

[AppKit](#) / Text Display

API Collection

Text Display

Display text and check spelling.

Overview

In most cases, you can lay out your app's text using the [NSTextField](#) or [NSTextView](#) classes (or their subclasses). Use the [NSTextField](#) class to add either a label or a simple text input. Use the [NSTextView](#) class to provide more comprehensive layout and editing features for larger bodies of text.

For example, [NSTextView](#) supports rich text, attachments (graphics, file, and other), input management and key binding, and marked text attributes. [NSTextView](#) works with the font panel and menu, rulers and paragraph styles, the Services facility (for example, the spell-checking service), and the pasteboard.

[NSTextView](#) also allows customizing through delegation and notifications—you rarely need to subclass [NSTextView](#). You rarely create instances of [NSTextView](#) programmatically either, because objects on Interface Builder's palettes, such as [NSTextField](#), [NSForm](#), and [NSScrollView](#), already contain [NSTextView](#) objects.

For even more powerful and more creative text manipulation (such as displaying text in a circle) see [TextKit](#).

Spell-checking

The [NSSpellServer](#) class lets you define a spell-checking service and provide it as a service to other apps. To connect your app to a spell-checking service, use the [NSSpellChecker](#) class. The [NSIgnoreMisspelledWords](#) and [NSChangeSpelling](#) protocols support the spell-checking mechanism.

Topics

Text views

`class NSTextField`

Text the user can select or edit to send an action message to a target when the user presses the Return key.

`protocol NSTextFieldDelegate`

A protocol that a text field delegate can use to control its field editor action menu.

`class NSTextView`

A view that draws text and handles user interactions with that text.

`protocol NSTextViewDelegate`

A set of optional methods that text view delegates can use to manage selection, set text attributes, work with the spell checker, and more.

`protocol NSTextDelegate`

A set of optional methods implemented by the delegate of an `NSText` object to edit text and change text formats.

`class NSText`

The most general programmatic interface for objects that manage text.

Text input

 Adopting the system text cursor in custom text views

Incorporate the system text cursor into your custom text UI in AppKit.

`class NSTextInputContext`

An object that represents the Cocoa text input system.

`protocol NSTextInputClient`

A set of methods that text views need to implement to interact properly with the text input management system.

`class NSTextAlternatives`

A list of alternative strings for a piece of text.

```
protocol NSTextContent
```

A protocol that describes specific kinds of input content types.

```
class NSTextInsertionIndicator
```

A view that represents the insertion indicator in text.

```
enum DisplayMode
```

Constants that determine how to display the system text cursor in a custom text UI.

```
struct AutomaticModeOptions
```

Options that affect the automatic display mode.

Text-checking

```
class NSTextCheckingController
```

```
protocol NSTextCheckingClient
```

```
protocol NSTextInputTraits
```

```
enum NSTextInputTraitType
```

Spell-checking

```
class NSSpellChecker
```

An interface to the Cocoa spell-checking service.

```
protocol NSChangeSpelling
```

A protocol that responder objects can implement to correct a misspelled word.

```
protocol NSIgnoreMisspelledWords
```

A protocol that enables the Ignore button in the Spelling panel to function properly.

Deprecated

```
protocol NSTextInput
```

A set of methods that text views need to implement to interact properly with the text input management system.

See Also

Text

TextKit

Manage text storage and perform custom layout of text-based content in your app's views.

Fonts

Manage the fonts used to display text.

Writing Tools

Add support for Writing Tools to your app's text views.