NSResponder

Inherits From: NSObject

Declared In: appkit/NSResponder.h

Managing the Next Responder

- (NSResponder *)**nextResponder**   Returns the receiver's next responder.
- (void)**setNextResponder:**(NSResponder *)*aResponder*
                                                        Makes *aResponder* the receiver's next responder.

Determining the First Responder

- (BOOL)**acceptsFirstResponder**   Subclasses override to accept or reject first responder status.
                                                           NSResponder's implementation simply returns NO.
- (BOOL)**becomeFirstResponder**   Notifies the receiver that it's the first responder.
- (BOOL)**resignFirstResponder**   Notifies the receiver that it's not the first responder.

Aiding Event Processing

- (BOOL)**performKeyEquivalent:**(NSEvent *)*theEvent*
                                                        Subclasses override to respond to keyboard input.   NSResponder's
                                                           implementation simply returns NO to indicate *theEvent* isn't handled.
- (BOOL)**tryToPerform:**(SEL)*anAction*   Aids in dispatching action messages.   Returns YES if a
      with:(id)*anObject*   Responder in the responder chain can perform the   *anAction* method,
                                                        which takes the single argument *anObject*.

Forwarding Event Messages

- (void)**flagsChanged:**(NSEvent *)*theEvent*

  Subclasses override to handle flags-changed events.   NSResponder's implementation passes the message to the receiver's next responder.

- (void)**helpRequested:**(NSEvent *)*theEvent*

  Causes the Help panel to display the help attached to the receiver.   If there's no attached help, passes the message to the receiver's next responder.

- (void)**keyDown:**(NSEvent *)*theEvent*

  Subclasses override to handle key-down events.   NSResponder's implementation passes the message to the receiver's next responder.

- (void)**keyUp:**(NSEvent *)*theEvent*

  Subclasses override to handle key-up events.   NSResponder's implementation passes the message to the receiver's next responder.

- (void)**mouseDown:**(NSEvent *)*theEvent*

  Subclasses override to handle mouse-down events.   NSResponder's implementation passes the message to the receiver's next responder.

- (void)**mouseDragged:**(NSEvent *)*theEvent*

  Subclasses override to handle mouse-dragged events.   NSResponder's implementation passes the message to the receiver's next responder.

- (void)**mouseEntered:**(NSEvent *)*theEvent*

  Subclasses override to handle mouse-entered events.   NSResponder's implementation passes the message to the receiver's next responder.

- (void)**mouseExited:**(NSEvent *)*theEvent*

  Subclasses override to handle mouse-exited events.   NSResponder's implementation passes the message to the receiver's next responder.

- (void)**mouseMoved:**(NSEvent *)*theEvent*

  Subclasses override to handle mouse-moved events.   NSResponder's implementation passes the message to the receiver's next responder.

- (void)**mouseUp:**(NSEvent *)*theEvent*

  Subclasses override to handle mouse-up events.   NSResponder's implementation passes the message to the receiver's next responder.

- (void)**noResponderFor:**(SEL)*eventSelector*

  Responds to an event message that has reached the end of the responder chain without finding an object that can respond.   When the event is a key down, generates a beep.

- (void)**rightMouseDown:**(NSEvent *)*theEvent*

  Subclasses override to handle right mouse-down events.   NSResponder's implementation passes the message to the receiver's next responder.

- (void)**rightMouseDragged:**(NSEvent *)*theEvent*

  Subclasses override to handle right mouse-dragged events. NSResponder's implementation passes the message to the receiver's next responder.

- (void)**rightMouseUp:**(NSEvent *)*theEvent*

  Subclasses override to handle right mouse-up events.   NSResponder's implementation passes the message to the receiver's next responder.

Services Menu Support

- (id)**validRequestorForSendType:**(NSString *)*typeSent*
    returnType:(NSString *)*typeReturned*   Subclasses override to determine which Services menu items are enabled at a given time.   Returning **self** enables services that can receive *typeSent* pasteboard types and can return *typeReturned* pasteboard types.   Returning **nil** disables them.   NSResponder's implementation passes the message to the receiver's next responder.