PPostEvent Enqueue an event and get its address

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

OSErr	**PPostEvent(***eventWhat*, *eventMsg*, *qElPtr* **)**;	
short	*eventWhat* ;	value for EventRecord.what
long	*eventMsg* ;	value for EventRecord.*message*
EvQElPtr	*******qElPtr* ;	receives address of a queue element pointer
	returns	0=noErr, 1=evtNotEnb

PPostEvent works like **PostEvent** (it stores an entry into the event queue) except that it returns, via its third parameter, the physical address of the stored queue element. This provides access so you can modify the contents of that element.

eventWhat specifies which type of event should be posted. It should be one of the event types listed in **GetNextEvent**.

eventMsg specifies the value to be placed in the *message* field of the EventRecord. It should correspond in type to the meaning of *eventWhat*.

qElPtr is the address of an EvQElPtr. Upon return, it contains the address of a 22-byte evQEl structure. See Notes, below for an example of how to access that record.

Returns: an Error Code. The following are possible:

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

Notes: The less-flexible **PostEvent** function lets you specify values for only two of the five EventRecord fields. By using **PPostEvent**, you can follow up by changing the where, when, and modifiers fields.

For instance, you could use **PPostEvent** to enqueue a command-key shifted mouseDown event with selected coordinates as follows:

```
EvQEl  *myQElPtr;

PPostEvent( mouseDown, 0, &myQElPtr );
myQElPtr->evtQModifiers = cmdKey;
SetPt( &(myQElPtr->evtQWhere), 100,100);
```

See EvQEl for the layout of event queue elements.

It is also possible to build a queue element from scratch and use **Enqueue** to insert it into the event queue. See **GetEvQHdr**.