NSetTrapAddress Install custom code to replace a system routine

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

void	**NSetTrapAddress(***trapAddr*, *trapNum, trapType* **);**		
long	*trapAddr* ;	address of custom code	
short	*trapNum* ;	the trap to intercept. See TrapWords.	
short	*trapType* ;	0= OS trap; 1=Toolbox trap	

NSetTrapAddress changes an element of the 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 an Operating System or Toolbox function.

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

trapType differentiates between traps by type, since the 128K ROMs use two separate trap dispatch tables. This must be one of:

OSTrap	**(0)**	Operating System trap
ToolTrap	**(1)**	Toolbox trap

Returns: none

Notes: There is a new interface to this routine, consisting of the calls **SetToolTrapAddress** and **SetOSTrapAddress**. These calls do not require the specification of the trap type as a parameter.

NSetTrapAddress 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**.