

## NSEvent

**Inherits From:** NSObject

**Conforms To:** NSObject,  
NSCopying

**Declared In:** appkit/NSEvent.h

### Getting General Event Information

- (NSDPSServerContext \*)**context**
- (NSPoint)**locationInWindow**
- (unsigned int)**modifierFlags**
- (NSTimeInterval)**timestamp**
- (NSEventType)**type**
  
- (NSWindow \*)**window**
- (int)**windowNumber**

Returns the PostScript context of the event.

Returns the event's location in the base coordinate system of its window.

Returns an integer bitfield containing modifier-key flags.

Returns the time the event occurred in seconds since system startup.

Returns the type of the event (left mouse up, right mouse dragged, key down, etc.).

Returns the window object associated with the event.

Returns the number of the window associated with the event.

### Getting Mouse Event Information

- (int)**clickCount**
- (int)**eventNumber**
  
- (float)**pressure**

Returns the number of mouse clicks associated with the mouse event.

Returns the event number of the latest mouse-down event. This information is also useful for handling tracking events.

Returns a value indicating the pressure applied to the input device (used for appropriate devices, not mice).

## Getting Key Event Information

- (NSString \*)**characters** Returns the character code (a string of characters generated by the key event).
- (NSString \*)**charactersIgnoringModifiers** Returns the string of characters generated by the key event as if no modifier key had been pressed (except for Shift).
- (BOOL)**isARepeat** Returns whether the key event is being repeated (user is holding down the key).
- (unsigned short)**keyCode** Returns the code that maps to a key on the keyboard.

## Getting Tracking Event Information

- (int)**trackingNumber** Returns the number that identifies the tracking rectangle.
- (int)**userData** Returns data arbitrarily associated with the event.

## Getting Kit-Defined Event Information

- (int)**data1** Returns special data associated with the event.
- (int)**data2** Returns special data associated with the event.
- (short)**subtype** Returns the identifier of the kit-defined event.

## Managing Periodic Events

- + (void)**startPeriodicEventsAfterDelay:(NSTimeInterval)delaySeconds  
withPeriod:(NSTimeInterval)periodSeconds** Start tracking loop with period *periodSeconds* after delay *delaySeconds* for current thread.
- + (void)**stopPeriodicEvents** Stop tracking loop for current thread.

## Creating NSEvent Objects

- + (NSEvent \*)**enterExitEventWithType:(NSEventType)type  
location:(NSPoint)location** Returns an NSEvent object initialized with general event

**modifierFlags:**(unsigned int)*flags* data and information specific to mouse tracking  
**timestamp:**(NSTimeInterval)*time* (*eventNum, trackingNum, userData*).  
**windowNumber:**(int)*windowNum*  
**context:**(NSDPSServerContext \*)*context*  
**eventNumber:**(int)*eventNum*  
**trackingNumber:**(int)*trackingNum*  
**userData:**(int)*userData*

+ (NSEvent \*)**keyEventWithType:**(NSEventType)*type*  
**location:**(NSPoint)*location* Returns an NSEvent object initialized with general event  
**modifierFlags:**(unsigned int)*flags* data and information specific to keyboard events (*keys,*  
**timestamp:**(NSTimeInterval)*time* *repeatKey, code, ukeys*). (*ukeys* sets the unmodified  
**windowNumber:**(int)*windowNum* character string.)  
**context:**(NSDPSServerContext \*)*context*  
**characters:**(NSString \*)*keys*  
**charactersIgnoringModifiers:**(NSString \*)*ukeys*  
**isARepet:**(BOOL)*repeatKey*  
**keyCode:**(unsigned short)*code*

+ (NSEvent \*)**mouseEventWithType:**(NSEventType)*type*  
**location:**(NSPoint)*location* Returns an NSEvent object initialized with general event  
**modifierFlags:**(unsigned int)*flags* data and information specific to mouse events  
**timestamp:**(NSTimeInterval)*time* (*eventNum, clickNum, pressureValue*).  
**windowNumber:**(int)*windowNum*  
**context:**(NSDPSServerContext \*)*context*  
**eventNumber:**(int)*eventNum*  
**clickCount:**(int)*clickNum*  
**pressure:**(float)*pressureValue*

+ (NSEvent \*)**otherEventWithType:**(NSEventType)*type*  
**location:**(NSPoint)*location* Returns an NSEvent object initialized with general event  
**modifierFlags:**(unsigned int)*flags* data and information specific to kit-defined events  
**timestamp:**(NSTimeInterval)*time* (*subType, data1, data2*).  
**windowNumber:**(int)*windowNum*  
**context:**(NSDPSServerContext \*)*context*  
**subtype:**(short)*subType*  
**data1:**(int)*data1*

**data2:**(int)*data2*