NSController Class Reference

Cocoa > Data Management

2007-04-02

Apple Inc.
1 Infinite Loop
Cupertino, CA 95014
408-996-1010

Simultaneously published in the United States and Canada.

Contents

NSController Class Reference

Inherits from	NSObject
Conforms to	NSCoding NSObject (NSObject)
Framework	/System/Library/Frameworks/AppKit.framework
Availability	Available in Mac OS X v10.3 and later.
Companion guide	Cocoa Bindings Programming Topics
Declared in	NSController.h

Overview

NSController is an abstract class that implements the NSEditor and NSEditorRegistration informal protocols required for controller classes.

Adopted Protocols

NSCoding

- `encodeWithCoder:`
- `initWithCoder:`

Tasks

Managing Editing

`- objectDidBeginEditing:` (page 8)
Invoked to inform the receiver that *`editor`* has uncommitted changes that can affect the receiver.

`- objectDidEndEditing:` (page 8)
Invoked to inform the receiver that *`editor`* has committed or discarded its changes.

`- commitEditing` (page 6)
Causes the receiver to attempt to commit any pending edits, returning `YES` if successful or no edits were pending.

Instance Methods

commitEditing

Causes the receiver to attempt to commit any pending edits, returning YES if successful or no edits were pending.

```
- (BOOL)commitEditing
```

Discussion
The receiver invokes commitEditing on any current editors, returning their response. A commit is denied if the receiver fails to apply the changes to the model object, perhaps due to a validation error.

Availability
Available in Mac OS X v10.3 and later.

See Also
- discardEditing (page 7)

Declared In
NSController.h

commitEditingWithDelegate:didCommitSelector:contextInfo:

Attempts to commit any pending changes in known editors of the receiver.

```
-(void)commitEditingWithDelegate:(id)delegate
    didCommitSelector:(SEL)didCommitSelector contextInfo:(void *)contextInfo
```

Parameters
delegate
An object that can serve as the receiver's delegate. It should implement the method specified by *didCommitSelector*.

didCommitSelector
A selector that is invoked on delegate. The method specified by the selector must have the same signature as the following method:

```
- (void)editor:(id)editor didCommit:(BOOL)didCommit contextInfo:(void
*)contextInfo
```

contextInfo

Contextual information that is sent as the `contextInfo` argument to delegate when *didCommitSelector* is invoked.

Discussion
Provides support for the `NSEditor` informal protocol. This method attempts to commit pending changes in known editors. Known editors are either instances of a subclass of `NSController` or (more rarely) user interface controls that may contain pending edits—such as text fields—that registered with the context using `objectDidBeginEditing:` and have not yet unregistered using a subsequent invocation of `objectDidEndEditing:`.

The receiver iterates through the array of its known editors and invokes `commitEditing` on each. The receiver then sends the message specified by the *didCommitSelector* selector to the specified delegate.

The `didCommit` argument is the value returned by the editor specified by `editor` from the `commitEditing` message. The `contextInfo` argument is the same value specified as the *contextInfo* parameter—you may use this value however you wish.

If an error occurs while attempting to commit, for example if key-value coding validation fails, your implementation of this method should typically send the view in which editing is being performed a `presentError:modalForWindow:delegate:didRecoverSelector:contextInfo:` message, specifying the view's containing window.

You may find this method useful in some situations (typically if you are using Cocoa Bindings) when you want to ensure that pending changes are applied before a change in user interface state. For example, you may need to ensure that changes pending in a text field are applied before a window is closed. See also `commitEditing` (page 6) which performs a similar function but which allows you to handle any errors directly, although it provides no information beyond simple success/failure.

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

See Also
- `commitEditing` (page 6)
- `discardEditing` (page 7)
- `objectDidBeginEditing:` (page 8)
- `objectDidEndEditing:` (page 8)

Declared In
`NSController.h`

discardEditing

Discards any pending changes by registered editors.

```
- (void)discardEditing
```

Discussion
The receiver invokes `discardEditing` on any current editors.

Availability
Available in Mac OS X v10.3 and later.

See Also
- commitEditing (page 6)

Declared In
NSController.h

isEditing

Returns YES if there are any editors currently registered with the receiver, NO otherwise.

```
- (BOOL)isEditing
```

Availability
Available in Mac OS X v10.3 and later.

Declared In
NSController.h

objectDidBeginEditing:

Invoked to inform the receiver that *editor* has uncommitted changes that can affect the receiver.

```
- (void)objectDidBeginEditing:(id)editor
```

Availability
Available in Mac OS X v10.3 and later.

See Also
- objectDidEndEditing: (page 8)

Declared In
NSController.h

objectDidEndEditing:

Invoked to inform the receiver that *editor* has committed or discarded its changes.

```
- (void)objectDidEndEditing:(id)editor
```

Availability
Available in Mac OS X v10.3 and later.

See Also
- objectDidBeginEditing: (page 8)

Declared In
NSController.h

Document Revision History

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

Date	Notes
2007-04-02	Updated for Mac OS X v10.5.
2006-05-23	First publication of this content as a separate document.

Document Revision History

Index