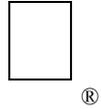


# Apple II Technical Notes



---

Developer Technical Support

## Apple IIGS #101: Patching the Toolbox

Written by:  
1991

Dave Lyons May

This Technical Note presents guidelines on when and how to patch Apple IIGS Toolbox functions.

---

### Introduction

There is normally no need to patch the toolbox; avoid patching whenever you can. If you must patch a toolbox function, be sure to have a good understanding of the call you're patching and how it interacts with the whole system.

No toolbox patch is risk-free. Future version of the toolboxes could change in ways that make your patch less useful. (For example, if you patched NewControl to have some global effect on controls being created, your patch became less useful when NewControl2 was introduced in System Software 5.0.)

For better compatibility, patch with care! If any parameters passed are outside the range that was allowed when you wrote your patch, just pass the call straight through; the new toolbox probably knows something your patch doesn't.

## Patching the Toolbox From an Application

An application can easily patch a function for the duration of that application.

After starting up the tools, construct a Function Pointer Table (FPT) the same size as the existing FPT (call `GetTSPtr` and examine the first word of the table; multiply it by four to get the size of the FPT in bytes). The first longword of your FPT is the number of functions in the tool set; do not hard-code this value! Get it from the existing FPT on the fly. Fill the rest of your FPT with zeroes, except for the functions you want to patch. You must always patch the `BootInit` function (the first function) to return no error. Remember that the function pointer values are one less than the addresses of your replacement functions.

On exit, when you call `TLShutDown` your patch will be automatically removed. (If you're using `ShutDownTools`, you should call `MMShutDown` and `TLShutDown` after you call `ShutDownTools`.)

Note: In the description of `SetTSPtr` on page 24-19 of Apple IIGS Toolbox Reference, Volume 2, there are several references to the TPT. Keep in mind that the TPT is the Toolset Pointer Table, not the Function Table Pointer you pass to `SetTSPtr`. While `SetTSPtr` copies the TPT to RAM if necessary, it does not make a copy of the FPT. After you call `SetTSPtr`, the FPT you passed is being used, and any zero values in your table were filled in.

## **Patching the Toolbox From a Desk Accessory or Setup File**

A permanent initialization file or Desk Accessory can patch toolbox functions at boot time by using constructing an FPT for `SetTSPtr`, as described for an application, but there is an extra step to make the patch “stick.”

Call `LoadOneTool` and then `SetTSPtr`; then call `SetDefaultTPT` (see *Apple IIGS Toolbox Reference* Volume 3, page 51-16).

It is not safe to call `SetDefaultTPT` while an application is running (temporary application patches would be made permanent, and later the application would go away). Since there are desk accessories that install other desk accessories while applications are running, desk accessory that wants to install a tool patch should make the class-one GS/OS `GetName` call; if the null string is returned, no application is executing yet, so it is safe to make the patch. (Otherwise the desk accessory should ask the user to put the desk accessory file in the `System:Desk.Accs` folder and restart the system.)

### **Avoid Tail Patching**

The best kind of patch is a pre-patch or head patch: it does some extra work and then jumps to the original function (as found in the FPT before applying the patch). Make sure the A, X, and Y registers contain the same values when you jump to the original function as they did when the patch got control.

A “tail patch” which calls the original function and then regains control is much more of a compatibility risk, because there are several instances where System Software patches examine return addresses to fix problems

in large toolbox calls which call small ones (by patching the small one to realize it's being called from the big one, many K of RAM remain available to your application).

If you tail patch a function which the system already patched, you may prevent the toolbox from working correctly.

## **Patching the Tool Dispatcher**

If you need to patch a large number of functions, especially for a general purpose utility like a debugger, it may make more sense to patch the tool dispatcher vectors instead of patching individual functions. See Apple IIGS Technical Note #87, Patching the Tool Dispatcher.

## **Further Reference**

---

- *Apple IIGS Toolbox Reference*
- Apple IIGS Technical Note #87, Patching the Tool Dispatcher