

how does canCloseDocumentWithDelegate work?

matt neuburg [how does canCloseDocumentWithDelegate work?](#)

Jul 07 2003, 00:47

The docs say that this is deprecated:

```
- (BOOL) canCloseDocument
```

Instead, you're supposed to use this:

```
- (void) canCloseDocumentWithDelegate: (id) delegate shouldCloseSelector:
(SEL) shouldCloseSelector contextInfo: (void *) contextInfo
```

What I don't understand is how that can prevent the document from closing, since a void is not a BOOL. How does this routine communicate back to the framework that one should not close?

I have the same problem with `shouldCloseWindowController:delegate:shouldCloseSelector:contextInfo:`. It returns void, and its `shouldCloseSelector` callback returns void too. So how does it communicate an answer to the "should" question?

And the docs on this routine say that it "Invokes `shouldCloseSelector` with the result of `canCloseDocumentWithDelegate:shouldCloseSelector:contextInfo:`". What result? As we just said, `canCloseDocumentWithDelegate:shouldCloseSelector:contextInfo:` returns a void! A void is **no** result. We seem to be going round in circles. At least, I am. m.

matt neuburg, phd = <matt...>, <http://www.tidbits.com/matt>
pantes anthropoi tou eidenai oregontai phusei
Subscribe to TidBITS! It's free and smart. <http://www.tidbits.com/>

cocoa-dev mailing list | <cocoa-dev...>
Help/Unsubscribe/Archives: <http://www.lists.apple.com/mailman/listinfo/cocoa-dev>
Do not post admin requests to the list. They will be ignored.

Sherm Pendley [Re: how does canCloseDocumentWithDelegate work?](#)

Jul 07 2003, 02:01

On Sunday, July 6, 2003, at 06:47 PM, matt neuburg wrote:

```
> - (void) canCloseDocumentWithDelegate: (id) delegate
> shouldCloseSelector:
> (SEL) shouldCloseSelector contextInfo: (void *) contextInfo
>
> What I don't understand is how that can prevent the document from
> closing,
> since a void is not a BOOL. How does this routine communicate back to
```

```
> the
> framework that one should not close?
```

If you're calling this method from a controller, the BOOL you're looking for is passed to the callback method you provide, instead of being returned directly.

```
// This snippet will presumably be part of a larger action method
[document canCloseDocumentWithDelegate:self
 shouldCloseSelector:@selector(document:shouldClose:contextInfo:)
 contextInfo:NULL];

// Here is the callback method, which is assumed to be an instance
method belonging
// to the same controller class the above action method is found in.
- (void) document:(NSDocument*)doc shouldClose:(BOOL)shouldClose
contextInfo:(void*)contextInfo {
if (YES == shouldClose) {
    [doc close];
}
}
```

The above is the simplest case; in a more complex scenario, such as a controller method that checks all open documents before exiting the application, your callback may want to set a flag in an instance variable if even one document indicates that it shouldn't be closed. It could even be an all-or-nothing proposition; the callback could simply set the failure flag according to the value of shouldClose, and then the calling code would close all the documents if none of them set the failure flag.

Conversely, if you're overriding this method in an NSDocument subclass, you need to pass YES or NO to the callback, rather than returning it directly to the caller. This is made more complicated by the fact that the selector you need to call is a variable, and that it takes non-object parameters. Thus, a simple performSelector withObject:... message won't suffice - you need to create an NSInvocation and invoke it, like this:

```
- (void)canCloseDocumentWithDelegate: (id)delegate
 shouldCloseSelector: (SEL)shouldCloseSelector
 contextInfo: (void*)contextInfo {

NSMethodSignature *ms;
NSInvocation *inv;
BOOL shouldClose;
```

```
// Do whatever document-specific processing is needed here. If the
document
// is not dirty, shouldClose can be set to YES. If the document is
dirty, you can
// present a sheet or dialog asking if the user wants to save it,
and set shouldClose
// according to the results of that.
```

```
// Just to be paranoid, verify that the delegate responds to the
given selector
```

```
if ([delegate respondsToSelector:shouldCloseSelector]) {

    // Create the invocation object
    ms = [delegate methodSignatureForSelector:shouldCloseSelector];
    inv = [NSInvocation invocationWithMethodSignature:ms];

    // Set the target, selector, and parameters
    [inv setTarget:delegate];
    [inv setSelector:shouldCloseSelector];
    [inv setArgument:&delegate atIndex:2];
    [inv setArgument:&shouldClose atIndex:3];
    [inv setArgument:&contextInfo atIndex:4];

    // Invoke!
    [inv invoke];
}
}
```

Disclaimer: Code typed into email, untested, etc.

sherm--

Though a program be but three lines long, someday it will have to be maintained.

-- The Tao of Programming

cocoa-dev mailing list | <cocoa-dev...>

Help/Unsubscribe/Archives: <http://www.lists.apple.com/mailman/listinfo/cocoa-dev>

Do not post admin requests to the list. They will be ignored.

Sherm Pendley Re: how does canCloseDocumentWithDelegate work?

Jul 07 2003, 03:36

D'oh!

Correction... The first argument passed to the callback method should be the document, not the delegate.

On Sunday, July 6, 2003, at 08:01 PM, Sherm Pendley wrote:

> show quoted text

The above line should be:

```
[inv setArgument:&self atIndex:2];

> [inv setArgument:&shouldClose atIndex:3];
> [inv setArgument:&contextInfo atIndex:4];
>
> // Invoke!
> [inv invoke];
> }
> }
```

sherm--

Heisenberg may have slept here.

cocoa-dev mailing list | <cocoa-dev...>

Help/Unsubscribe/Archives: <http://www.lists.apple.com/mailman/listinfo/cocoa-dev>

Do not post admin requests to the list. They will be ignored.

Cocoa - July 07, 2003

july 2003						
M	T	W	T	F	S	S
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			
Go to today						



Cocoabuilder.com *beta* - Site by Bertrand Mansion - [contact](#)

This site is an archive of mailing lists related to osx and iphone development.

Hint :configure your browser Monospace font in the Preferences.