

YES if the last font selection recorded has multiple fonts; NO if it's a single font.

### Availability

Available in Mac OS X v10.0 and later.

### See Also

- [setSelectedFont:isMultiple:](#) (page 33)
- [selectedFont](#) (page 30)

### Declared In

NSFontManager.h

## localizedNameForFamily:face:

Returns a localized string with the name of the specified font family and face, if one exists.

- (NSString \*)localizedNameForFamily:(NSString \*)family face:(NSString \*)face

### Parameters

*family*

The font family, for example, @"Times".

*face*

The font face, for example, @"Roman".

### Return Value

A localized string with the name of the specified font family and face, or, if *face* is *nil*, the font family only.

### Discussion

The user's locale is determined from the user's `NSLanguages` default setting. The method also loads the localized strings for the font, if they aren't already loaded.

### Availability

Available in Mac OS X v10.0 and later.

### Declared In

NSFontManager.h

## modifyFont:

This action method causes the receiver to send its action message up the responder chain.

- (void)modifyFont:(id)sender

### Parameters

*sender*

The control that sent the message.

**Discussion**

By default, the action message is [changeFont:](#) (page 35).

When a responder replies by providing a font to convert in a [convertFont:](#) (page 16) message, the receiver converts the font in the manner specified by *sender*. The conversion is determined by sending a tag message to *sender* and invoking a corresponding method:

Sender's Tag	Method Used
NSNoFontChangeAction	None; the font is returned unchanged.
NSViaPanelFontAction	The Font panel's <code>panelConvertFont:.</code>
NSAddTraitFontAction	<a href="#">convertFont:toHaveTrait:</a> (page 18).
NSRemoveTraitFontAction	<a href="#">convertFont:toNotHaveTrait:</a> (page 19).
NSSizeUpFontAction	<a href="#">convertFont:toSize:</a> (page 20).
NSSizeDownFontAction	<a href="#">convertFont:toSize:</a> (page 20).
NSHeavierFontAction	<a href="#">convertWeight:ofFont:</a> (page 21).
NSLighterFontAction	<a href="#">convertWeight:ofFont:</a> (page 21).

**Availability**

Available in Mac OS X v10.0 and later.

**See Also**

- [addFontTrait:](#) (page 12)
- [removeFontTrait:](#) (page 29)
- [modifyFontViaPanel:](#) (page 27)

**Declared In**

NSFontManager.h

**modifyFontViaPanel:**

This action method causes the receiver to send its action message up the responder chain.

```
- (void)modifyFontViaPanel:(id)sender
```

**Parameters**

*sender*

The control that sent the message.

**Discussion**

By default, the action message is [changeFont:](#) (page 35).

When a responder replies by providing a font to convert in a [convertFont:](#) (page 16) message, the receiver converts the font by sending a `panelConvertFont:` message to the Font panel. The panel in turn may send [convertFont:toFamily:](#) (page 18), [convertFont:toHaveTrait:](#) (page 18), and other specific conversion methods to make its change.

### Availability

Available in Mac OS X v10.0 and later.

### See Also

- [addFontTrait:](#) (page 12)
- [removeFontTrait:](#) (page 29)
- [modifyFont:](#) (page 26)

### Declared In

NSFontManager.h

## orderFrontFontPanel:

This action method opens the Font panel by sending it an `orderFront:` message, creating the Font panel if necessary.

```
- (void)orderFrontFontPanel:(id)sender
```

### Parameters

*sender*

The control that sent the message.

### Availability

Available in Mac OS X v10.0 and later.

### See Also

- [fontPanel:](#) (page 24)
- + [setFontPanelFactory:](#) (page 10)

### Related Sample Code

Quartz Composer WWDC 2005 TextEdit

### Declared In

NSFontManager.h

## orderFrontStylesPanel:

This action method opens the Font styles panel.

```
- (void)orderFrontStylesPanel:(id)sender
```

### Parameters

*sender*

The control that sent the message.

### Availability

Available in Mac OS X v10.3 and later.

### Declared In

NSFontManager.h

**removeCollection:**

Removes the specified font collection.

```
- (BOOL)removeCollection:(NSString *)collectionName
```

**Parameters**

*collectionName*

The collection to remove.

**Return Value**

YES if the font collection was successfully removed; otherwise, NO.

**Availability**

Available in Mac OS X v10.3 and later.

**See Also**

- [addCollection:options:](#) (page 12)

**Declared In**

NSFontManager.h

**removeFontDescriptor:fromCollection:**

Removes the specified font descriptor from the specified collection.

```
- (void)removeFontDescriptor:(NSFontDescriptor *)descriptor fromCollection:(NSString *)collection
```

**Parameters**

*descriptor*

The font descriptor to remove.

*collection*

The font collection from which to remove the descriptor.

**Availability**

Available in Mac OS X v10.3 and later.

**See Also**

- [addFontDescriptors:toCollection:](#) (page 12)

**Declared In**

NSFontManager.h

**removeFontTrait:**

This action method causes the receiver to send its action message up the responder chain.

```
- (void)removeFontTrait:(id)sender
```

**Parameters**

*sender*

The control that sent the message.

**Discussion**

By default, the action message is [changeFont:](#) (page 35).

When a responder replies by providing a font to convert in a [convertFont:](#) (page 16) message, the receiver converts the font by removing the trait specified by *sender*. This trait is determined by sending a `tag` message to *sender* and interpreting it as a font trait mask for a [convertFont:toNotHaveTrait:](#) (page 19) message.

**Availability**

Available in Mac OS X v10.0 and later.

**See Also**

- [addFontTrait:](#) (page 12)
- [modifyFont:](#) (page 26)
- [modifyFontViaPanel:](#) (page 27)

**Declared In**

NSFontManager.h

**selectedFont**

Returns the last font recorded.

- (NSFont \*)selectedFont

**Return Value**

The last font recorded with a [setSelectedFont:isMultiple:](#) (page 33) message

**Discussion**

While fonts are being converted in response to a [convertFont:](#) (page 16) message, you can determine the font selected in the Font panel like this:

```
NSFontManager *fontManager = [NSFontManager sharedFontManager];
panelFont = [fontManager convertFont:[fontManager selectedFont]];
```

**Availability**

Available in Mac OS X v10.0 and later.

**See Also**

- [isMultiple](#) (page 26)

**Related Sample Code**

QTKitTimeCode

**Declared In**

NSFontManager.h

**sendAction**

Sends the receiver's action message up the responder chain, initiating a font change for whatever conversion and trait to change were last requested.

- (BOOL)sendAction

**Return Value**

YES if some object handled the [changeFont:](#) (page 35) message; NO if the message went unheard.

**Discussion**

By default, the receiver's action message is [changeFont:](#) (page 35).

This method is used internally by the font conversion system. You should never need to invoke it directly. Instead, use the action methods such as [addFontTrait:](#) (page 12) or [modifyFontViaPanel:](#) (page 27).

**Availability**

Available in Mac OS X v10.0 and later.

**See Also**

- [setAction:](#) (page 31)

**Declared In**

NSFontManager.h

**setAction:**

Sets the action that's sent to the first responder, when the user selects a new font from the Font panel or chooses a command from the Font menu, to the given selector.

```
- (void)setAction:(SEL)aSelector
```

**Parameters**

*aSelector*

The selector to set.

**Discussion**

The default action is [changeFont:](#) (page 35). You should rarely need to change this setting.

**Availability**

Available in Mac OS X v10.0 and later.

**See Also**

- [action](#) (page 11)

**Declared In**

NSFontManager.h

**setDelegate:**

Sets the receiver's delegate to the given object.

```
- (void)setDelegate:(id)anObject
```

**Parameters**

*anObject*

The new delegate.

**Availability**

Available in Mac OS X v10.0 and later.

**See Also**

- [delegate](#) (page 22)

**Related Sample Code**

PDF Annotation Editor

**Declared In**

NSFontManager.h

**setEnabled:**

Controls whether the font conversion system's user interface items (the Font panel and Font menu items) are enabled.

```
- (void)setEnabled:(BOOL)flag
```

**Parameters**

*flag*

If YES, they're enabled; if NO, they're disabled.

**Availability**

Available in Mac OS X v10.0 and later.

**See Also**

- [setEnabled:](#) (NSFontPanel)

- [isEnabled](#) (page 25)

**Declared In**

NSFontManager.h

**setFontMenu:**

Records the given menu as the application's Font menu.

```
- (void)setFontMenu:(NSMenu *)aMenu
```

**Parameters**

*aMenu*

The new Font menu.

**Availability**

Available in Mac OS X v10.0 and later.

**See Also**

- [fontMenu:](#) (page 23)

**Declared In**

NSFontManager.h

**setSelectedAttributes:isMultiple:**

Informs the paragraph and character formatting panels when text in a selection has changed attributes.

```
- (void)setSelectedAttributes:(NSDictionary *)attributes isMultiple:(BOOL)flag
```

**Parameters**

*attributes*

The new attributes.

*flag*

If YES, informs the panel that multiple fonts or attributes are enclosed within the selection.

**Discussion**

This method is used primarily by NSTextView.

**Availability**

Available in Mac OS X v10.3 and later.

**See Also**

- [convertAttributes:](#) (page 16)

**Declared In**

NSFontManager.h

**setSelectedFont:isMultiple:**

Records the given font as the currently selected font and updates the Font panel to reflect this.

```
- (void)setSelectedFont:(NSFont *)aFont isMultiple:(BOOL)flag
```

**Parameters**

*aFont*

The font to set as selected.

*flag*

If YES, the Font panel indicates that more than one font is contained in the selection; if NO, it does not.

**Discussion**

An object that manipulates fonts should invoke this method whenever it becomes first responder and whenever its selection changes. It shouldn't invoke this method in the process of handling a [changeFont:](#) (page 35) message, as this causes the font manager to lose the information necessary to effect the change. After all fonts have been converted, the font manager itself records the new selected font.

**Availability**

Available in Mac OS X v10.0 and later.

**See Also**

- [selectedFont](#) (page 30)

- [isMultiple](#) (page 26)

**Related Sample Code**

QTKitTimeCode

Quartz Composer WWDC 2005 TextEdit

**Declared In**

NSFontManager.h

**setTarget:**

Sets the target for the [sendAction](#) (page 30) method.

```
- (void)setTarget:(id)aTarget
```

**Parameters**

*aTarget*

The target to set.

**Availability**

Available in Mac OS X v10.5 and later.

**See Also**

- [target](#) (page 34)

**Related Sample Code**

CoreTextArcCocoa

**Declared In**

NSFontManager.h

**target**

Returns the target for the [sendAction](#) (page 30) method.

```
- (id)target
```

**Return Value**

The target for the receiver's [sendAction](#) (page 30) method.

**Availability**

Available in Mac OS X v10.5 and later.

**See Also**

- [setTarget:](#) (page 34)

**Declared In**

NSFontManager.h

**traitsOfFont:**

Returns the traits of the given font.

```
- (NSFontTraitMask)traitsOfFont:(NSFont *)aFont
```

**Parameters**

*aFont*

The font whose traits are returned.

**Return Value**

The font traits, returned as a mask created by combining values listed in ["Constants"](#) (page 37) with the C bitwise OR operator.

**Availability**

Available in Mac OS X v10.0 and later.

**Related Sample Code**

QTKitTimeCode

**Declared In**

NSFontManager.h

**weightOfFont:**

Returns a rough numeric measure the weight of the given font.

```
- (NSInteger)weightOfFont:(NSFont *)aFont
```

**Parameters**

*aFont*

The font whose weight is returned.

**Return Value**

A rough numeric measure the weight of the given font, where 0 indicates the lightest possible weight, 5 indicates a normal or book weight, and 9 or more indicates a bold or heavier weight.

**Availability**

Available in Mac OS X v10.0 and later.

**Declared In**

NSFontManager.h

## Delegate Methods

**changeFont:**

Informs responders of a font change.

```
- (void)changeFont:(id)sender
```

**Parameters**

*sender*

The object that sent the message.

**Discussion**

Generally this change is because the user changed the font either in the selection of a rich text field or in a whole plain text field. Any object that contains a font the user can change must respond to the `changeFont:` message by sending a `convertFont:` (page 16) message back to *sender* (an NSFontManager object) for each font in the selection. For more information, see “Responding to Font Changes”.

Be aware that `selectedFont` (page 30) at this point may return unpredictable results. The font returned from this method may not be the last font selected, or there may be multiple fonts selected at the time `changeFont:` is called. The use of `selectedFont` from within `changeFont:` is strongly discouraged.

**Availability**

Available in Mac OS X v10.0 and later.

**See Also**

- [addFontTrait:](#) (page 12)
- [convertFont:toHaveTrait:](#) (page 18)
- [convertFont:toFace:](#) (page 17)
- [convertFont:toFamily:](#) (page 18)
- [convertFont:toNotHaveTrait:](#) (page 19)
- [convertFont:toSize:](#) (page 20)
- [convertWeight:ofFont:](#) (page 21)
- [convertFont:](#) (page 16)
- [removeFontTrait:](#) (page 29)
- [modifyFontViaPanel:](#) (page 27)
- [modifyFont:](#) (page 26)

**Declared In**

NSFontManager.h

**fontManager:willIncludeFont:**

Requests permission from the Font panel delegate to display the given font name in the Font panel.

```
- (BOOL)fontManager:(id)theFontManager willIncludeFont:(NSString *)fontName
```

**Parameters**

*theFontManager*

The font manager making the request.

*fontName*

The full PostScript name of the font to display, such as Helvetica-BoldOblique or Helvetica-Narrow-Bold.

**Return Value**

If the Font panel delegate returns YES, *fontName* is listed; if the delegate returns NO, it isn't.

**Discussion**

In Mac OS X versions 10.2 and earlier, this method is invoked repeatedly as necessary whenever the Font panel needs updating, such as when the Font panel is first loaded, and when the user selects a family name to see which typefaces in that family are available. Your implementation should execute fairly quickly to ensure the responsiveness of the Font panel.

**Important:** This delegate method is not called in Mac OS X versions 10.3 and later.

**Availability**

Available in Mac OS X v10.0 and later.

**Declared In**

NSFontManager.h

## Constants

### Font Collection Mask

This constant specifies options accepted by `addCollection:options:` (page 12).

```
enum {
    NSFontCollectionApplicationOnlyMask = 1 << 0
};
```

#### Constants

`NSFontCollectionApplicationOnlyMask`

Makes the collection available only to the application. This option is not yet implemented.

Available in Mac OS X v10.3 and later.

Declared in `NSFontManager.h`.

#### Declared In

`NSFontManager.h`

### NSFontTraitMask

Mask of traits assigned to a font, assigned using the values in `Font traits` (page 37).

```
typedef unsigned int NSFontTraitMask;
```

#### Availability

Available in Mac OS X v10.0 and later.

#### Declared In

`NSFontManager.h`

### Font traits

Font traits defined and supported by `NSFontManager`.

```
enum {
    NSItalicFontMask = 0x00000001,
    NSBoldFontMask = 0x00000002,
    NSUnboldFontMask = 0x00000004,
    NSNonStandardCharacterSetFontMask = 0x00000008,
    NSNarrowFontMask = 0x00000010,
    NSExpandedFontMask = 0x00000020,
    NSCondensedFontMask = 0x00000040,
    NSSmallCapsFontMask = 0x00000080,
    NSPosterFontMask = 0x00000100,
    NSCompressedFontMask = 0x00000200,
    NSFittedPitchFontMask = 0x00000400,
    NSUnitalicFontMask = 0x01000000
};
```

**Constants**

NSItalicFontMask

**A mask that specifies an italic font.**

Available in Mac OS X v10.0 and later.

Declared in NSFontManager.h.

NSBoldFontMask

**A mask that specifies a bold font.**

Available in Mac OS X v10.0 and later.

Declared in NSFontManager.h.

NSUnboldFontMask

**A mask that specifies a font that is not bold.**

Available in Mac OS X v10.0 and later.

Declared in NSFontManager.h.

NSNonStandardCharacterSetFontMask

**A mask that specifies a font that uses a non-standard character set.**

Available in Mac OS X v10.0 and later.

Declared in NSFontManager.h.

NSNarrowFontMask

**A mask that specifies a narrow font.**

Available in Mac OS X v10.0 and later.

Declared in NSFontManager.h.

NSExpandedFontMask

**A mask that specifies an expanded font.**

Available in Mac OS X v10.0 and later.

Declared in NSFontManager.h.

NSCondensedFontMask

**A mask that specifies a condensed font.**

Available in Mac OS X v10.0 and later.

Declared in NSFontManager.h.

`NSSmallCapsFontMask`

A mask that specifies a small-caps font.

Available in Mac OS X v10.0 and later.

Declared in `NSFontManager.h`.

`NSPosterFontMask`

A mask that specifies a poster-style font.

Available in Mac OS X v10.0 and later.

Declared in `NSFontManager.h`.

`NSCompressedFontMask`

A mask that specifies a compressed font.

Available in Mac OS X v10.0 and later.

Declared in `NSFontManager.h`.

`NSFixedPitchFontMask`

A mask that specifies a fixed pitch font.

Available in Mac OS X v10.0 and later.

Declared in `NSFontManager.h`.

`NSUnitalicFontMask`

A mask that specifies a font that is not italic.

Available in Mac OS X v10.0 and later.

Declared in `NSFontManager.h`.

**Discussion**

`NSFontManager` categorizes fonts according to a small set of traits. You can convert fonts by adding and removing individual traits, and you can get a font with a specific combination of traits.

These pairs of traits are mutually exclusive:

`NSCondensedFontMask` and `NSExpandedFontMask`

`NSBoldFontMask` and `NSUnboldFontMask`

`NSItalicFontMask` and `NSUnitalicFontMask`

**Declared In**

`NSFontManager.h`

**NSFontAction**

These constants specify what action `modifyFont:` (page 26) will take.

```
typedef enum _NSFontAction {
    NSNoFontChangeAction = 0,
    NSViaPanelFontAction = 1,
    NSAddTraitFontAction = 2,
    NSSizeUpFontAction = 3,
    NSSizeDownFontAction = 4,
    NSHeavierFontAction = 5,
    NSLighterFontAction = 6,
    NSRemoveTraitFontAction = 7
} NSFontAction;
```

**Constants**

NSNoFontChangeAction

No action; the font is returned unchanged.

Available in Mac OS X v10.0 and later.

Declared in NSFontManager.h.

NSViaPanelFontAction

Converts the font according to the NSFontPanel method panelConvertFont:.

Available in Mac OS X v10.0 and later.

Declared in NSFontManager.h.

NSAddTraitFontAction

Converts the font to have an additional trait using [convertFont:toHaveTrait:](#) (page 18).

Available in Mac OS X v10.0 and later.

Declared in NSFontManager.h.

NSSizeUpFontAction

Converts the font to a larger size using [convertFont:toSize:](#) (page 20).

Available in Mac OS X v10.0 and later.

Declared in NSFontManager.h.

NSSizeDownFontAction

Converts the font to a smaller size using [convertFont:toSize:](#) (page 20).

Available in Mac OS X v10.0 and later.

Declared in NSFontManager.h.

NSHeavierFontAction

Converts the font to a heavier weight using [convertWeight:ofFont:](#) (page 21).

Available in Mac OS X v10.0 and later.

Declared in NSFontManager.h.

NSLighterFontAction

Converts the font to a lighter weight using [convertWeight:ofFont:](#) (page 21).

Available in Mac OS X v10.0 and later.

Declared in NSFontManager.h.

NSRemoveTraitFontAction

Converts the font to remove a trait using [convertFont:toNotHaveTrait:](#) (page 19).

Available in Mac OS X v10.0 and later.

Declared in NSFontManager.h.

**Availability**

Available in Mac OS X v10.0 and later.

**Declared In**

NSFontManager.h



# Document Revision History

---

This table describes the changes to *NSFontManager Class Reference*.

Date	Notes
2007-04-06	Documented methods added in Mac OS X v10.5.
2006-05-23	First publication of this content as a separate document.

## REVISION HISTORY

### Document Revision History

# Index

---

## A

---

action **instance method** 11  
addCollection:options: **instance method** 12  
addFontDescriptors:toCollection: **instance method** 12  
addFontTrait: **instance method** 12  
availableFontFamilies **instance method** 13  
availableFontNamesMatchingFontDescriptor: **instance method** 13  
availableFontNamesWithTraits: **instance method** 14  
availableFonts **instance method** 14  
availableMembersOfFontFamily: **instance method** 15

## C

---

changeFont: <NSObject> delegate **method** 35  
collectionNames **instance method** 15  
convertAttributes: **instance method** 16  
convertFont: **instance method** 16  
convertFont:toFace: **instance method** 17  
convertFont:toFamily: **instance method** 18  
convertFont:toHaveTrait: **instance method** 18  
convertFont:toNotHaveTrait: **instance method** 19  
convertFont:toSize: **instance method** 20  
convertFontTraits: **instance method** 20  
convertWeight:ofFont: **instance method** 21  
currentFontAction **instance method** 22

## D

---

delegate **instance method** 22

## F

---

Font Collection Mask 37  
Font traits 37  
fontDescriptorsInCollection: **instance method** 23  
fontManager:willIncludeFont: <NSObject> delegate **method** 36  
fontMenu: **instance method** 23  
fontNamed:hasTraits: **instance method** 24  
fontPanel: **instance method** 24  
fontWithFamily:traits:weight:size: **instance method** 25

## I

---

isEnabled **instance method** 25  
isMultiple **instance method** 26

## L

---

localizedNameForFamily:face: **instance method** 26

## M

---

modifyFont: **instance method** 26  
modifyFontViaPanel: **instance method** 27

## N

---

NSAddTraitFontAction **constant** 40  
NSBoldFontMask **constant** 38  
NSCompressedFontMask **constant** 39  
NSCondensedFontMask **constant** 38  
NSExpandedFontMask **constant** 38  
NSFixedPitchFontMask **constant** 39  
NSFontAction **data type** 39

NSFontCollectionApplicationOnlyMask **constant** [37](#)  
 NSFontTraitMask **data type** [37](#)  
 NSHeavierFontAction **constant** [40](#)  
 NSItalicFontMask **constant** [38](#)  
 NSLighterFontAction **constant** [40](#)  
 NSNarrowFontMask **constant** [38](#)  
 NSNoFontChangeAction **constant** [40](#)  
 NSNonStandardCharacterSetFontMask **constant** [38](#)  
 NSPosterFontMask **constant** [39](#)  
 NSRemoveTraitFontAction **constant** [40](#)  
 NSSizeDownFontAction **constant** [40](#)  
 NSSizeUpFontAction **constant** [40](#)  
 NSSmallCapsFontMask **constant** [39](#)  
 NSUnboldFontMask **constant** [38](#)  
 NSUnitalicFontMask **constant** [39](#)  
 NSViaPanelFontAction **constant** [40](#)

## O

---

orderFrontFontPanel: **instance method** [28](#)  
 orderFrontStylesPanel: **instance method** [28](#)

## R

---

removeCollection: **instance method** [29](#)  
 removeFontDescriptor:fromCollection: **instance method** [29](#)  
 removeFontTrait: **instance method** [29](#)

## S

---

selectedFont **instance method** [30](#)  
 sendAction **instance method** [30](#)  
 setAction: **instance method** [31](#)  
 setDelegate: **instance method** [31](#)  
 setEnabled: **instance method** [32](#)  
 setFontManagerFactory: **class method** [10](#)  
 setFontMenu: **instance method** [32](#)  
 setFontPanelFactory: **class method** [10](#)  
 setSelectedAttributes:isMultiple: **instance method** [32](#)  
 setSelectedFont:isMultiple: **instance method** [33](#)  
 setTarget: **instance method** [34](#)  
 sharedFontManager **class method** [11](#)

## T

---

target **instance method** [34](#)  
 traitsOfFont: **instance method** [34](#)

## W

---

weightOfFont: **instance method** [35](#)