NSTokenFieldCell Class Reference

Cocoa > User Experience

2007-03-05

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

NSTokenFieldCell Class Reference

Inherits from	NSTextFieldCell : NSActionCell : NSCell : NSObject
Conforms to	NSCoding (NSCell) NSCopying (NSCell) NSObject (NSObject)
Framework	/System/Library/Frameworks/AppKit.framework
Availability	Available in Mac OS X v10.4 and later.
Companion guide	Token Field Programming Guide for Cocoa
Declared in	NSTokenFieldCell.h

Overview

`NSTokenFieldCell` is a subclass of `NSTextFieldCell` that provides tokenized editing of an array of objects similar to the address field in the Mail application. The objects may be strings or objects that can be represented as strings. A single token field cell can be presented in an `NSTokenField` control.

Tasks

Managing the Token Style

– `setTokenStyle:` (page 9)
Sets the token style of the receiver.

– `tokenStyle` (page 10)
Returns the receiver's token style.

Managing the Tokenizing Character Set

+ `defaultTokenizingCharacterSet` (page 7)
Returns the default tokenizing character set.

– `setTokenizingCharacterSet:` (page 9)
Sets the receiver's tokenizing character set to a given character set.

`– tokenizingCharacterSet` (page 10)
    Returns the receiver's tokenizing character set.

Configuring the Completion Delay

`– setCompletionDelay:` (page 8)
    Sets the receiver's completion delay to a given delay.

`– completionDelay` (page 8)
    Returns the receiver's completion delay.

`+ defaultCompletionDelay` (page 7)
    Returns the default completion delay.

Managing the Delegate

`– delegate` (page 8)
    Returns the receiver's delegate.

`– setDelegate:` (page 9)
    Sets the receiver's delegate.

Displaying Tokenized Strings

`– tokenFieldCell:displayStringForRepresentedObject:` (page 11) *delegate method*
    Allows the delegate to provide a string to be displayed as a proxy for the represented object.

`– tokenFieldCell:styleForRepresentedObject:` (page 14) *delegate method*
    Allows the delegate to return the token style for editing the specified represented object.

Editing Tokenized Strings

`– tokenFieldCell:editingStringForRepresentedObject:` (page 12) *delegate method*
    Allows the delegate to provide a string to be edited as a proxy for the represented object.

`– tokenFieldCell:completionsForSubstring:indexOfToken:indexOfSelectedItem:` (page 11) *delegate method*
    Allows the delegate to provide an array of appropriate completions for the contents of the receiver.

`– tokenFieldCell:representedObjectForEditingString:` (page 13) *delegate method*
    Allows the delegate to provide a represented object for the string being edited.

`– tokenFieldCell:shouldAddObjects:atIndex:` (page 13) *delegate method*
    Allows the delegate to validate the tokens to be added to the receiver at a given index.

Reading From and Writing to the Pasteboard

`– tokenFieldCell:readFromPasteboard:` (page 12) *delegate method*
    Allows the delegate to return an array of objects representing the data read from *pboard*.

`– tokenFieldCell:writeRepresentedObjects:toPasteboard:` (page 14) *delegate method*
    Allows the delegate the opportunity to write custom pasteboard types to the pasteboard for the represented objects in *`objects`*.

Managing Menus for Represented Objects

`– tokenFieldCell:hasMenuForRepresentedObject:` (page 12) *delegate method*
    Allows the delegate to specify whether the represented object provides a menu.

`– tokenFieldCell:menuForRepresentedObject:` (page 12) *delegate method*
    Allows the delegate to provide a menu for the specified represented object.

Class Methods

defaultCompletionDelay

Returns the default completion delay.

```
+ (NSTimeInterval)defaultCompletionDelay
```

Return Value
The default completion delay.

Discussion
The default completion delay is `0`.

Availability
Available in Mac OS X v10.4 and later.

Declared In
`NSTokenFieldCell.h`

defaultTokenizingCharacterSet

Returns the default tokenizing character set.

```
+ (NSCharacterSet *)defaultTokenizingCharacterSet
```

Return Value
The default tokenizing character set.

Discussion
The default tokenizing character set contains the single character "`,`".

Availability
Available in Mac OS X v10.4 and later.

Declared In
`NSTokenFieldCell.h`

Instance Methods

completionDelay

Returns the receiver's completion delay.

```
- (NSTimeInterval)completionDelay
```

Return Value
The receiver's completion delay.

Availability
Available in Mac OS X v10.4 and later.

See Also
`- setCompletionDelay:` (page 8)
`+ defaultCompletionDelay` (page 7)

Declared In
`NSTokenFieldCell.h`

delegate

Returns the receiver's delegate.

```
- (id)delegate
```

Return Value
The receiver's delegate.

Availability
Available in Mac OS X v10.4 and later.

See Also
`- setDelegate:` (page 9)

Declared In
`NSTokenFieldCell.h`

setCompletionDelay:

Sets the receiver's completion delay to a given delay.

```
- (void)setCompletionDelay:(NSTimeInterval)delay
```

Parameters

delay

The delay for the receiver.

Availability
Available in Mac OS X v10.4 and later.

See Also
`- completionDelay` (page 8)

Declared In
`NSTokenFieldCell.h`

setDelegate:

Sets the receiver's delegate.

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

Parameters

anObject
The delegate for the receiver.

Availability
Available in Mac OS X v10.4 and later.

See Also
`- delegate` (page 8)

Declared In
`NSTokenFieldCell.h`

setTokenizingCharacterSet:

Sets the receiver's tokenizing character set to a given character set.

```
- (void)setTokenizingCharacterSet:(NSCharacterSet *)characterSet
```

Parameters

characterSet
The tokenizing character set for the receiver.

Availability
Available in Mac OS X v10.4 and later.

See Also
`- tokenizingCharacterSet` (page 10)

Declared In
`NSTokenFieldCell.h`

setTokenStyle:

Sets the token style of the receiver.

```
- (void)setTokenStyle:(NSTokenStyle)style
```

Parameters

`style`

The token style for the receiver.

Availability
Available in Mac OS X v10.4 and later.

See Also
`- tokenStyle` (page 10)

Declared In
`NSTokenFieldCell.h`

tokenizingCharacterSet

Returns the receiver's tokenizing character set.

```
- (NSCharacterSet *)tokenizingCharacterSet
```

Return Value
The receiver's tokenizing character set.

Availability
Available in Mac OS X v10.4 and later.

See Also
`- setTokenizingCharacterSet:` (page 9)
`+ defaultTokenizingCharacterSet` (page 7)

Declared In
`NSTokenFieldCell.h`

tokenStyle

Returns the receiver's token style.

```
- (NSTokenStyle)tokenStyle
```

Return Value
The receiver's token style.

Availability
Available in Mac OS X v10.4 and later.

See Also
`- setTokenStyle:` (page 9)

Declared In
`NSTokenFieldCell.h`

Delegate Methods

tokenFieldCell:completionsForSubstring:indexOfToken:indexOfSelectedItem:

Allows the delegate to provide an array of appropriate completions for the contents of the receiver.

```
- (NSArray *)tokenFieldCell:(NSTokenFieldCell *)tokenFieldCell
    completionsForSubstring:(NSString *)substring indexOfToken:(NSInteger)tokenIndex
     indexOfSelectedItem:(NSInteger *)selectedIndex
```

Parameters

tokenFieldCell

The token field cell sending the message to the delegate.

substring

The partial string that to be completed.

tokenIndex

The index of the token being edited.

selectedIndex

You may return by reference an index in the returned array that specifies which of the completions should be initially selected. If you do not want an initial selection, return `NULL`.

Return Value

An array of strings for completing `substring`.

Discussion

The default behavior provides no completions.

Availability

Available in Mac OS X v10.4 and later.

Declared In

`NSTokenFieldCell.h`

tokenFieldCell:displayStringForRepresentedObject:

Allows the delegate to provide a string to be displayed as a proxy for the represented object.

```
- (NSString *)tokenFieldCell:(NSTokenFieldCell *)tokenFieldCell
    displayStringForRepresentedObject:(id)representedObject
```

Discussion

If you return `nil` or do not implement this method, then *representedObject* is displayed as the string.

Availability

Available in Mac OS X v10.4 and later.

Declared In

`NSTokenFieldCell.h`

tokenFieldCell:editingStringForRepresentedObject:

Allows the delegate to provide a string to be edited as a proxy for the represented object.

```
- (NSString *)tokenFieldCell:(NSTokenFieldCell *)tokenFieldCell
    editingStringForRepresentedObject:(id)representedObject
```

Discussion

If you return `nil` or do not implement this method, then the value returned by `tokenFieldCell:displayStringForRepresentedObject:` (page 11) is used for editing.

Availability

Available in Mac OS X v10.4 and later.

Declared In

`NSTokenFieldCell.h`

tokenFieldCell:hasMenuForRepresentedObject:

Allows the delegate to specify whether the represented object provides a menu.

```
- (BOOL)tokenFieldCell:(NSTokenFieldCell *)tokenFieldCell
    hasMenuForRepresentedObject:(id)representedObject
```

Discussion

By default tokens have no menus.

Availability

Available in Mac OS X v10.4 and later.

Declared In

`NSTokenFieldCell.h`

tokenFieldCell:menuForRepresentedObject:

Allows the delegate to provide a menu for the specified represented object.

```
- (NSMenu *)tokenFieldCell:(NSTokenFieldCell *)tokenFieldCell
    menuForRepresentedObject:(id)representedObject
```

Discussion

The returned menu should be autoreleased. By default tokens do not return menus.

Availability

Available in Mac OS X v10.4 and later.

Declared In

`NSTokenFieldCell.h`

tokenFieldCell:readFromPasteboard:

Allows the delegate to return an array of objects representing the data read from *pboard*.

```
- (NSArray *)tokenFieldCell:(NSTokenFieldCell *)tokenFieldCell
    readFromPasteboard:(NSPasteboard *)pboard
```

Availability
Available in Mac OS X v10.4 and later.

Declared In
`NSTokenFieldCell.h`

tokenFieldCell:representedObjectForEditingString:

Allows the delegate to provide a represented object for the string being edited.

```
- (id)tokenFieldCell:(NSTokenFieldCell *)tokenFieldCell
    representedObjectForEditingString:(NSString *)editingString
```

Parameters

tokenFieldCell

The token field cell sending the message to the delegate.

editingString

The string being edited.

Return Value
A represented object associated with *editingString*.

Discussion
If your application uses some object other than an `NSString` object for their represented objects, you should return a new, autoreleased instance of that object from this method.

Availability
Available in Mac OS X v10.4 and later.

Declared In
`NSTokenFieldCell.h`

tokenFieldCell:shouldAddObjects:atIndex:

Allows the delegate to validate the tokens to be added to the receiver at a given index.

```
- (NSArray *)tokenFieldCell:(NSTokenFieldCell *)tokenFieldCell
    shouldAddObjects:(NSArray *)tokens atIndex:(NSUInteger)index
```

Parameters

tokenFieldCell

The token field cell sending the message to the delegate.

tokens

An array of tokens to be added to the receiver.

index

The index in the receiver where the tokens are to be added.

Return Value
An array of validated tokens.

Discussion
The delegate can return the array unchanged or return a modified array of tokens. To reject the add completely, return an empty array. Returning `nil` causes an error.

Availability
Available in Mac OS X v10.4 and later.

Declared In
`NSTokenFieldCell.h`

tokenFieldCell:styleForRepresentedObject:

Allows the delegate to return the token style for editing the specified represented object.

```
- (NSTokenStyle)tokenFieldCell:(NSTokenFieldCell *)tokenFieldCell
    styleForRepresentedObject:(id)representedObject
```

Discussion
The delegate should return `NSDefaultTokenStyle`, `NSPlainTextTokenStyle` or `NSRoundedTokenStyle`.

Availability
Available in Mac OS X v10.4 and later.

Declared In
`NSTokenFieldCell.h`

tokenFieldCell:writeRepresentedObjects:toPasteboard:

Allows the delegate the opportunity to write custom pasteboard types to the pasteboard for the represented objects in *objects*.

```
- (BOOL)tokenFieldCell:(NSTokenFieldCell *)tokenFieldCell
    writeRepresentedObjects:(NSArray *)objects toPasteboard:(NSPasteboard *)pboard
```

Discussion
The display strings for the represented objects have already been placed on the pasteboard as both `NSStringPboardType` and an array of `NSString` objects.

Availability
Available in Mac OS X v10.4 and later.

Declared In
`NSTokenFieldCell.h`

Constants

NSTokenStyle

The `NSTokenStyle` type provides storage for values defined in "`NSTokenStyle Values`".

```
typedef NSUInteger NSTokenStyle;
```

Availability
Available in Mac OS X v10.4 and later.

Declared In
`NSTokenFieldCell.h`

NSTokenStyle Values

The NSTokenStyle constants define how tokens are displayed and editable in the `NSTokenFieldCell`. These values are used by `tokenStyle` (page 10), `setTokenStyle:` (page 9) and the delegate method `tokenFieldCell:styleForRepresentedObject:` (page 14).

```
enum {
    NSDefaultTokenStyle,
    NSPlainTextTokenStyle,
    NSRoundedTokenStyle
};
```

Constants

`NSDefaultTokenStyle`

Style best used for keyword type tokens.

Available in Mac OS X v10.4 and later.

Declared in `NSTokenFieldCell.h`.

`NSPlainTextTokenStyle`

Style to use for data you want represented as plain-text and without any token background.

Available in Mac OS X v10.4 and later.

Declared in `NSTokenFieldCell.h`.

`NSRoundedTokenStyle`

Style best used for address type tokens.

Available in Mac OS X v10.4 and later.

Declared in `NSTokenFieldCell.h`.

Declared In
`NSTokenFieldCell.h`

Document Revision History

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

Date	Notes
2007-03-05	Update with Mac OS X v10.5 API.
2007-02-08	Corrected the declaration of the NSTokenStyle enumeration.
2006-11-07	Corrected description of setTokenStyle:.
2006-05-23	First publication of this content as a separate document.

Index