

Release 1.7.0 Copyright © 1995 by Don Yacktman. All Rights Reserved.

# *The MiscKit*

<b>Library:</b>	libMiscKit.a
<b>Header File Directory:</b>	/LocalDeveloper/Headers/misckit
<b>Import:</b>	misckit/misckit.h

## Introduction

The MiscKit is a collection of useful classes and Interface Builder™ palettes donated by a multitude of authors around the world to speed up development of mission critical applications. A large number of the MiscKit objects are available on InterfaceBuilder™ palettes to further simplify development of applications. Most of the objects found in the MiscKit fall into one of several categories:

- Foundation classes.

- Data encapsulation classes.
- Classes which encapsulate often used UNIX™ and Mach™ protocols.
- User interface extensions.
- Other object kits.
- Useful examples from test suites used to validate MiscKit objects to complete example applications.
- Functions and Macros.

The size of the MiscKit is often staggering to newcomers. We have tried to at least provide references and pointers to all the neat sections of the kit from this document, but some things might have been missed. Be sure to check out class documentation, header files, and, at times, source files. Even folks who have been using the MiscKit a while can benefit from reviewing things—there's too much here to remember all of it! Don't expect to learn it all in a day, either. It will take months to learn how to use everything in the MiscKit. Don't let that discourage you; the effort will pay you back many times over, beginning with the thousands of lines of code here that you won't have to write. The best thing to do is start using MiscKit classes one by one, learning about things as you need them and find a use for them.

The MiscKit is constantly growing and improving. If you have suggestions for improvement, forward them to the appropriate authors or to the MiscKit developers' mailing list [misckit@byu.edu](mailto:misckit@byu.edu). If you would like to make a contribution to the MiscKit—and you are heartily invited to do so—simply contact the MiscKit administrator [Don\\_Yacktman@byu.edu](mailto:Don_Yacktman@byu.edu).

## MiscKit Classes and Protocols

Below is a diagram showing the heirarchy of MiscKit classes. The `^MiscTree_Browser` example app allows you to interactively view this heirarchy as well as a nearly complete list of MiscKit categories, functions, macros,

types, and constants. It is recommended as a way to help learn what is available in the MiscKit.

MiscKit1.6.1.eps ↵

**Figure 1.** MiscKit classes

## Foundation classes

There are several classes which implement basic data structures. For containing lists of other objects, there are the MiscList, MiscStorage, MiscSortedList, MiscSortedListStorage, MiscLinkedList, MiscQueue, MiscPriorityQueue, MiscStack, and MiscTree. Both MiscList and MiscStorage extend standard NeXT™ objects to use a cursor. For keeping sorted lists of objects or data structures, use MiscSortedList or MiscSortedListStorage respectively. The MiscQueue implements a FIFO (First In, First Out) data structure. If you need to line up objects in a prioritized order, use the MiscPriority queue, which sorts objects according to their priorities, using a heap data structure internally. If you need a LIFO (Last In First Out) data structure, use the MiscStack. Use the MiscTree to create tree data structures with arbitrary numbers of children on each branch.

The MiscDictionary implements a simple key/value table where the keys are MiscStrings and the values can be any object. Arrays of strings may be manipulated with the MiscStringArray and MiscUniqueStringArray objects. Various types of graphs may be handled via the MiscGraphNode class. The MiscTableSource is a special collection which can be used as the data source for a DBTableView. The MiscSearcher class makes it easy to scan a List for text. Future versions of the MiscKit will probably have other basic data structures included as they become available.

Categories of Object, List, and HashTable and several macros extend the usefulness classes, as well. Constructs for sorting, looping, and manipulating the Objective-C runtime exist. Some of the interesting runtime manipulations include morphing one class into another and an <sup>a</sup>object recycler<sup>o</sup>. The MiscTee object allows action messages to be sent to multiple destinations. The MiscDistributor takes this a step further to allow connections across different .nib files. To spread announcements around to various object in a way similar to

the Foundation Kit's™ NSNotificationCenter class, there is the MiscAnnouncer class. The MiscClassDecoder allows detailed information about a class to be dumped out of a running application, great for hackers who like to snoop around each other's work.

## **Data classes**

Several classes in the MiscKit encapsulate simple datum and provide for easy manipulation. For example, MiscTime handles time to just about any degree of accuracy you need, from days of the year down to the nearest microsecond. There's even a MiscStopwatch which allows tracking of elapsed time with high accuracy. For text string handling, there is the MiscString class. There are also three classes which make color manipulation easier and more powerful than ever before: MiscColor, MiscHalftoneColor, and MiscScreenColor. The MiscClassVariable allows creation of a true class variable, an improvement over using static variables. The MiscRtMatrix class encapsulates the RTMatrix data type so that it may be passed across DO connections.

## **OS wrappers**

Some of the MiscKit classes wrap around common OS features. With the MiscKit, you can easily do simple file locking (MiscLockFile) and log errors to a file (MiscLogFile).

Communications may be facilitated via use of the MiscSerialPort and MiscXmodem classes. To aid in networking, there are the MiscSocket, MiscINETSocket, and MiscINETAddress objects.

Multi-threaded applications can benefit from the use of MiscThreadedObject and MiscTimedEntry.

The MiscKit helps applications find their way through the file system. MiscFileFinder, MiscDirectory, \_MiscDirectoryStream, MiscFile, MiscAppFile. Dealing with users and groups is done with the MiscUser and MiscUserGroup classes. UNIX database files (such as the password file) may be parsed with an instance of MiscRecordParser.

The MiscSubprocess and MiscRemoteSubprocess classes allow subprocess to be started, monitored, and

controlled easily. MiscShell takes this a step further and allows a shell script to easily be attached to a user interface. The MiscRemoteApplication makes it easy to connect to another application, launching it if necessary. Communications with NeXT's Mail.app are facilitated with the MiscMailApp and MiscMailSpeaker classes.

## **Appkit™ extensions**

The MiscKit also provides new UI widgets and enhances several of the AppKit™ classes.

The MiscSavePanel extends the standard NeXT SavePanel. The MiscFindPanel makes it easy to add a Find menu and Find¼ panel to an application. The MiscSearchText category extends the AppKit's™ Text object so that it can work seamlessly with the MiscFindPanel. MiscBackWindow implements a window that can always be at the back of the screen or always at the front just like BackSpace.app™. MiscTaggedImage is an NXImage subclass that has a tag, which can be useful when keeping track of large numbers of images.

The MiscClockView implements a clock and calendar like the one seen in Preferences.app™. Calendars are available, too, via the MiscCalendarView class. The MiscDateView handles display of dates in linked TextFields.

Using the various progress views — MiscGaugeView, MiscProgressView, MiscProgressBar, and MiscProgressPie — makes it easy to give the user feedback about long-running operations. The MiscProgressPie looks just like the one seen in older versions of Installer.app — still seen in the Workspace.app™ Processes¼ panel.

The MiscCoolButtons palette contains several prototypes of commonly used buttons, saving developers the time otherwise required to put together all those little buttons every app seems to need. There are also new types of buttons in the MiscKit. The MiscThreeStateButton is a button which can take on a third state and the MiscArrowButton implements a simple way to choose between two possible options while showing both options simultaneously, but with an appearance which is more elegant than two radio buttons.

The MiscValueField combines a TextFieldCell and two ButtonCells to create a new kind of cell, with up/down arrows, which may be used to select among specific values in a list of strings or step through numeric values. The MiscSliderField is similar, combining a Slider and TextField in a single object. The MiscClipTextField and

MiscPathField clip text in special ways so that it is obvious the text has been clipped. Look at the path as displayed in the Workspace app™ attributes inspector for an example. The public domain EmacsText object is currently included on the MiscShell palette, and will hopefully soon be enhanced and available as a new MiscEmacsText object.

Many of the bugs in the NeXT SoundView object are fixed by the MiscSoundView, which also adds extensive new functionality to the object. The MiscVolumeMeter and MiscVolumeLight are enhanced over NeXT's SoundMeter, though they are not subclasses—they can display for all outgoing sound and not just a particular Sound object. The MiscTapper and MiscSoundTracker are used to do this as well as keep track of other parts of NeXT's sound API.

The MiscReadOnlyColorWell allows creation of color wells which are read-only. The MiscDragViews include an abstract superclass and both an Image Well and Icon (File) Well. MiscTabMatrix is a matrix of folder tabs like those seen in the newer InterfaceBuilder™ that is included with NEXTSTEP™ 3.3 and EOF™. For enhancing the cosmetics of GUI displays, MiscColorView displays a rectangle of color and MiscPaperView draws a paper°. MiscAwakeAction allows an action message to be sent when a .nib file is unarchived, to allow for things such as initialization of GUI controls.

The MiscLogSlider implements a slider whose values are logarithmic as the slider traverses its path. The MiscCircularSlider is a slider in the shape of a circle. It can look like a regular slider bent around (ring-like), a shuttle wheel, or a pie chart.

A target/action paradigm is implemented for the View class via the MiscTargetActionView. The MiscFramingView draws a colored background surrounded by a frame.

MiscMatrix is a subclass of Matrix within which the rows and columns may each have their own widths. MiscSelectionMatrix adds extra functionality for handling selections to the Matrix class.

MiscTableScroll is a re-write of the NXTableView that you find in DBKit/EOF applications. It is easier to modify than the NeXT object since source code is provided.

There are also several Application class categories which provide the programmer with extra information about

the runtime environment and make it easier to do things such as drawing in the application's icon. Several categories of Window and Text also exist, to further enhance the AppKit™.

## Other <sup>a</sup>kits<sup>o</sup>

The MiscKit also contains several classes which may be used to painlessly implement a typical Info submenu including an animated Info¼ panel, registration and order form panels, suggestions to the author, and other features. The MiscInfoController, MiscInfo, MiscOrderForm, and MiscRegistration classes work together to provide this service. Many Info¼ menus, such as those in Stone Design's™ apps, also sport the <sup>a</sup>Show/Hide Menus<sup>o</sup> option; you can use the MiscExplodingMenu to implement it in your app.

If you need swapping views like in Preferences.app™ or inspectors, there are two kits (MiscSwapKit and MiscInspectorKit) which contain basic classes that may be used as a starting point. At the moment there is no tutorial for the inspector kit—it is in the works—so it is recommended that you obtain the source code for the <sup>a</sup>BeakerBoy<sup>o</sup> application to see how these objects can be put to use in a real application.

A complete document control system is currently in the works for the MiscKit; this will make it very easy to create multiple document applications using the MiscKit. The currently available (but in alpha test) classes from this kit are: MiscNibController, MiscDocument, MiscDocManager, and MiscDocType.

The MiscGISKit handles conversions between various coordinate systems as well as a way to store coordinate information. It includes the following classes: MiscCoord, MiscMathCoord, MiscCartesianCoord, MiscCylindricalCoord, MiscSphericalCoord, MiscWorldCoord, MiscUTMCoord, MiscZoneUTMCoord, MiscUKUTMCoord, MiscIrelandUTMCoord, MiscIrelandOldUTMCoord, MiscUTMConstants, MiscCoordConverter, MiscMathCoordConverter, MiscPlanetCoordConverter, and MiscIrelandCoordConverter.

If you need template merging capabilities, the MiscMergeKit should nicely fill your needs. The MiscMergeDriver, MiscMergeEngine, MiscMergeCommand, and MiscIfStack implement a very powerful and extensible merge engine.

## Example applications

There are two main types of example applications in the MiscKit. First are the validation/test suites which were (and are) used to test the MiscKit objects. These simple programs, usually run from the command line, put the MiscKit objects through their paces—some more thoroughly than others. There are currently programs to test the MiscString, MiscLogFile, MiscLockFile, and MiscPriorityQueue classes as well as several others. These are in the source distribution but are not normally installed when building and subsequently installing the kit. Not all the test suites are provided, however, since it depends upon the object's author to donate them.

There are also applications which make use of MiscKit classes in a setting which is (at least slightly) useful. There is MiscStringService, which uses the MiscString class to provide a few useful Services menu items. The TreeView application displays a scrolling view of a tree, useful for the display of heirarchical data such as inheritance trees. If you'd like to compare the speed of the MiscKit's searching algorithms to the slower built in routines, try out SearchBench. You'll be amazed. As a simple example of the MiscSerialPort, there is the TinyTerm application which allows you to send and receive data through one of the serial ports. The receiptfilter program allows you to monitor outgoing NeXTMail read-receipts and choose to send (or not send) them. There are many newer examples which are far more impressive—take a wander through the MiscKit examples for a feel of just some of the things you can accomplish easily with the MiscKit. Be sure not to miss UNIX-HatersTool, UsersAndGroups, qlipo, and autodoc! Some extremely useful Perl scripts—programmer's aids—may also be found here.

Also included are several templates which can be used by developers to create documentation which follows the same styles as the NeXT™ on-line documentation. Sample InterfaceBuilder™ files are also provided to demonstrate parts of the MiscKit, such as the Info menu. Sample ProjectBuilder™ projects exist to help jump-start MiscKit-based projects. These files can also be used as templates by developers.

## MiscKit Functions and Macros

The MiscKit, although it is primarily a library of objects, also includes several useful functions. The currently

documented functions provide daemon, debugging, parsing, searching, or pattern matching algorithms.

## Daemons

To aid in the creation of UNIX daemons, the following functions are provided:

- `daemonize()`, `daemon_log()`, `daemon_exit()`, `Misc_daemon_debug()`, `Misc_daemon_set_debug()`

There are also a few C macros available:

- `daemon_assert()`, `daemon_alloc()`, `daemon_free()`

## Debugging

Some MiscKit objects will print helpful diagnostics to `stderr` if you turn on MiscKit debugging.

- `Misc_DebugState()`, `Misc_SetDebugState()`, `Misc_DebugOn()`, `Misc_DebugOff()`

## MiscParseTableFile algorithms

These functions make it easy to parse the `^table^` files such as `*.addresses`, `PB.project`, `data.classes`, etc. that are found throughout NEXTSTEP™.

- `MiscParseTableFile()`, `MiscParseTableStream()`, `MiscWriteTableFile()`, `MiscWriteTableStream()`

## Fast searching algorithms

There is a series of functions which implement the Tuned Boyer-Moore algorithm for searching literal string patterns.

- Misc\_TBMKpattern\_alloc(), Misc\_TBMKpattern\_free(), Misc\_TBMKsearch()

## **Regular expression searching package**

The MiscKit includes the public domain regular expression searching package written by Tatu Ylonen. This package includes the following functions:

- re\_set\_syntax(), re\_compile\_pattern(), re\_match(), re\_match\_2(), re\_search(), re\_search\_2(), re\_compile\_fastmap(), re\_comp(), re\_exec()

## **Miscellaneous**

The MiscKit includes several useful macros. Take a look at the MiscBase.h, FREE.h, SELECT.h, MiscListIteration.h, and MiscKeys.h headers to see the full treatment. Using these macros could lead to cleaner, more concise code.

All trademarks used herein are owned by their respective owners. We're just borrowing them to give you a frame of reference by which you can understand what the MiscKit does.