

NSNotification Class Cluster

Class Cluster Description

NSNotification objects contain notification information communicated from an object to that object's observers. The NSNotification cluster's single public class, NSNotification, declares the programmatic interface for notifications.

The objects you create using this class are referred to as *notification objects* (or simply, *notifications*). They are immutable objects. Because of the nature of class clusters, notification objects are not actual instances of the NSNotification class but of one of its private subclasses. Although a notification object's class is private, its interface is public, as declared by the abstract superclass, NSNotification. (See ^aClass Clusters^o in the introduction to the Foundation Kit for more information on class clusters and creating subclasses within a cluster.)

You instantiate a notification object directly by sending the **notificationWithName:object:** message to the NSNotification class object. You can also create notifications indirectly, through the NSNotificationCenter class.

NSNotification

Inherits From: NSObject

Conforms To: NSCopying

Returns the name of the notification. Examples of this might be `^PortIsInvalid` or `^PhoneRinging`. Typically, you invoke this method on the notification object passed in to your notification-handling or notification-dispatch method.

You can compose notification names with any string. To avoid name collisions, however, you might want to use a prefix that is specific to your application.

notificationObject

- notificationObject

Returns the object that initiated the notification or that is somehow connected to the notification. Typically, you invoke this method on the notification object passed in to your notification-handling or notification-dispatch method.

To illustrate this, let's assume that you've added your object as an observer of a port. When that port dies, the object monitoring that port will post an `^NSPortInvalid` notification to the Notification Center. The Notification Center then invokes the method you've registered to handle that notification:

```
- (void)handlePortDeath:(NSNotification *)notification
{
    // ...
    [self reclaimResourcesForPort:[notification object]];
    // ...
}
```