

Edit the crontab's daily, weekly and monthly tasks...

One of the “potential problems” with the Macintosh operating system is the use of crontabs to perform scheduled maintenance of the machine. Crontabs are nothing more than a schedule that tells the Mac to do certain things at certain times. The Unix subsystem in Mac OS X was originally written for machines that were never shut off. Many things those old UNIX boxes were set up to do still exist in OS X. As a result, there are many maintenance tasks that are scheduled to run between 3 and 5 a.m. and on weekends in all versions of OS X through Panther. If the computer is turned off or in sleep mode these tasks will not take place. These items we are concerned with here are the scheduled “Daily,” “Weekly” and “Monthly” cleanup tasks. Each of these tasks deletes unnecessary cache, log and system database files that the operating system generates and accumulates. Allowing these files to accumulate will have a negative affect on the performance of the Mac.

For laptops, I strongly recommend the use of a freeware named MacJanitor written by Brian Hill and available at http://personalpages.tds.net/~brian_hill/macjanitor.html. Simply download this utility and drag it to the Applications folder. Add it to the Dock and then run the application, selecting “All Tasks” from the options in the interface, every couple of weeks and you should have few problems caused by the buildup of log and database files.

For your desktop computers there is an easier solution. Simply reprogram the crontab file so that the tasks will run when you know the computer will be on and awake. This requires the use of the Terminal application, which is not nearly as scary a place as many believe. With that said, make sure you do not make any typos and double-check each command before pressing the return key.

Launch Terminal. It is in the Applications/Utilities folder. The crontab file is an invisible file in the private/etc folder, which is also invisible. But that is not a problem since we are using Terminal.

When Terminal launches you should see something similar to this.

```
Last login: Fri Feb 11 08:12:53 on console
Welcome to Darwin!
[Your-Name's-Computer:~] your_name%
```

The % sign is called “the prompt.” Without including the quotation marks, at the prompt type “cd /private/etc” and press the return key. There should be one space between cd and the /. This Unix command tells the computer to change directories (cd) to the private/etc folder. If you want to make sure you are now in the correct place type “ls” return. This should produce a list of files that are in the private/etc directory. One of those files should be named “crontab.”

Typing “clear” return will clear the Terminal window and leave you with a fresh prompt for more commands.

At this point it would probably be a good idea to make a copy of the crontab file. Type “sudo cp crontab crontab.bak” return. You will be prompted for a password. This can be any administrative password on the machine. You will not see any letters being typed as you enter the password. If you have no password leave the field blank and press return. The “sudo” command tells the computer you are a “super user” and have the power of God as far as the computer is concerned. cp crontab crontab.bak means copy (cp) the file crontab and name the copy crontab.bak. If you enter “ls” return after copying the crontab file you should now see the crontab.bak file in the list of files.

We are now, finally, ready to edit the crontab file. For something like this I prefer the text editor pico, which is a program included in the Unix files for use in Terminal.

At the prompt enter “sudo pico crontab” return. This tells the computer to use pico to open the crontab file, and yes, I am still God of the Computer. Something similar to this should open.

```
89# /etc/crontab
SHELL=/bin/sh
PATH=/etc:/bin:/sbin:/usr/bin:/usr/sbin
HOME=/var/log
```

```
#
# Run daily/weekly/monthly jobs.
15 3 * * * root periodic daily
30 4 * * 6 root periodic weekly
30 5 1 * * root periodic monthly
```

The lines of code we will edit are:

```
15 3 * * * root periodic daily
30 4 * * 6 root periodic weekly
30 5 1 * * root periodic monthly
```

There are five columns for numbers in this code. Some of the columns are filled with an *. From left to right, the columns are for minutes, hour, day of the month, month, and day of the week. An * in a field is called a wildcard and means any and all values are entered in the field. Therefore, looking at the code, we see the periodic daily task is scheduled to run at 3:15 a.m. every day. The periodic weekly task will run at 4:30 a.m. on every Saturday (6). The periodic monthly task will run at 5:30 a.m. on the first day of every month.

We need to change all that.

Using the arrow keys, move the cursor to the first space after the 3 in the periodic daily line and press delete. This should remove the 3. Using a 24-hour clock (i.e. military time) enter an hour of the day you know your computer will be on. For example, 9.

Arrow key to the space right after the 4 in the periodic weekly line and change this to something at least an hour later than the periodic daily time. For example, 10. Key over to the space after the 6 in this same line, delete the 6 and replace it with something like Wednesday, which is a 3. Monday is 1, Tuesday is 2, etc.

Arrow key to the space right after the 5 in the periodic monthly line and change this value to something at least an hour later than the periodic weekly task. For example, 11. I looked at a calendar and noticed that if I leave the 1 in the date field for the monthly task there will be three times during a year when the task may not happen because the first of the month falls on a weekend. I can live with that.

Now these lines of code should look like this:

```
15 9 * * * root periodic daily
30 10 * * 3 root periodic weekly
30 11 1 * * root periodic monthly
```

Press the control and o keys to “write out” the changes. This appears at the bottom of the Terminal window.

```
File Name to write : crontab
^G Get Help      ^T To Files
^C Cancel
```

Press the return key to save the file as crontab. You should then see something near the bottom of the window that says something like “wrote 13 lines.” (Your number may vary depending on other tasks you may have scheduled your computer to do by using the System Preferences or other applications.)

Press control and x to exit pico.

Quit the Terminal application. The crontabs should now run during times your computer is on and awake.

Files stuck in the trash...

Ever have a file in the trash that the computer will not let you delete, even though you know there is no reason for you to be denied permission to delete the file? The Terminal can ride to the rescue.

Click on the trash can. This should open the trash so you can see the files in it. Launch Terminal. At the prompt type `rm -drf` and a space, then drag the file from the trash can window into the Terminal window. This drag will enter the path to the file in the Terminal. The text in the Terminal should look something like this.

```
% rm -dr /Users/your_name/.Trash/that-pesky-file-you-want-to-delete
```

`rm` is the remove command. `-dr` are special instructions (arguments) to the `rm` command.

`-d` tells `rm` to remove directories (folders) as well as the files that are in them. If you have a folder in the trash you will need to include `-d`.

`-r` tells `rm` to remove the file hierarchy in each file.

Other arguments you might consider are `-f` and `-P`.

`-f` tells `rm` to remove files without prompting for confirmation, regardless of the files permissions. Using this one removes any second chance of not deleting a file. Use it only if you are sure the files need to be deleted without a second thought.

`-P` is another handy argument to add to a `rm` command if you have security concerns. `-P` will cause the files to be over written before deleting them. Files are overwritten three times, first with the byte pattern `0xff`, then `0x00`, and then `0xff` again, before they are deleted.

All the arguments may be added in a continuous string. For example, `rm -drfP` will remove a file and perform all four arguments.

I have encountered some files, especially if they have been downloaded from a server, that may require the use of `sudo` at the prompt with the `rm` command before the files will be deleted from the trash.

Enable Safari debug menu...

In it's early days, Safari included a Debug menu. The Debug menu does add some interesting features to Safari so you may find this to be a useful hack. Quit Safari. Launch Terminal. Do not include the quotation marks and cut and paste this into the Terminal.

```
"defaults write com.apple.Safari IncludeDebugMenu 1"
```

If you ever wish to disable it again, just repeat the command with a "0" instead of a "1".]

Return arrows at both ends of Finder windows...

This one comes from Chat:

Hello Mac Friends..

I never liked the scroll bars in OS-X. But if you paste this command in a terminal window:

defaults write "Apple Global Domain" AppleScrollBarVariant DoubleBoth

and log out and log back in, you'll get double arrows on both ends of your scroll bars. Took two seconds to do.