David Fedchenko -- 16 June 1994

MiscTee

Inherits From: Object

Declared In: MiscTee.h

Class Description

MiscTee is a class to allow a message to be sent to a single object and have it then be propagated to many others automatically. It is meant primarily for use within its accompanying palette which allows the connections to all the objects to be made graphically, but do what you will with it.

There are two ways an object can be connected to a MiscTee which provide different functionality. The first is to connect it with an action selector. This is meant for providing multiple targets each with their own actions for any other object. You simply connect the sending object to the MiscTee as the target with **ping:** as the action and the MiscTee will then send each object in its list its own private action message using the original sender as the sender so the MiscTee is transparent. You could use this, for example, to synchronize a TextField with a Slider **and** a controller object all from within Interface

Builder.

The second way to connect an object is without an action. This then allows messages that are sent to the MiscTee to be forwarded to these objects for handling (without also dealing with any action side effects). This is useful if you want to have two different delegates for the same object that each take care of particular functionality. Keep in mind, however, that the basic Object methods as well as the few new methods below will be handled by the MiscTee object, so be aware of this when setting things up, so you don't have strange bugs.

Instance Variables

id **idConnections**;

idConnections The list of objects to forward messages to.

Method Types

Creating and freeing instances	±€init ±€free
Making and Using Connections	± addConnection:with: ± ping: ± removeConnection:

Utility Methods	± forward:: ±(BOOL) respondsTo: ± read: ±€write:

Instance Methods

addConnection:with:
±€**addConnection:***anObject*
with:(SEL)*anAction*

Adds *anObject* to the connection List. If *anAction* is **nil** the object will only be forwarded messages that MiscTee doesn't understand inself. If *anAction* is a selector, it will be stored with the object and the action will be sent when **ping:** is sent to the MiscTee. Returns **self**.

See also: ±€**forward::, ping:, removeConnection:, respondsTo:**

init
± **init**

Initializes the List object that will hold all the connected objects.

See also: ±€**free**

forward::

±€**forward:**(SEL)*aSelector*
:(marg_list)*argFrame*

Returns the value returned by the first object in the connection list that respondsTo: *aSelector.* This return value still has the regular forward:: restriction that the type of the value must fit in the space of an **id**. Since there is only **addConnection:with:** and **ping:** defined in this class it's very likely that everything will get forwarded properly. Because MiscTee is a subclass of Object, you will not be able to forward:: any of the regular Object messages (or, for that matter, any Object Category messages) as they will function for the MiscTee object itself.

See also: ±€**addConnection:with:, ping:, respondsTo:**

free

±€**free**

Frees the List object that holds the connections, but not the connected objects.

See also: ±€**init**

ping:

±€**ping:***sender*

Walks through the connection List and sends each object in the List it's choosen action message with *sender* as the *sender* of said action. This keeps the MiscTee invisible in the process. Objects connected without actions will be left alone. If **sender** is in the connection List it will be skipped. This allows one MiscTee to be a hub of coordination for several objects that all need to be kept in sync. Returns **self**.

See also: ±€**addConnection:with:, forward::, respondsTo:**

read:

±€**read:**(NXTypedStream *)*stream*

Reads the connection List object from the stream to recreate the connections that were setup when the MiscTee object was archived.

See also: ±€**write:**

removeConnection:

±€**removeConnection:***anObject*

Removes *anObject* from the connection List if it's in the List. Returns **self**.

See also: ±€**addConnection:with:**

respondsTo:

±€(BOOL)**respondsTo:**(SEL)*aSelector*

Returns **YES** if any object in the connection List respondsTo: *aSelector.* The type of connection is irrelevant, all objects are asked. Returns **NO** if none of the connected objects respond to *aSelector.* Also returns **YES** for the methods that MiscTee respondsTo: ± in which case the MiscTee will perform the requested method if it is called.

See also: ±€**addConnection:with:, forward::, ping:**

write:

±€**write:**(NXTypedStream *)*stream*

Writes the connection List object out to the stream.

See also: ±€**read:**