

MyClass

Inherits From:	SuperClass : Object
Conforms To:	MyProtocol AnotherProtocolLine
Declared In:	MyClass.h
Depends On:	MySecondClass.h misckit/misckit.h

Class Description

Place class description here. The following text might be stupid and impossible to read. It was only intended to give you some cut/copy/paste templates for NeXTs documentation style.
A small description might look like:

- First item.
- Next item.
- Even more items.

But you might choose different one.

Sub paragraphs...and Method lists

If you need to create sub paragraphs...do so. They might look like that.

A collection of methods that you want to give a pointer to might look like the following part.:

```
textWillResize:  
textDidResize:oldBounds:invalid:  
textWillChange:
```

Note: This should also be used inside the documentation of a method if you for example want to list the methods a subclass...or delegate must implement.

But you might also decide to show them like a normal bold **someMethod:With:** selector entry. Listing too many methods in this way might not look very nice. So use it for single references only.

Example code

To show how to use this class you might include some example code sections:

```
[[Application workspace] openFile:"/tmp/README"  
    withApplication:"Edit"];
```

I hope you can use those templates when creating your documentation.

Aloha ± Tomi.

Instance Variables

```

id panel;
id userField;
struct _cFlags1 {
    unsigned int state:1;
    unsigned int highlighted:1;
    unsigned int disabled:1;
    unsigned int editable:1;
} cFlags1;

```

firstID	Just a temp.
secondID	Another temp.
cFlags1.state	The state of the MyClass (0 or 1).
cFlags1.highlighted	True if the MyClass is highlighted.
cFlags1.disabled	True if the MyClass is disabled.
cFlags1.editable	True if the text in the MyClass is editable.

Method Types

Creating new instances	+ new
Doing other work	- myMethod - findText:ignoreCase:backwards:wrap:

Class Methods

new

+ new

Creates, if necessary, and returns the shared instance of NILLoginPanel.

See also: **+ otherAlloc**

Instance Methods

myMethod

- myMethod

Method description here. If parameters reflect a certain instance variable you can write it like
NX_PERIODICMASK (**cflags1.continuous**). A list of possible parameter might look like that:

NX_MOUSEUPMASK
NX_MOUSEDOWNMASK
NX_MOUSEDRAGGEDMASK
NX_PERIODICMASK

But here we either it returns **nil** or **self**.

findText:ignoreCase:backwards:wrap:

- (BOOL)**findText**:(const char *)*string*
 ignoreCase:(BOOL)*ignoreCaseflag*
 backwards:(BOOL)*backwardsflag*
 wrap:(BOOL)*wrapflag*

Here you can copy a multiline ruler. If you need to show the constants a method accepts you might type it in the following way:

Constant

NX_LEFTALIGNED

NX_JUSTIFIED

Alignment

Flush to left edge of the **bodyRect**.

Flush to left and right edges of the **bodyRect**; justified. Not yet implemented.

...or like that:

Constant

NX_WHITE

NX_LTGRAY

NX_DKGRAY

NX_BLACK

Shade

White

Light gray

Dark gray

Black

A **setTextGray:** message doesn't cause the text to be redrawn. Returns TRUE.

Flush to left and right edges object's default font (see **setDefaultFont:**) and uses object's default font (see **setDefaultFont:**) and uses

Methods Implemented by the Delegate

setTextFilter:

- **setTextFilter:**(NXTextFilterFunc)*aFunc*

Sets the text filter function, the function that analyzes text the user enters.

The text filter function is called with the following arguments:

```
[super copyFromZone:zone];
```

This is how example source might be added. How cross class references might look like can be found below. See that the colon between the references is not bold and the dash is in the Symbol font. And don't forget to mention the returned data. Returns **self**.

See also: - **textFilter**, - **textFilter:** (OtherObject), - **moreTextFilter**