Mauncher

A System 7 Mounting and Launch Utility

Version 1.0 beta5

Software and Documentation
By Henry C. Hsu
Academic Computing
Cornell University Medical College
New York City, New York 10021

Internet:
HENRY_HSU@QMCUMC.MAIL.CORNELL. EDU

WHAT MAUNCHER IS

Mauncher is a mounting and launch utility which executes simple text
instructions.

Mauncher will mount AppleShare Volumes and has the ability to launch
applications
with or without documents. Mauncher was designed to run under System 7 and
uses
Apple Events. It has not been tested on earlier systems.

Mauncher saves the complete information for the mounting of AppleShare
volumes with user names and passwords intact. It saves the information in a
scrambled format to maintain security.

Mauncher contains a text editor so it can edit documents. Its default mode is to auto execute any Mauncher file that is opened. To open a Mauncher document for editing,
hold the CONTROL key down while you are opening it. This is required whenever a
Mauncher document is opened from dialogs or the finder.

MauncherPad is the same application with the editing functions disabled. It is an execute-only version.

At present, Mauncher is being used at the Cornell University Medical College in the management of network server volume access. The student workstations at the College have had Choosers removed

from the System configuration. This was done to reduce the chances of damage or alteration of network accessible locations. Mauncher allows control of access for individual applications and situations.

An actual example of Mauncher use is the management of Knowledge Server. Knowledge Server involves the mounting of multiple volumes of Medline to provide comprehensive access to information. The Mauncher script that handles Medline will mount all the necessary volumes, launch Knowledge Server, wait for Knowledge Server to quit, and unmount all the Knowledge Server volumes when it does.

In short, Mauncher allows increased security for the administrator and increased simplicity for the user.

Mauncher is also being used in the development phase to automatically mount volumes with write permissions and to reset a workstation back to an equivalent student set-up. One script document contains development settings and the other contains student user settings for testing.

Mauncher is still in an early stage of development and your feedback and bug reports will be
useful and appreciated.

COMMANDS

The available commands are:
- **'M'** for **M**OUNT
- **'U'** for **U**NMOUNT
- **'L'** for **L**AUNCH
- **'N'** for **N**ANNY
- **'P'** for **P**ASSWORD
- **'W'** for **W**AIT
- **'K'** for **K**ILL
- **'Q'** for **Q**UIT

The LAUNCH command has options designated by the second characters:
- 'L**C**' LAUNCH an application and **C**ONTINUE with mauncher in the background.
- 'L**D**' LAUNCH an application , keep Mauncher alive but **D**ON'T SWITCH.

Keeps Mauncher in foreground, new application in background.

The commands may be either upper or lower case but must be the first character of a line. Each line is terminated by a return.

All parameters are enclosed in double quotes.

MOUNTING

Syntax:

M "AppleShareZone" "AppleShareServer" "AppleShareVolume" "UserName" "UserPassword" "VolumePassword"

The command M is followed by the names of the defined parameters each surrounded by double quotes. If the UserName, UserPassword, and VolumePassword parameters are left empty, the indicated volume will be mounted with guest status if allowed. All parameters must be present or the mounting command will not work. Empty parameters must be present as empty double quotes such as "". The second example below is a guest log on.

Examples:

M "AppleZone" "MacintoshServer" "68040HD" "Quadra" "900" "900"

M "Apple" "IBM" "Taligent" "" "" ""

Notes:

Volumes that are already mounted will be ignored.

UNMOUNTING

Syntax:

U "AppleShareVolume"

The command U is followed by the name of the volume surrounded by double quotes.

Example:

U "PathMAC"

Notes:

Volumes that are not mounted will be ignored.

LAUNCHING

Syntax:

LC "Pathname:To:The:Application" "An:Optional:Document"

The command L with or without an option is followed by full pathnames for the application and an
optional document. The pathnames must be surrounded by double quotes. The default action of a launch command is for Mauncher to quit after the specified application has been launched. Launch and Nanny pathnames have a special character set to represent the System Folder on the host system, regardless of its location (Mauncher will automatically look for it). The System Folder is represented as ∑ƒ and is produced by the option-W and option-F keystrokes. The second example will show a System Folder launch. Another special character,"®" produced by the option-R keystroke, is for the root directory on the startup volume.

Example:

LD "HD:APPS:TEXT:TeachText" "HD:APPS:TEXT:Instructions"

LC "∑ƒ:TeachText"

Notes:

Options must be specified for Mauncher to continue.

NANNY

Syntax:

N "Pathname:To:The:Application" "An:Optional:Document"

The command N is followed by full pathnames for the application and an optional document. It
functions like launch with the feature of waiting for the launched app to quit before continuing.
When the nannied application quits, subsequent instructions are executed.

Example:

N "HD:APPS:TEXT:TeachText" "HD:APPS:TEXT:Instructions"

KILL

Syntax:

K "Application Name"

The command K is followed by the name of the application to terminate.

Example:

N "TeachText"

Notes:

KILL will not continue until it has confirmation from the target application that it has
terminated. A delay has been built in to allow the target application time to clean up
and close files. KILL will ignore non-running applications.

WAIT

Syntax:

W "TimeInTicks"

The command W is followed by the number of ticks to wait.

Example:

W "120"

Ticks are 1/60th second increments. Therefore, 60 ticks is one second. There are times when Mauncher needs to wait for other processes to stabilize before further execution is possible.

PASSWORD

Syntax:

P "YourPassword"

The command P is followed by a user specified password up to 255 characters in length, delimited by
double quotes.

Example:

P "computer"

Notes:

The presence of a password can protect both the editing and the execution of the document. When a password is requested and the user chooses cancel, Mauncher will quit. The password command and

parameter must be the very first line in the Mauncher document if the editing of the document is to be protected. Passwords located at later portions of the script protect the execution of subsequent instructions. Thus multiple passwords can be inserted at later points for execution control.

QUITTING

Syntax:

Q

The command will force a quit at the point of its placement, even if commands follow it.

COMMENTS

Syntax:

//

The convention of the double slash is used at the start of a line for comments.
At this point, any text not corresponding with the instruction set are ignored. However,
as Mauncher develops, text intended as comments but which do not have // at the beginning of the
line may correspond with commands in later versions.

HOW TO USE IT

The mauncher app has an Execute menu option which will allow you to test a script when you write
one. Beware, however, of Q commands as you will exit Mauncher. Q commands should therefore be
placed when the script is being saved or at least commented out when you are testing.

Generally speaking, NANNY should be used instead of LAUNCH whenever possible. This is because a launched application may hold files open on volumes that Mauncher will unmount in a following instruction. If launched applications have no dependencies or actions on volumes that will be mounted or unmounted, then LAUNCH can be used.

SAMPLE SCRIPT WITH COMMENTED EXPLANATIONS

The following is a sample script with all commands utilized. Comments will accompany the instructions to provide further detail.

```
P "PreventEditing"
// This previous and first line protects the document from being edited

// A Sample Script with everything in it
// Henry C. Hsu     March 1992

// Mount the first of two Educational Software Servers with a username and password
M "Education Center" "Courseware Server" "Courseware Volume" "Administrator" "Secret" "Secret"

// Wait 3 seconds for no reason
W "180"

// Mount the second of the Educational Servers with guest status
M "Education Center" "Images Server" "Images Volume" "" "" ""

// Launch Word with a record document to do some note taking
LC "HD:TEXT:WORD:Microsoft Word 5.0" "HD:TEXT:WORD:Development Notes"

// Nanny Resedit for Changes on an application for courses
N "HD:TOOLS:ResEdit" "Courseware Volume:Student Torturer"

// When Resedit quits unmount the Courseware Server with admin status
U "Courseware Volume"

// Remount the Courseware Server with student status
M "Education Center" "Courseware Server" "Courseware Volume" "Student" "Tired" "Tired"

// Kill Word since note taking is finished
K "Microsoft Word 5.0"

// Quit
Q
```

KNOWN BUGS

Mauncher does not like reopening documents that have been auto executed. It also does not handle nested Mauncher documents. That is, it will not support the launching of another Mauncher document from within the execution of another Mauncher document. Only one document can be executed at a time.

When you wish to edit a script, do not CONTROL-double click from the Finder on your document. Mauncher does not handle this cleanly. You should CONTROL-Open the document from within Mauncher.

Mauncher performs **minimal** error checking. Therefore, it is the scriptor's responsibility to make sure syntax and content of parameters are legal. As far as I can tell, Mauncher should ignore most syntax and parameter errors.