

```
perform:(SEL)aSelector  
  with:anObject  
  afterDelay:(int)ms  
  cancelPrevious:(BOOL)flag
```

Registers a timed entry to send an aSelector message to the receiver after a delay of at least ms milliseconds, provided ms is 1 or greater. This method returns before the aSelector message is sent. Passing zero causes the message to be sent as soon as possible after the application resumes getting user events, though not immediately. If you pass a negative number for the delay, the message is never sent use this form to cancel the previous request registered with this method. In all cases, this method returns self.

The timed entry that this method registers can be called only after the application finishes responding to the current event and is ready to get the next event. Therefore, program activity could delay the message well beyond ms milliseconds. The timed entry is registered at a priority of NX_RUNMODALTHRESHOLD, which means that it can be called when getting an event in the main event loop or in a modal event loop for an attention panel, but not during a modal loop for a button, slider, or other control device.

The aSelector method should not have a significant return value and should take a single argument of type id anObject will be the argument passed in the message.

If flag is YES and another perform:with:afterDelay:cancelPrevious: message is sent to the same receiver to have it perform the same aSelector method, the first request to perform the aSelector method is canceled. Thus successive perform:with:afterDelay:cancelPrevious: messages can repeatedly postpone the aSelector message.

If flag is NO, each perform:with:afterDelay:cancelPrevious: message will cause another delayed aSelector message to be sent.

This method permits you to register an action in response to a user event (such as a click), but delay it in case subsequent events alter the environment in which the action would be performed (for example, if the click turns out to be double-click). It can also be used to postpone a message that updates a display until after a number of changes have accumulated, or to delay a free message to an object until after the application has finished responding to the current event. (Application's delayedFree: method offers another way to delay free messages.)

```
perform:with: (Object), delayedFree: (Application class)
```