SetOSTrapAddress Install custom code to replace an operating system routine

#include <OSUtils.h> **Operating System Utilities**

void	**SetOSTrapAddress (***trapAddr*, *trapNum,***);**	
long	*trapAddr* ;	address of custom code
short	*trapNum* ;	the trap to intercept. See TrapWords.

SetOSTrapAddress changes an element of the operating system trap dispatch table so that subsequent invocations of that trap will cause execution to go to a specified address. Use this function (and not **SetTrapAddress) if** your application will run in a Mac equipped with a ROM version later than the 64K ROMs (see **About Compatibility).**

trapAddr is the address of some code to handle execution of a operating system function.

trapNum identifies the ROM routine you wish to replace. See TrapWords for a list.

Returns: none

Notes: **SetOSTrapAddress** is part of a new interface to the routine **NSetTrapAddress**. **SetOSTrapAddress** does not require the specification of the trap type as a parameter as **NSetTrapAddress** does. Instead, either **SetToolTrapAddress** or **SetOSTrapAddress** should be called, depending on which trap dispatch table you wish to modify. It is recommended that you use one of these routines in place of **NSetTrapAddress**. See **About Compatibility** for more information on tool traps and OS traps.

SetOSTrapAddress is used mostly by assembly-language programers . It is most often used in device drivers of INIT code, rather an by an application.

> **Note**: Be sure to change all traps back to their original addresses before your application exits!

The trap dispatcher changed between the 64K and 128K ROMs. For more information see **About Compatibility**.