How to use the MiscFindPanel class

Incorporating the find panel into a project

Barring the need to debug the MiscFindPanel class itself (and hopefully you won't have to do that), the find panel bundle can be compiled once and installed into a project directory and (for the most part) forgotten. The most difficult part of the process of adding a find panel to your project is implementing the methods of the SearchableText protocol.

In this version of the MiscFindPanel class, the find panel searches for a "first conformer" (see the document **MiscFindPanel.rtf**) in the same places a responder is searched for when an action message destined for the first responder is sent (for instance, the "cut:" message from the Edit/Cut menu item). The objects in the key and main windows' responder chains, those windows' delegates, and the NXApp object and its delegate are the objects that, potentially, the find panel might operate on. This means that, while the SearchableText protocol could be implemented by any object, the only objects that the find panel might try to operate on are Responders of some type: views in a window and windows themselves typically (the exceptions being the key and main windows' delegates and NXApp's delegate, which may be objects of any class). Most commonly, it is a View of some sort in the main window. For most projects, this is not a problem; the intent in adding a find panel to the application is to operate on a view in the main window.

There are five simple steps to adding a find panel to your project:

1. Implement the SearchableText protocol in one of your objects.

2. Build the *MiscFindPanel.bundle* and install it in your project's directory. Either:
   a. On the command line: "make install INSTALLDIR=*path*" within the *MiscFindPanel/MiscFindPanel* directory, or
   b. In Project Builder: load the file *MiscFindPanel/MiscFindPanel/PB.project* and type "install INSTALLDIR=*path*" in the argument text field before building.

Replace *path* with the path to your project's directory.

3. Copy or link the files *MiscFindPanelClass.h*, *MiscFindPanelClass.m*, *MiscFindPanel.h*, and *SearchableText.h* into your project's directory.

4. Add the new files to your project:
   a. Add *MiscFindPanel.bundle* to "Other Resources".
   b. Add *MiscFindPanelClass.m* to "Other Sources".
   c. Add *MiscFindPanelClass.h, MiscFindPanel.h,* and *SearchableText.h* to "Headers".

5. In Interface Builder, add a Find menu to the Edit menu of the main menu of your application, and enable each of the menu cells (unless you plan on enabling them programmatically). Using the class inspector, add the methods (findNext:, findPrevious:, enterSelection:, jumpToSelection:, orderFrontFindPanel:) to the FirstResponder class. Make one connection from each of the Find menu cells to the First Responder object in the File Window, connecting the appropriate method.

Your application must be linked with the shared library *libNeXT_s*. It probably already is. If you want to strip the main executable of your application, or the install process is going to strip it for you, you need to add the line

```
APP_STRIP_OPTS = $(DYLD_APP_STRIP_OPTS)
```

to the file *Makefile.postamble* (creating the file if it doesn't exist). This will cause those symbols which might be needed by the dynamically loaded MiscFindPanel class to not be removed from the main executable. See the manual pages *rld(3)* and *strip(1)* if you want to know more.

The MiscFindPanel class can also be statically linked into an application, and the localized files added to others in a project's language directories. In this case, the MiscFindPanelClass category could still be used to "catch" messages and forward them to the find panel, but some modifications would have to be made so that the code does not try to dynamically load the MiscFindPanel class. The MiscFindPanel code itself should work without modification. In this case, of course, an application could be "fully" stripped.

The SearchableText categories & string-searching routines

To ease the incorporation of the find panel into a project, two ready-to-use string-searching packages and an implementation of the SearchableText protocol for the Text class are included with the MiscFindPanel distribution, in the directory *MiscFindPanel/SearchCategories/*. See the documents **README** and **MiscSearchText.rtf** in that directory for more information.

Notes on localization

The *MiscFindPanel.bundle* contains translated panels and strings for the languages: English, French, German, Italian, Spanish, and Swedish. If you are not supporting some or all of these languages in your project, you will want to remove the appropriate .lproj directory(s) in the *MiscFindPanel.bundle*, so that if, for instance, your application does not support Spanish, a user will never see a Spanish find panel and the rest of your application in some other language.

If you are supporting more than one language in your application, you will want the Find menu items to be localized as well. The following table shows possible translations for the Find menu items. Copying and pasting the text out of this document may be the easiest way to get the characters with the diacriticals.

English	Find	Find Panel...	Find Next	Find Previous	Enter Selection	Jump to Selection

French	Rechercher	Panneau de recherche...	Rechercher le suivant	Rechercher le prÝcÝdent	Entrer la sÝlection	Aller Õ la sÝlection
German	Suchen	Dialogfenster "Suchen"	Weitersuchen (vorwÙrts)	Weitersuchen (röckwÙrts)	Auswahl öbernehmen	Zur Auswahl springen
Italian	Trova	Pannello di ricerca...	Trova il seguente	Trova il precedente	Riporta la selezione	Salta alla selezione
Spanish	Buscar	Panel de bósqueda...	Buscar siguiente	Buscar anterior	Introducir selecciín	Pasar a selecciín
Swedish	Sðk	Sðkpanel...	Sðk nÙsta	Sðk fðregÚende	Kopiera markering	GÚ till markering

MiscFindPanel miscellanea

Help is not provided for the find panel and its controls. Under NeXTSTEP 3.x, in my opinion, help text from a bundle is not integrated well into the NXHelpPanel, so I've opted not to provide any. Also, help is much more application-specific than the find panel itself, and I also did not want to translate the help text I would have provided into the non-English languages. Help text similar to that contained in Edit.app for its find panel is suitable for this one as well.

The find panel does not save its frame to the defaults database. Simply use the appropriate Window methods if you want this behavior, sending them to the find panel.