PostEvent Place an EventRecord in the event queue

#include <OSEvents.h> **Event Manager**

OSErr	**PostEvent(***eventWhat*, *eventMsg* **);**	
short	*eventWhat* ;	value for EventRecord.what
long	*eventMsg* ;	value for EventRecord.*message*
	returns	0=noErr, 1=evtNotEnb

PostEvent stores an EventRecord into the event queue where it can be read via **GetNextEvent**, **EventAvail** or **WaitNextEvent**.

eventWhat specifies which type of event should be posted. It should be one of the event types listed in Event Types. Typically, this will be the app3Evt event

eventMsg specifies the value to be placed in the *message* field of the EventRecord. It should correspond in type to the meaning of *eventWhat*. For instance, in keyUp and keyDown events, the high word is 0, and the low word is a scan code and character code. For application-defined events, this can be any 32-bit value, such as a handle to a bunch of data.

Returns: a System Error Code. The following are possible:

noErr	(0)	worked without error
evtNotEnb	(1)	*eventWhat* is disabled. See **SetEventMask**

Notes: **PostEvent** creates the EventRecord using the current time, modifiers, and mouse position. If you need to control these values, you may have to modify the queue itself. See **PPostEvent** for a way to alter the EventRecord after it is enqueued and see **GetEvQHdr** for additional information.

It is probably unwise to post window update and activate events (updateEvt or activateEvt) since these are actually generated by the **Event Manager** at the time of the **GetNextEvent** request, and are never actually stored in the queue.

An example of usage might be to post a menu-changing event. For instance, define an app3Evt to be one that causes a window name to be added or removed from your application's Window menu. You could use **PostEvent(**app3Evt, windowID) whenever the user opens or closes a document window. That way, you can handle all window-related menu manipulation as a function of your main event loop. Note: This is not necessarily a *better* way to do it, just an alternative.