Traps Patched by INITs Definitions

Note: These Traps were patched by INITs which loaded after TattleTale INIT. They are listed by Trap and show which INIT(s) patched the Trap. This list is useful for helping to identify INIT conflicts. INITs which do not work together have most likely patched the same Trap in an incompatible manner.

NOTE: Data Captured - Date and Time TattleINIT collected data

----

- Trap Patched = Hex Trap number and (Trap Name) if known

◊ Patched By = File which patched trap. This name also appears in the INIT Related Files Loaded listing with more complete information about the file.

$XXXX -> $XXXX : Prior Trap address -> address of patch

Other Trap Definitions

Note: Traps are the mechanism for programmers to access system functions. The three following formats for listing traps are all derived from the Traps file which is part of MPW. This file lists the “official” Mac traps along with the names which they have been assigned. This list is incomplete for a few reasons: some traps were left out by mistake, some were left out because they are subject to change in the future, and some are new.

Unavailable Traps : This lists those “official” traps that for various reasons are not available on your combination of hardware and software. They are listed in the following format:

$Trap Number in Hex = _TrapName

Note: The [address] listed in the following two formats may either be the the original address as shipped by Apple or may represent patches that were made by various INITs, control panels, etc. If you are running in MultiFinder or System 7, they represent patches that apply to the entire system.

Available Traps (Named): This lists “official” traps available to your machine in the following format:

$Trap Number in Hex = _TrapName [$Address of trap in hex]

Available Traps (UnNamed): This lists unofficial but active traps on your machine in the format following. Since some traps point to the same code, one can compare the addresses to determine which are duplicates. Duplicate is really not the correct term as Traps which point to the same code are generally variations of the original trap. For example, _SetFilType shows the same address at $A043, $A243, and $A443. This occurs because, in the case of this type of trap, the $A243 version executes the $A043 command immediately, and the $A443 version executes the command asynchronously. Different kinds of traps (e.g. Memory, File, Device, etc.) have similar variations with different meanings.

$Trap Number in Hex [$Address of trap in hex]