NSNotificationCenter

Inherits From: NSObject

Declared In: foundation/NSNotification.h

Class Description

The NSNotificationCenter class defines the behavior of notification-center objects (or simply, *notification centers*). A notification center is essentially a notification dispatch table. It notifies all observers of notifications meeting the specific criteria of notification name and object. Client objects register themselves as observers of a specific notification originating in another object. When the condition occurs to signal a notification, that other object notifies all of its observers by posting an appropriate notification object to the notification center. (See the class specification of NSNotification for more on notification objects.) The notification center dispatches a message to each observer (using the selector provided by the observer), with the notification as the sole argument.

As an example, suppose you have an array of text-converter objects (for instance, MIF to RTF or RTF to ASCII), whose services a word-processing application can access. The word-processing application has a client object that wants to be notified when converter objects are added to or removed from the array, allowing the application to reflect the available options in its menus. The application would register itself as an observer by sending the following message to the notification center:

```
[[NSNotificationCenter defaultCenter] addObserver:self
       selector:@selector(updateConvMenu:)
       notificationName:@"NSConvertersChanged" object:converterArray];
```

When a user installs or removes a converter, the object managing the array sends the following message to the notification center:

```
[[NSNotificationCenter defaultCenter]
postNotificationName:@"NSConvertersChanged" object:self];
```

The notification center identifies all observers of converterArray who are interested in the NSConvertersChanged notification and invokes the method they specified in the selector argument of **addObserver:selector:notificationName:object**. In the case of our example observer, that is updateConvMenu:

```
– updateConvMenu:(NSNotification *)notification
{
```

```
    NSEnumerator *enumerator = [[notification object]
                                objectEnumerator];
    id object;

    while (object = [enumerator nextObject]) {
        // update menu ...
    }
    // ...
}
```

There is one notification center per task. Subclassing NSNotificationCenter is not recommended.

Instance Variables

None declared in this class.

Method Types

Accessing the default center	+ defaultCenter
Adding and removing observers	– addObserver:selector:notificationName:object: – removeObserver: – removeObserver:notificationName:object:
Posting notifications	– postNotification: – postNotificationName:object:

Class Methods

defaultCenter

+ (NSNotificationCenter *)**defaultCenter**

Returns the notification-center object for the current task.

Instance Methods

addObserver:selector:notificationName:object:

– (void)**addObserver:***anObserver*
selector:(SEL)*aSelector*
notificationName:(NSString *)*notificationName*
object:*anObject*

By invoking this method, an object that wants to be notified of a notification registers itself as an observer of the object originating the notification. The method selector specified in *aSelector* must have one and only one argument (which, by default, is the notification object). If *notificationName* is **nil**, the notification center notifies the observer of all notifications with an object matching *anObject*. If *anObject* is **nil**, the notification center notifies the object of all notifications with the notification name, regardless of object. This method does not return anything.

The notification center does not retain the observer object or the object specified in *anObject*. Therefore, you should always send **removeObserver:** to the notification center before invalidating these objects.

See also: – **removeObserver:**

postNotification:

– (void)**postNotification:**(NSNotification *)*aNotification*

Posts the notification object *aNotification* to the notification center. You can create this object with the NSNotification class method **notificationWithName:object:**.

See also: – **postNotificationName:object:**

postNotificationName:object:

– (void)**postNotificationName:**(NSString *)*notificationName* **object:***anObject*

Creates a notification name *notificationName* from originating object *anObject* and posts it to the notification center. This method is the preferred one for posting notifications. To post anonymously, make *anObject* **nil**.

See also: – **postNotification:**

removeObserver:

– (void)**removeObserver:***anObserver*

Removes the *anObserver* object from all notification associations in the notification center. Be sure to invoke this method (or **removeObserver:notificationName:object:**) before deallocating the observer object or any object specified in **addObserver:selector:notificationName:object:**.

See also: – **addObserver:selector:notificationName:object:**, – **removeObserver:notificationName:object:**

removeObserver:notificationName:object:

– (void)**removeObserver:***anObserver*
notificationName:*(NSString *)notificationName*
object:*anObject*

Removes all *anObserver* objects with the same *notificationName* and *anObject* (even when *anObject* is nil) from the notification center. Be sure to invoke this method (or **removeObserver:**) before deallocating the observer object or any object specified in **addObserver:selector:notificationName:object:**.

See also: – **addObserver:selector:notificationName:object:**, – **removeObserver:**