CNAppManager

Inherits From: MySuperObject

Conforms To: MyProtocol

Declared In: CNAppManager.h

Depends On: MySecondClass.h

Class Description

Place class description here.

Instance Variables

id **firstID**;
id **secondID**;

firstID	Just a temp.
secondID	Another temp.

Method Types

- myMethod

Class Methods

Instance Methods

textDidChange:
- **textDidChange:***sender*

Method description here.

See also: - **myReference**

app:powerOffIn:andSave:
- **app:***sender* **powerOffIn:**(int)*ms* **andSave:**(int)*aFlag*

Method description here.

See also: - **myReference**

powerOff:
- **powerOff:**(NXEvent *)*theEvent*

Method description here.

See also: - **myReference**

new:
- **new:***sender*

Method description here.

See also: - **myReference**

_pasteIt
- **_pasteIt**

Here you should be able to read and write the documentation that is stored in the MyClass.rtf file.

Well this all could be integrated into HeaderViewer. But there is no API...so I would have to do all that *brain dead* file parsing stuff on my own. I'm not sure if I will do that until it is clear that NeXTSTEP 4.0 does not have a tool like this one.

Keeping track of know bugs, cross references and the support for systems like CVS would be really nice. Add well¼all in a drag&drop fashion.

app:openFile:type:
- (int)**app:***sender* **openFile:**(const char *)*path* **type:**(const char *)*type*

Method description here.

See also: - **myReference**

appAcceptsAnotherFile:
- (BOOL)**appAcceptsAnotherFile:***sender*

Method description here.

See also: - **myReference**

appDidBecomeActive:
- **appDidBecomeActive:***sender*

Method description here.

See also: - **myReference**

appDidInit:
- **appDidInit:***sender*

Method description here.

See also: - **myReference**

appWillInit:
- **appWillInit:***sender*

This is the init part we need to pass before we get the messages that we have to open some files. This app won't load anything other then our preferences.

See also: - **appAcceptsAnotherFile**

appWillTerminate:
- **appWillTerminate:***sender*

If we are escaping with unsaved notes we get the alter panel telling that we should save them.
Regulat quitting will cause a silent autosave.

See also: - **myReference**

copyNotesTemplate
- **copyNotesTemplate**

Method description here.

See also: - **myReference**

escape:
- **escape:***sender*

Escaping will terminate the application without saving the contents of the notes view. This will cause an alert if there are unsaved notes.

See also: - **appWillTerminate**

note:userData:error:
- **note:**(id)*pb* **userData:**(const char *)*udata* **error:**(char **)*ermsg*

This is our service method. It gets triggerd by the services menu and will add a new note with the contents of the specified pasteboard.

See also: - **myReference**

open:
- **open:***sender*

Opening will load the notes file called ClipNotes.rtfd by searching the defined path. If the sender is *self* then the notes window will become key.

See also: - **myReference**

openDefaults
- **openDefaults**

Loads the default values. The *NotesTemplate.rtfd* document might be stored either in the ~/Library section or will be read from the app wrapper.

See also: - **save Defaults**

pasteAsNote:
- **pasteAsNote:***sender*

Method description here.

See also: - **myReference**

preferences
- **preferences**

Method description here.

See also: - **myReference**

print:
- **print:***sender*

Method description here.

See also: - **myReference**

revert:

- **revert:***sender*

Reverts the contents of the notes view to the saved version. If there are unsaved changes we will ask frist .

See also: - **myReference**

save:

- **save:***sender*

Method description here.

See also: - **myReference**

saveDefaults

- **saveDefaults**

Write down all the defaults. The notes template will always go into the user library...but only if it was edited before.

See also: - **myReference**

showInfo:

- **showInfo:***sender*

Dummy action method. Might be used in the future to bring up the info panel from a seperate NIB.

See also: - **show Preferences**

showPreferences:

- **showPreferences:***sender*

Method description here.

See also: - **myReference**