VBLTask structure

#include <Retrace.h>

typedef struct **VBLTask** {

		Size	Offset	Description
QElemPtr	qLink;	4	0	Address of next item in VBL task list
short	qType;	2	4	Queue type (always vType = 1)
ProcPtr	vblAddr;	4	6	Address of routine to process the task
short	vblCount;	2	10	Number of 1/60th-sec ticks between tasks
short	vblPhase;	2	12	Phase offset for multiple tasks (usually 0)

} **VBLTask**; 14

typedef VBLTask ***VBLQElPtr**;

Notes: The VBLTask structure is used in calls to **VInstall** and **VRemove**.

The vblAddr field is the address of a routine to be called during the vertical retrace blanking interrupt. See **VInstall** for guidelines in writing such a routine.

The vblCount value is the number of ticks to skip between calls to the routine whose address is in vblAddr. Each 16.66 ms (1/60-th of a second), this field is decremented. When it reaches 0, the VBL task routine is called. The task routine is responsible for setting this back to a non-zero value to restart the timing cycle. If left at 0, the task is removed from the queue.

In the event that you add more than one VBL task at the same time, and both have the same vblCount interval, you may want them to to be executed on different ticks (so as not to load up the system too much). In that case, set vblPhase to some non-zero value (but less than vblCount) in the second and subsequent VBLTasks which you install.

A call to **GetVBLQHdr** returns the address of a QHdr which contains a pointer to the first of the VBLTask structures in the vertical retrace task queue. Or you can simply examine the global variable VBLQueue (at 0x0160).