

**EventRecord** structure

```
#include <Events.h>
```

typedef struct	<b>EventRecord</b>	{	<u>Size</u>	<u>Offset</u>	<u>Description</u>
<u>short</u>	what;		2	0	Type of event (0=nothing) (for more information, see <a href="#">Event Types</a> )
<u>long</u>	message;		4	2	Varies per event type (see Notes)
<u>long</u>	when;		4	6	Timestamp (ticks since system startup)
<u>Point</u>	where;		4	10	Mouse position, in global coordinates
<u>short</u>	modifiers;		2	14	State of shift, cmd, option, ctrl keys et al.
} <b>EventRecord</b> ;			16		

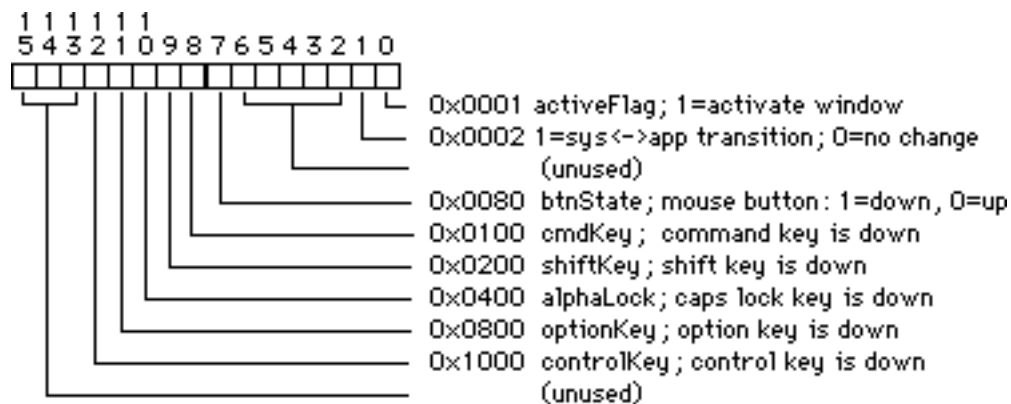
Note: No typedef exists for an (EventRecord \*)

Notes: The contents of the message field varies, depending upon the type of event (found in the what field), as follows:

**Event Type   Contents of EventRecord.message**

nullEvent      (undefined)  
 keyDown bits 0-7=char; bits 8-15=key code (see [GetNextEvent](#))  
 keyUp      (same)  
 mouseDown      (undefined) Note: mouse position is in the where field  
 mouseUp (undefined)  
 updateEvt      WindowPtr; identifies relevant window  
 activateEvt      (same) Note: bit 0 of modifiers indicates  
 activate/deactivate  
 diskEvt      bits 0-15=drive; bits 16-31=error (see [DIBadMount](#))  
 driverEvt      defined by driver  
 app1Evt      application-defined (see [PostEvent](#))  
 app2Evt      (same)  
 app3Evt      (same)  
 app4Evt      (used by [MultiFinder](#) for suspend/resume and mouse-move events)

The modifiers field is a set of bit flags having the following layout. Named constants are defined in EventMgr.h:



Macintosh documentation does not provide a standard name for a pointer to an EventRecord structure. Since passing such a pointer is a common operation, you might wish to use:

```
typedef EventRecord * EventPtr ;
```