

GetTrapAddress Obtain address of code executing system functions

#include <OSUtils.h>

Operating System Utilities

<u>long</u>	GetTrapAddress (<i>trapNum</i>);	
<u>short</u>	<i>trapNum</i> ;	trap number of system routine. See <u>TrapWords</u> .
	returns	address of the trap handler

GetTrapAddress returns the address of a system routine - an element of the trap dispatch table. This can be used to intercept or patch ROM routines. Use **NGetTrapAddress**, **GetToolTrapAddress** or **GetOSTrapAddress** if you know you are running on 128K ROMs or later (see **About Compatibility** for more information).

trapNum identifies the ROM routine whose address you want.

Returns: a 32-bit value; the address of the system routine that corresponds to *trapNum*.

Notes: **GetTrapAddress** can help you to speed up your program. Also, if you know what you're doing, it's possible to intercept OS or Toolbox calls and perform pre- or post-processing, or even replace the function altogether.

Note: If a ROM routine has already been patched, you should NOT do any post-processing. And in any case, you must preserve the registers and maintain the stack. Check the return value of **GetTrapAddress** against the global variable ROMBase to see if the trap has already been redirected to a RAM address (lower in memory than ROM).

If you call a ROM or OS routine directly instead of using a trap or glue, you lose one level of register saves. Be aware that registers A2-A6 and D3-D7 will be preserved, but all others may be modified.

The trap dispatcher changed between the 64K and 128K ROMs. Before using **GetTrapAddress** or **NGetTrapAddress**, see **About Compatibility** to check the ROM version and be aware that the value of *trapNum* must be correct for that version of the ROM.