EODelayedObserver

Inherits From:	NSObject
Conforms To:	EODelayedObserving NSObject (NSObject)
Declared In:	EOControl/EOObserver.h

Class Description

EODelayedObserver is an abstract superclass defining the basic functionality for coalescing change notifications for multiple objects and postponing notification according to a prioritized queue. See the EODelayedObserverQueue class specification for general information.

Creating a Subclass of EODelayedObserver

EODelayedObserver implements the basic **objectWillChange:** method to simply enqueue the receiver on an EODelayedObserverQueue. Regardless of how many of these messages the receiver gets during the run loop, it receives a single **subjectChanged** message from the queue—at the end of the run loop. In this method the EODelayedObserver can check for changes and take whatever action is necessary. Subclasses should record objects they're interested in, perhaps in an **init** method, and examine them in **subjectChanged**. An EOAssociation, for example, examines each of the EODisplayGroups it's bound to in order to find out what has changed. Another kind of subclass might record each changed object for later examination by overriding **objectWillChange:**, but it must be sure to invoke **super**'s implementation when doing so.

The rest of EODelayedObserver's methods have meaningful, if static, default implementations. EODelayedObserverQueue sends change notifications according to the priority of each enqueued observer. EODelayedObserver's **priority** method returns EOObserverPriorityThird. Your subclass can override it to return a higher or lower priority, or to have a settable priority. The other method a subclass might override is **observerQueue**, which returns a default EODelayedObserverQueue normally shared by all EODelayedObservers. Because sharing a single queue keeps all EODelayedObserver's synchronized according to their priority, you should rarely override this method, doing so only if your subclass is involved in a completely independent system.

A final method, **discardPendingNotification**, need never be overridden by subclasses, but must be invoked from their implementation of **dealloc**. This prevents observers from being sent change notifications after they've been deallocated.

Adopted Protocols

EOObserving	– objectWillChange:

Method Types

Change notification	– subjectChanged
Canceling change notification	– discardPendingNotification
Getting the queue and priority	– observerQueue – priority

Instance Methods

discardPendingNotification

– (void)**discardPendingNotification**

Sends a **dequeueObserver:** message to the receiver's EODelayedObserverQueue to clear it from receiving a change notification. A subclass of EODelayedObserver should invoke this method in its implementation of **dealloc**.

See also: – **observerQueue**

objectWillChange:

@protocol EOObserving
– (void)**objectWillChange:**(id)*anObject*

Implemented by EODelayedObserver to enqueue the receiver on its EODelayedObserverQueue. Subclasses shouldn't need to override this method; if they do, they must be sure to invoke **super**'s implementation.

See also: – **observerQueue**, – **enqueueObject:** (EODelayedObserverQueue)

observerQueue

– (EODelayedObserverQueue *)**observerQueue**

Overridden by subclasses to return the receiver's designated EODelayedObserverQueue. EODelayedObserver's implementation returns the default EODelayedObserverQueue.

See also: + **defaultObserverQueue**

priority

– (EOObserverPriority)**priority**

Overridden by subclasses to return the receiver's change notification priority, one of:

EOObserverPriorityImmediate	EOObserverPriorityFourth
EOObserverPriorityFirst	EOObserverPriorityFifth
EOObserverPrioritySecond	EOObserverPrioritySixth
EOObserverPriorityThird	EOObserverPriorityLater

EODelayedObserver's implementation returns EOObserverPriorityThird. See the EODelayedObserverQueue class specification for more information on priorities.

subjectChanged

– (void)**subjectChanged**

Implemented by subclasses to examine the receiver's observed objects and take whatever action is necessary. EODelayedObserver's implementation does nothing.