

Adventures in Programming: Logout

It was at a party one Saturday night when a friend suggested a program which, upon shutdown, will say how long you've been using your Mac. I had previously done similar programs, including one which logged to disk the time and date of startup. I had planned to eventually add the shutdown time and produce an interesting listing of, to excuse the expression, MacUse.

There is a program floating around the domain called something like "Elapsed Time". This is a program which will display how long it had been since startup. This is no amazing feat because the information can be obtained from the global variable **Ticks** (long word located at \$16A), as revealed on Inside Macintosh page III-231.

The difficult part in displaying the shutdown time would be the actual displaying part. Whenever the Macintosh is correctly shutdown via the **ShutDownPower** command (page V-587), the System Error Handler is called (page II-362) with the value \$2A which, as any avid hacker would realise, is the number 42. [By the way, Douglas Adams denies completely anything to do with this coincidence. Truly.]

The requirement to add a line of text (revealing the time since startup) into the shutdown dialog box (that is, the "You may now safely turn off..." dialog) was indeed a challenge. Fortunately, I had already done much of the work necessary in previously programming (see *DDH, the Definitive DSAT Handler*, if it ever gets finished). A short precis follows.

Whenever the System Error Handler is called (Inside Macintosh, volume II, chapter 12) is called, the dialog to be displayed is retrieved from a DSAT resource kept in memory. The dialogs are by no means in the standard DLOG format. Instead, a special format is used to keep space to a minimum and to keep functionality low enough so that the box can still be drawn in case of major Macintosh heart failure. It's very confusing, but actually quite neat. (Please excuse this somewhat American expression.)

The dialog box may be of any size, can contain up to two strings of text, one icon, two buttons and can include a specialised routine such as the one which displays the error ID when the Mac bombs. The error handler is very useful even without errors. It is used for the disk swap dialog, the "Welcome" dialog, Macsbug and HD20 messages and various assorted bombs. In fact, a different dialog box may be displayed for every different error.

The needs of Logout could therefore be approached in two different ways, either by the inclusion of text or a procedure. I chose the first because it is easier and also because the thought of the second method had not crossed my mind until writing this story.

Nicely enough, there are additional routines supplied with the **Shutdown Manager** (page V-585) which allow routines to be installed into memory which are automatically executed when the shutdown command is called. Therefore, I could install a shutdown routine which could create the required text string and modify the DSAT.

When creating a shutdown routine, a pure code resource must be used. This is identical to an INIT resource. It should be loaded into the system heap, and locked into place. The easiest way to perform this task is to create a resource with the shutdown code (but not named as INIT or CODE resources) which are loaded into the heap and

then detached. Loading into the heap can be accomplished by selecting the **System Heap** bit in ResEdit's Get Info dialog box, or by setting the System Zone to be the Current Zone before loading. (Don't forget to set it back to whatever it was before!)

Another hint... The Shutdown Manager routines are reasonably new and may not be available in every programming language. They are especially hard to access because they use **routine selectors**. This is a device used by Apple to cut back on the number of trap numbers assigned to routines. After all, there are a limited number of them available and the Macintosh may survive forever! The four shutdown routines all use the same trap number, but are selected by pushing the respective routine number onto the stack. In Lightspeed Pascal (if you have not yet got the new library interfaces) it can be done thus:

procedure ShutDwnInstall (ShutDwnProc : ProcPtr; flags : integer);
inline \$3F3C, \$0003, \$A895;

This also comes in handy when your Macintosh bombs. By going into the debugger by pressing interrupt and typing the following, the Macintosh will correctly restart:

```
SM 0 3F3C 0002 A895      { or 0001 instead of 0002 will shutdown }  
G 0
```

Enough of that. Back to the program. The next task is to modify the DSAT. The DSAT is loaded into memory during startup. Firstly DSAT 0 is used for startup messages and then DSAT 2 is loaded for normal operation (*not* INIT 2 as incorrectly stated in Inside Macintosh on II-359). The global variable **DSAlertTab** (pointer at location \$2BA) points to the resource in memory.

The task of adding an additional text line to the dialog, without modifying the resource in the system file (modifications to the system file should never be done if at all avoidable), would require adding an additional entry into the table, adding a reference to the entry and modifying the entry counter. The DSAT needs to expand in memory, therefore a new and larger copy must be created.

But how big is the DSAT in memory? The pointer is not a resource pointer and will not give up its secret with a call to **GetPtrSize**. And the exact size *must* be known to tack an additional entry onto the end. The shutdown routine could load the DSAT resource to discover its size, but this must be accomplished before the disk volumes have been unmounted. It also requires that the same system file be in use as during startup. Altogether a bit risky. Therefore, a form of parameter passing was required.

The technique used to inform the shutdown routine of the DSAT size was to create a dummy variable, with a value such as \$AAAA which the installation routine in the INIT would seek out. This value would then be replaced with the DSAT size. Knowing this, the shutdown routine could copy the DSAT in memory, add the new entry and the exit quite happily. The Error Handler would automatically display the new string on shutdown.

[This is an extremely difficult process to perfect, as I discovered while writing the routine. At a point at which I was very frustrated, a friend phoned me an invitation to visit. After a grumble and a sigh over the fact that my program was not working, I shut down the system and, believe it or not, the computer actually told me my usage time since startup! It *did* work, but I hadn't realised the fact! Serendipity, Dr Who, serendipity!]

It should be mentioned that the program consisted of two parts at this stage. Firstly an INIT which loaded the shutdown routine into memory and passed up the DSAT size, and secondly the actual shutdown routine which was only activated a few milliseconds before shutdown.

At this point I was rather pleased with my efforts, but one fact still nagged at the back of my mind. On the Macintosh II, no shutdown dialog appears. In fact, the computer simply turns off. So how were poor (rich?) Macintosh II users to use my lovingly created program? Could I simply desert them and gift only those Mac 512/Plus/SE owners? Hell, no! I'm a hacker!

Therefore, it was necessary to display some form of dialog box displaying the suitable information just previous to shutdown. A time delay would be required to provide enough time to read the message before shutting down. A simple 'click here' would intimidate people who simply want the system to turn off.

This dialog was soon programmed and turned into a second shutdown routine, to be loaded in place of the first whenever the INIT detects a Macintosh II. The only problem was that my development language at the time did not handle the **SysEnviron** call to say whether it was a Macintosh II. Therefore, the procedure in Tech Note #119 to determine the presence of colour QuickDraw was utilised. [Additional note: QuickDraw hates itself during the

shutdown phase, much like the INIT phase. The standard QuickDraw variable **gray** did not work as desired when displaying a shutdown timer bar.]

This routine worked so nicely that I decided to add the option of having it display on restart, in addition to shutdown. I turned the INIT into a cdev to provide the user with the choice of restart notification. Also, the choice of time delay was provided. Power to the User!

The time delay selection in the cdev was accomplished via the use of the List Manager, which was forced into believing all selections were made while holding down the command key. It works quite well.

The configuration information is stored in a resource which is examined by the INIT upon startup. The relevant routines are loaded into memory and configured for shutdown and restart, if requested. In the case of the Macintosh II, the same routine is called for both cases.

Only one further problem existed (ask Mr Murphy!). After shutting down on a pre-Macintosh II, the dialog box provides the option to restart. By clicking on the Restart button the Macintosh is, indeed, restarted. But the restart routine is also called! This is completely unwanted because the shutdown dialog box already displayed the usage time and it should not be displayed again.

This difficulty was circumvented by passing the address of the restart routine to the shutdown routine. Upon shutdown, the routine would remove the restart routine from the Shutdown Manager. [At least that's what it eventually did. I ran into some difficulties until I discovered that I was passing the wrong procedure pointer to the routine, and it was actually attempting to remove itself from memory!]

Additional hassles not described in this story did, of course, arise. One small hint to pass on, however, is that the control panel appears to dispose of cdev controls before giving the cdev routine the **closeDev** message (page V-333). Therefore, do not rely on the value of controls to save configurations upon closing.

So, I hope you learned something from the fair tale. And I hope you enjoy *Logout*. Oh, and how about sending me a postcard too?

— John Rotenstein

Identification:

Adventures In Programming #2: Logout

Date: 10 December 1988

Copyright 1988 by John Rotenstein

All rights reserved

No printed reproduction allowed without permission of the Author

John Rotenstein

PO Box 165

Double Bay NSW 2028

AUSTRALIA.