Script Files, *Part I, by Ted Roche*

Are you tired of calling up a shell and typing the same old set of commands (ASSIGN FONTS: SHANG-HAI:FONTS) each time you start an application? Wouldn't it be easier if you could just click an icon to do those commands? Well, you can!

Script files are the secret. A script file (also known as a "batch" or "command" file) is an ASCII text file of AmigaDOS commands, read and performed by the EXE-CUTE command. Script files can be handy for mouse-keteers, those with DOS-phobia, fumble-fingered typists, or anyone who dreads typing a set of AmigaDOS commands more than once.

Creating a script file is easy. I suggest you start by writing, editing and testing the file in RAM: or RAD: to get the best speed out of the write-test-edit cycle (it may not come out just right the first time.) You can create a short script file by typing:

COPY * to filename

from a CLI or Shell. (NOTE: ARP users must SET BCPL TRUE for this command to work properly.) Type the lines of the script file, checking each before pressing Return - once Return is pressed, you cannot go back and edit the file without invoking a text file editor. Press Control and / to complete entry. (You can also use a text editor, like ED, to create and edit the files.) Let's start with a simple example. From the Shell type:

COPY * TO RAM:SCRIPT1
ECHO "Hello, World!"
(now hold down Ctrl and press backslash)

The Shell prompt should return. To run the file, type:

EXECUTE RAM:SCRIPT1

The Shell should reply "Hello, World!" Great! You're a script file programmer!

But typing EXECUTE <filename> each time can be a bit tiring. Wouldn't it be nice if you could just type "SCRIPT1", press Return, and your program runs, just like those programs written in "real" computer languages? If you own AmigaDOS 1.3, you can! Type:

PROTECT RAM:SCRIPT1 +S
RAM:SCRIPT1

Once again you see "Hello, World!" The PROTECT command allows you to set a "protection bit," a characteristic stored with files; "+S" tells AmigaDOS that the file is a script file, and to use the EXECUTE command to run it.

Now let's move on to the big times - the Workbench environment and its icons. If your script file doesn't require parameters to tell it how to work (for example, having to specify DF0: or DF1:, which file to work on, etc.) it is a great candidate for "iconizing" - setting it up so that it can be run from the Workbench by clicking on its icon. There are three basic steps: find or make an icon of the correct type, attach it to your file, then tell the icon it's to run a script file. That's it! Now for the details:

An easy way to create a text file with an icon is to use the Notepad from Workbench's Utilities drawer. Start up the Notepad and type in the following lines:

echo "The NOLINE option in echo lets you combine"
echo "your script text with AmigaDOS output; for
    example:"
echo "Your current directory is " NOLINE
cd
echo "Today is " NOLINE
date
echo "Your FONTS: directory is assigned to " NOLINE
assign fonts: exists
wait 10 sec

Now save the file as "RAM:Demo" and quit the Notepad. Double-click the RAM: icon on the Workbench, and you will see an icon named Demo. Click once on the file and then select INFO from Workbench's menu. You will see that this is a Project icon, which means that the file that it's associated with needs another program to run. This program that it needs is identified in the "DEFAULT TOOL" box. Click in the DEFAULT TOOL

box, erase the entry there (Right Amiga-X) and type in C:ICONX. IconX is a program provided by Commodore in the c: directory of your Workbench disk. IconX, like the EXECUTE command, will run a script file, but will run it from a Workbench icon rather than a CLI. Click on the SAVE gadget to save the changes you have made. Now double-click on the Demo icon and enjoy the show!

If you want, you can use TOOL TYPES to change the window' size and name. Again use Workbench's INFO option. Click the DEL gadget twice to delete the tool types left over from using the Notepad. Click on the ADD gadget and in the TOOL TYPES box type in the specifications for the window you want, in the format "WINDOW=Handler:x/y/w/h/ name" where "Handler" is normally CON:, "x" and "y" are the coordinates of the upper left corner of the window on the Workbench's 640 x 200 screen, "w" and "h" are the width and height of the window expressed in pixels, and, finally, "name" is the name you wish to appear in the title bar. For example, type:

WINDOW=CON:160/10/320/100/Script Demo

into the TOOL TYPES window. Activating the icon will create a window that is one quarter of the way from the left edge of the screen (160/640 = 1/4), 10 pixels down from the top, occupies half the screen's width and height, and has "Script Demo" appearing in the title bar.

Now that you've seen the basics of script files, try some of your own. I find "iconized" script files handy for switching between different FONTS: directories, backing up multiple files to floppies, configuring the Amiga's environment for a particular program, and many other tasks that had previously forced me to drag out a Shell and remember the correct AmigaDOS syntax.

Pointers:
•If complex "nesting" of conditions or routines is used, indent subroutines for ease of readability and troubleshooting.
•Comment lines (starting with a semicolon) will also help with troubleshooting and for explaining your code to others.
•Use Ctrl-C or Ctrl-D to stop script file execution if it's not going right.
•Make sure that you rename the icon file to file-name.info (DON'T FORGET .INFO!!!) Forgetting the ".info" is a simple mistake which will destroy your work by overwriting your file with its icon.

Ted Roche is co-owner of Computer Resource, *a New Hampshire-based consulting firm specializing in training on Amiga software and hardware applications. Copyright © 1989 by Ted Roche. All Rights Reserved.*