

Customizing Mascot

This document will give examples on how to make the following changes to mascot:

- 1) Modifying the Root Menu to add applications
- 2) Have an “accelerator key” associated with each application/window which can either be used to open the application or cause it to come to the foreground if the event it is already opened.
- 3) Automating Mascot so that when a user starts mascot, it will automatically open up to 5 windows with a specific application in each window;
- 4) Change the Key Sequences which activate mascot
- 5) Additional tips

NOTE: I highly recommend making backup copies of files mentioned below, prior to making any changes to them. Also, when Mascot starts up it will look in the HOME directory of the user. If it finds any specific mascot configuration files in the HOME directory it will use those rather than the standard files found in the Mascot central directory (/usr/mvw). This can be useful when you want to do testing and do not wish to impact your users.

STEP 1 - Editing the Root Menu to Add New Applications

The purpose of this section is to have the new applications show up on the Mascot Root Menu screen.

1. The file which will be edited is /usr/mvw/.mvw_usermenu (note that this is a hidden file so that normally running a standard list will not show it. `ls -la` should cause it to list. This is a useful feature in that you can put these configuration files in your user’s HOME directories with little risk that they may inadvertently delete them). Either make a backup copy of this file first or put a copy of it into your HOME directory and edit it there.
2. Below is a portion of that file. Please note that you will want to change the permissions on the file in order to edit and save changes. (`chmod +w filename`)

```
##
##
##      *****
##      The Mascot default USER command file.
##      *****
##
##      Copyright (C) JSB Computer Systems Ltd., 1995.
```

```
##
##
#RootMenu
" Root Menu " f.title
" &Clients -> " f.menu ClientsMenu
" &Shell " f.exec /bin/sh
" Shuffle &Up " f.circle_up
" Shuffle &Down " f.circle_down
" &Refresh " f.refresh
no-label f.separator
" &About " f.about
" S&napshot " f.snapshot
no-label f.separator
" &End Session " f.quit_mwm
##
```

In our example we will ADD a Line item which will start up the WordPerfect Application. Note the change shown below in bold

```
##
##
## *****
## The Mascot default USER command file.
## *****
##
## Copyright (C) JSB Computer Systems Ltd., 1995.
##
##
#RootMenu
" Root Menu " f.title
" &Clients -> " f.menu ClientsMenu
" &WordPerfect " f.exec /jsbshare/wp/wpbin/wp
" &Shell " f.exec /bin/sh
" Shuffle &Up " f.circle_up
" Shuffle &Down " f.circle_down
" &Refresh " f.refresh
no-label f.separator
" &About " f.about
" S&napshot " f.snapshot
no-label f.separator
" &End Session " f.quit_mwm
##
```

An explanation of the entry:

First off, the Quotes around line item define where the Root Menu Box lines will be drawn (they use the longest line). The item within the quotes is what the Root Menu will read and the & in the quotes defines what letter is the quick key in selecting from the menu. The f.exec then distinguishes that the next item is an executable and you can either define the executable in relative or absolute path (I prefer absolute path for those users who may not have the proper PATH variable defined and exported).

Hints: If there are items you wish not to have defined in the menu you can either delete them or simply put a double hash (##) mark in front to act as a comment.

STEP II - Defining Accelerator Keys

Next, you want to define an accelerator key sequence associated with the new application. For this example it associates the Esc-a key sequence with WordPerfect.

1. Again edit the `.mvw_usermenu` file to add an accelerator key callout. The argument `-Kkeynumber` will bind a key defined in the `mvw_termcap` to the application where `keynumber` is in the range of 1 to 31.

```
#RootMenu
" Root Menu          "          f.title
" &Clients  ->      "          f.menu ClientsMenu
" &WordPerfect      "  -K1     f.exec /jsbshare/wp/wpbin/wp
" &Shell            "          f.exec /bin/sh
" Shuffle &Up        "          f.circle_up
" Shuffle &Down      "          f.circle_down
" &Refresh          "          f.refresh
no-label            f.separator
```

2. The trickier part to this is to find and edit the proper place in the `mvw_termcap` file to specify this accelerator key definition. (Remember to change the permission to make this file writeable before trying to edit). Besides some trial-and-error, the best advice I can give is to search the `mvw_termcap` file for your TERM type and add it in there. If the accelerator key seems to be ignored, then search that section for a line specifying `tc=termtype`. The `tc` mnemonic indicates that the termcap definition is supplemented by this additional terminal definition. Sometimes there is a chain of `tc` specifications so I would suggest testing to confirm that the section you are editing is the correct one BEFORE adding a lot of changes.
3. For example for a Wyse60 Terminal Type. The appropriate section is shown below. Note that these accelerator keys are commented out initially.

```
w6|wyse60|wy60|Wyse 60:\
#
# Sample accelerator key descriptions follow:
#
# Mascot accelerator keys,
#   :SM=\E :RM=\E1:NW=\E\t:ML=\E2:LO=\E3:CL=\E4:RW=\E5:MW=\E=\
#   :SW=\E8:MN=\E9:MX=\E0:\
#   :SM1=\0233 :RM1=\0233\1:NW1=\0233\t:ML1=\0233\2:\
#   :LO1=\0233\3:CL1=\0233\4:RW1=\0233\5:MW1=\0233\=\
#   :SW1=\0233\8:MN1=\0233\9:MX1=\0233\0:\
#
# Application accelerator keys. In the sample, function keys F1 to F6
# have been used. You may choose your own key combinations.
#   :AK1=\^A@\r:AK2=\^AA\r:AK3=\^AB\r:AK4=\^AC\r:AK5=\^AD\r:AK6=\^AE\r:\
#
# End of accelerator key description
```

4. We'll copy and edit a line to look like (note that we remove the Comment Hash to make this line readable):

```
w6|wyse60|wy60|Wyse 60:\
#
# Sample accelerator key descriptions follow:
#
# Mascot accelerator keys,
#   :SM=\E :RM=\E1:NW=\E\t:ML=\E2:LO=\E3:CL=\E4:RW=\E5:MW=\E=\
```

```

# :SW=\E8:MN=\E9:MX=\E0:\
# :SM1=\0233 :RM1=\0233\1:NW1=\0233\t:ML1=\0233\2:\
# :LO1=\0233\3:CL1=\0233\4:RW1=\0233\5:MW1=\0233\=\:\
# :SW1=\0233\8:MN1=\0233\9:MX1=\0233\0:\
#
# Application accelerator keys. In the sample, function keys F1 to F6
# have been used. You may choose your own key combinations.
# :AK1=^A@\r:AK2=^AA\r:AK3=^AB\r:AK4=^AC\r:AK5=^AD\r:AK6=^AE\r:\
# :AK1=\Ea:\
#
# End of accelerator key description

```

5. Test and confirm that when you open mascot, and see the root menu if you hit esc-a, the WordPerfect application opens automatically.

STEP III - Automating the Startup Procedure

There are two ways in which you can have applications open automatically when starting Mascot.

1. Open Mascot. Start up the application that you have configured to be on the root menu and with the accelerator key. Then hit Esc-1 to open the Root Menu. Select "Snapshot" from the menu. It will look as if nothing has happened. However, if you now look in your HOME directory you will see that you have a file called .mvw_startup. You will then see that after the #STARTUP line there is an uncommented line which specifies your WordPerfect Application, complete with the accelerator key callout.

```

## Mascot run-up.
##
#STARTUP
" WordPerfect " -K1 f.exec /jsbshare/wp/wpbin/wp

```

2. You can hand-edit the .mvw_startup file to call out the line exactly as it is called out in the .mvw_usermenu file.

STEP IV – Changing the Default Mascot Key Sequences

By default, mascot responds to certain key strokes. For example the Escape 1 key sequence causes the root window to open, Escape 2 displays a Mascot help window, Escape Tab shuffles through available mascot windows, etc. Because of conflicts with other applications, some users prefer to use different key sequences to activate mascot's functions. This section will show how to make that change. In addition, we'll show how to eliminate some of Mascot's functionality.

<u>Mnemonic</u>	<u>Key Sequence</u>	<u>Action</u>
SM	<ESC><space>	Window Menu
RM	<ESC> 1	Root Menu
NW	<ESC><tab>	Next Application
ML	<ESC> 2	Key Help
LO	<ESC> 3	Lower
CL	<ESC> 4	Close
RW	<ESC> 5	Restore
MW	<ESC> =	Move
SW	<ESC> 8	Size
MN	<ESC> 9	Minimize

MX

<ESC> 0

Maximize

If we wanted to change the Root Menu, and Switch window mnemonic to use a Control-W instead of the escape key, then we could edit the /usr/mvw/mvw_termcap file and make the following change:

Original file:

```
#
w6|wyse60|wy60|Wyse 60:\
#
# Sample accelerator key descriptions follow:
#
# Mascot accelerator keys,
#   :SM=\E :RM=\E1:NW=\E\t:ML=\E2:LO=\E3:CL=\E4:RW=\E5:MW=\E=:\
#   :SW=\E8:MN=\E9:MX=\E0:\
#   :SM1=\0233 :RM1=\0233\1:NW1=\0233\t:ML1=\0233\2:\
#   :LO1=\0233\3:CL1=\0233\4:RW1=\0233\5:MW1=\0233\=\:\
#   :SW1=\0233\8:MN1=\0233\9:MX1=\0233\0:\
#
```

Edited file:

```
#
w6|wyse60|wy60|Wyse 60:\
#
# Sample accelerator key descriptions follow:
#
# Mascot accelerator keys,
#   :RM=^W1:NW=^W\t:\
#   :SM=\E :RM=\E1:NW=\E\t:ML=\E2:LO=\E3:CL=\E4:RW=\E5:MW=\E=:\
#   :SW=\E8:MN=\E9:MX=\E0:\
#   :SM1=\0233 :RM1=\0233\1:NW1=\0233\t:ML1=\0233\2:\
#   :LO1=\0233\3:CL1=\0233\4:RW1=\0233\5:MW1=\0233\=\:\
#   :SW1=\0233\8:MN1=\0233\9:MX1=\0233\0:\
#
```

Finally, if you only wanted to activate the Root Menu and the Switch Window mode, but inactivate all of the other default mnemonics you could add the NDK (no default key) mnemonic as shown below:

```
#
w6|wyse60|wy60|Wyse 60:\
#
# Sample accelerator key descriptions follow:
#
# Mascot accelerator keys,
#   :RM=^W1:NW=^W\t:NDK:\
#   :SM=\E :RM=\E1:NW=\E\t:ML=\E2:LO=\E3:CL=\E4:RW=\E5:MW=\E=:\
#   :SW=\E8:MN=\E9:MX=\E0:\
#   :SM1=\0233 :RM1=\0233\1:NW1=\0233\t:ML1=\0233\2:\
#   :LO1=\0233\3:CL1=\0233\4:RW1=\0233\5:MW1=\0233\=\:\
#   :SW1=\0233\8:MN1=\0233\9:MX1=\0233\0:\
#
```

Final Pointers/Tips

1. If you plan to have several windows running the same application, that means that you need to specify a unique accelerator key for each window/application. Therefore if you were to have four windows running the same application, you would have four entries in the root menu and in the startup file.

For performance improvement you may want to consider NOT having the same application open in several windows at the same time. Rather than having them open automatically, the user can utilize the accelerator keys to open subsequent windows on an as-needed basis. However, I recognize this as a personal preference issue so I'll let you decide what works best for your customer.

2. After you have the startup files configured as you like, I highly recommend commenting the snapshot item off of the root menu. Otherwise your users may inadvertently select it and create their own .mvw_startup file which may not be what you had in mind.
3. If you wish to make your users close their application before exiting Mascot, you can add a -s0 flag in the menu/startup lines. With that designation if they try and exit mascot with an application still open it will tell them to close their application first.

Example: " WordPerfect " -K1 -s0 f.exec /jsbshare/wp/wpbin/wp

4. For a real lock down, some administrators incorporate the following to that they effectively prevent their users from ever getting to a UNIX shell prompt:
 - a) Remove the UNIX shell applications from the .mvw_usermenu file
 - b) Edit the user's .profile to do the following:

```
exec mascot -m
```

the -m option causes Mascot to terminate when the last application window closes without bringing up the root menu again. The exec statement will then cause the user to log out of UNIX once the Mascot application is closed. In most of these cases, the administrators are also using either accelerator keys or the Esc-Tab key to switch windows but they have blocked out all other of Mascots default accelerator keys (such as Esc-1 for root menu and Esc-space for the Windows menu). They also incorporate the .mvw_startup file to start the applications that the users will be limited to.

5. Starting mascot with a -v option is useful for the administrator in that it shows the Serial Number, number of licenses, number of users running Mascot, and which files mascot is using.

Good luck!