Drag and Drop

Two methods of drag and drop access have been added to the nShell. The first, using text files, allows the user to quickly start and run scripts within the nShell application. The second method, using a separate engine, allows scripts to be converted to stand alone drag and drop applications. This second method allows files, folders and disks to be dropped on scripts and accessed as standard script parameters.

Either kind of script may also be run by typing its name on the command line. The script would then be run in the existing window.

System Variables

On startup, both kinds of scripts search out the nShell application and set their paths based upon that directory. These paths may be modified by the script. The initial values are:

PWD = "script directory"
HOME = "nShell application directory"
PATH = ":/nShell application directory:bin"
TMP = "nShell application directory:tmp"

TEXT scripts

You may write TEXT scripts using your any text editor, including the TeachText or SimpleText editor that came with your Macintosh. The easiest way to run a TEXT script is to drop it on the nShell application. A new shell window will be opened, and the script will be run. When the script completes, a prompt is provided to the user.

To make the script clickable, change the file's creator to 'NSHA':

chattr my_script -c NSHA

The Droplet Engine

The advantage of using a drag and drop engine, is that it allows files, folders and disks to be dropped on scripts and accessed as standard script parameters.

Droplets terminate automatically when their scripts complete.

Writing a Droplet

Make a copy of the "nShell™ droplet" engine. You can use the finder "Duplicate"

function, or the nShell "cp" command. Give this copy of the drag and drop template a unique name.

Method 1 - Direct Editing

Drag your new icon onto a BBEdit icon, and type in your script. Close the file and you're done.

NOTE: If you use "Save As" within BBEdit, a new text file will be created, and the executable portion of the droplet will not be copied.

Method 2 - Appending a file

If you have an existing script file which you would like to make into a drag and drop application, you can 'cat' the file onto a copy of the "nShell™ droplet":

cat my_script >> template_copy

and you're done.

NOTE: Never use the ">" operator, as this will delete and replace the target with a new TEXT file.

Parameters

The normal script parameters $#, $0...$n will be set up with any items dropped onto the application. Specifically:

$# = The number of parameters (1 = script only, 2 = script + 1 param, etc.)
$0 = The name of the script (not a full pathname)
$1 = The full pathname of the first dropped item
$2 = The full pathname of the second dropped item
...

The parameters may be files, folders, or disks but are always represented as pathnames. Remember to use quotes around these variables, as in

chattr "$1" -c 'NSHA'

In nShell-Pro, a while-shift loop may be used to process a large number of dropped files:

```
#
# Set all text files to Creator = 'NSHA'
#
echo "working..."
```

```
while shift do

ls "$0" -l | read crea type extra

if .eq. $type TEXT then echo ' ' ; chattr "$0" -c NSHA endif
done
```

Limitations: The total length of all dropped pathnames may not exceed 10k characters. Beyond this input is ignored. Only those files dropped on the application as it is opened are set as parameters. Anything dropped on the script after it is running is ignored (the finder is given an errAEEventNotHandled).

Memory

The default memory size for droplets is 250k. That should be enough for quite a lot of processing. If you think you are going to need it, bump the memory. The nShell and the droplet engine start complaining about low memory when 80k is left.

Hints

Hint #1

If you're not sure what's going on with a drag an drop script, add

```
env
echo ' '
echo 'Press <Return> to continue...'
echo ' '
read foo
```

as the first lines. That'll tell you what your dropped files ended up looking like.

And if you don't want to fall off the end of the script right away, use 'delay 5' or 'read foo' or something as the last line.

Hint #2

To run a TEXT script in a new window, you could double-click it in the Finder or type:

```
odoc my_script
```

To run a droplet as a stand-alone application, you could double-click it in the Finder or launch it:

```
launch my_droplet
```

Or use "odoc" to send it a parameter:

odoc my_parameter my_droplet

Hint #3

Any script can run any other script. So create a drag and drop application containing the line

"$1"

and you have a program that will run any nShell script dropped on it and then terminate.

In nShell-Pro, a while-shift loop could be used to execute all dropped scripts:

```
while shift do

ls "$0" -l | read crea type extra

if .eq. $crea NSHA then


if .eq. $type TEXT then



"$0"


endif

endif
done
```