MiscNotificationCenter

Inherits From: NSNotificationCenter

Conforms To: NSObject (NSObject)

Declared In: MiscNotificationCenter.h

Class Description

MiscNotificationCenter is a subclass of OpenStep's NSNotificationCenter which provides a number of additional functionalities to notification centers. The various sets of additional functionality are explained in greater detail below.

Although an object of the MiscNotificationCenter class cannot be installed as the default NSNotificationCenter, some of the same effect (at a small cost to efficiency) can be achieved by "chaining" an instance of this class to the default notification center. For example, suppose you wanted a MiscNotificationCenter to substitute a proxy object for objects in each notification before sending the notification to the observers. You would register these objects you wish to hide with

the "Hidden Objects" methods, explained below, then chain the MiscNotificationCenter instance to the default notification center:

```
myNotificationCenter = [[MiscNotificationCenter alloc] init];
...
[myNotificationCenter addObject:[NSNotificationCenter defaultCenter] selector:@selector(postNotification:) name:nil object:nil];
```

In this code example, the default center will receive all notifications arriving at *myNotificationCenter*. Observers can register with the default NSNotificationCenter or your center to get any notifications sent to your center. Note that in this case, notifications posted to the default center with `[[NSNotificationCenter defaultCenter] postNotification:aNotification]` (as does probably all code written by other people that you are also using, like the AppKit) will not have "hidden object" filtering applied to them. You could also do the reverse:

```
myNotificationCenter = [[MiscNotificationCenter alloc] init];
...
[[NSNotificationCenter defaultCenter] addObject:myNotificationCenter selector:@selector(postNotification:) name:nil object:nil];
```

Here, your notification center will receive all notifications sent to the default center, as well as directly to your center, and can have "hidden object" filtering applied to them. However, observers who have registered directly with the default notification center (as do probably all objects written by other people that you are also using, like the AppKit) will receive notifications sent to the default center before any filtering can be applied.

If you really need some functionality that this class provides in the default center, having a subclass of NSNotificationCenter pose as NSNotificationCenter may be the only solution. Note that MiscNotificationCenter cannot pose as NSNotificationCenter; you need to create your own subclass to do that (for instance, give the subclass a static

variable, an additional class method +setDefaultCenter: and override +defaultCenter; then install a MiscNotificationCenter instance as the default center).

Hidden Objects

Occasionally there are times when you do not want to reveal the existance of an object (or make the object itself available). Perhaps, for instance, you do not want the object revealed over Distributed Objects (where some other object may get a hold of it and make mischief). Or perhaps the object is internal to a subsystem, access to which should only go through a facade class/object which you've created.

In such cases, you may have taken appropriate precautions, but one place where objects might be inadvertantly leaked is in notifications. The two "hidden object" methods of MiscNotificationCenter can be used to register and unregister objects which are filtered from notifications sent to the center before being passed on to observers of the notification.

Registering Hidden Objects and Proxies

- (void)**addHiddenObject:**(id)*anObject*
  withProxy:(id)*aProxy*
  Registers *aProxy* to replace *anObject* when *anObject* appears as the object of a notification posted to a MiscNotificationCenter. *aProxy* may be nil. If the object is already registered, it is re-registered with (possibly different) *aProxy*.

- (void)**removeHiddenObject:**(id)*anObject*
  Removes the registration of *anObject* as a hidden object.