Overview

This document:

- Presents a problem with VMware Fusion (version 1.x – 3.x) running FreeBSD as a guest operating system, where the Command ("Apple") key appears to behave as the Escape key;
- Discusses possible methods available to work around the problem.

Problem

Background

The keyboard shortcut Ctrl-Command is used when keyboard and mouse focuses need to be "ungrabbed" from the virtual machine. There exist other keyboard shortcuts that involve the Command key, which also involve the Command key.

A standard PS/2 or USB keyboard—also emulated by VMware—is required to transmit a keyboard "scan code" as soon as a key is pressed ("down"), then another scan code when the key is released ("up"). This means that as soon as the Command key is pressed as part of a keyboard shortcut, the virtual machine must transmit the "down" scan code for the GUI key (also known as the "Windows" key, to which the Command key is mapped in VMware Fusion) to the guest operating system.

Normally, a "down" event for a modifier key, such as Ctrl of Shift, does not generate any real character sequence into the terminal, and this is also how the GUI key behaves on most guest operating systems, including Windows and Linux (running in text console mode). However, the default keyboard mapping of FreeBSD system console (syscons) treats GUI keys as extra function keys (fkey62 and fkey63 for left/right respectively), which are non-modifier keys and feed the active terminal with character data for function key escape sequences (ESC [J for the left GUI key, or fkey62, for instance).

Symptoms

The first byte in the function key escape sequence—ESC—is interpreted as a literal Escape keystroke by the foreground application running on the terminal. Notably, the FreeBSD installer (sysinstall) does this, and "cancels" whichever dialog window active when the Command is received.

Solution

Initial Installation (sysinstall) – Workaround

When installing FreeBSD for the first time onto a virtual machine, one can work around this problem by temporarily mapping the Command key to a modifier key.

To do this:

1. Select the **VMware Fusion – Preferences…** menu item;
2. Switch to the **Keyboard & Mouse** tab in the preferences dialog window;
3. Click **Key Mappings** to activate the mapping configuration pane;
4. Double-click the Command key item from the list box (which should, by default, indicate that

the virtual machine shortcut is the Windows key);

5. In the **To** section of mapping configuration pop-up, click the Windows key button to deselect it, and click the Alt key button to select it, then press OK.
6. Close the preferences dialog window.

This workaround is **not recommended** as a long-term solution, as the modified mapping is global in VMware and affects all the virtual machines, including Windows guest operating systems where the Windows key is used frequently; after installation of FreeBSD is finished, one should follow steps similar to the above to revert the mapping back from the Alt key to the Windows key.

Normal Usage – Long-term Solution

Once the installation is finished, one can instruct the FreeBSD system console (syscons) not to treat Command keys as function keys that generate ESC character data, by creating a custom syscons keyboard mapping. Follow these steps to do so:

1. Log in as root.
2. Determine which keyboard mapping is in use, by running:

   ```
   # grep ^keymap= /etc/rc.conf
   keymap="us.iso"
   #
   ```

   In this case, the "us.iso" mapping is being used.

   If the grep command produces no output, assume the "us.iso" mapping, which is a reasonable default.
3. Copy the mapping file for the active mapping, found in /usr/share/syscons/keymaps with a ".kbd" extension, under a new, similar name:

   ```
   # cd /usr/share/syscons/keymaps
   # cp us.iso.kbd us.iso.nocmd.kbd
   #
   ```
4. Edit the new keyboard mapping file using a text editor, e.g. Vi or Emacs.
5. Locate two lines that begin with numbers 105 and 106, which are the keyboard scan codes for left/right GUI keys respectively.
6. The next 8 words on these lines (after the number) define how to treat the keystroke under various modifiers. For example, us.iso maps them to fkey62 and fkey63 respectively, i.e. the lines originally look like:

   ```
   105   fkey62 fkey62 fkey62 fkey62 fkey62 fkey62 fkey62 fkey62  O
   106   fkey63 fkey63 fkey63 fkey63 fkey63 fkey63 fkey63 fkey63  O
   ```

   Replace each of them with the word "nop" (no operation), which instructs syscons to ignore the keystrokes. The modified lines will look like:

   ```
   105   nop    nop    nop    nop    nop    nop    nop    nop     O
   106   nop    nop    nop    nop    nop    nop    nop    nop     O
   ```
7. Save the modified file and exit the editor.
8. Edit /etc/rc.conf using a text editor, e.g. Vi or Emacs.
9. Locate the "keymap=" line seen in Step 2, and change its value to the new mapping (omit the

".kbd" extension because the startup script adds it automatically):

```
keymap="us.iso.nocmd"
```

If no line was seen in Step 2 (and "us.iso" was assumed), add a new line at the end of the file instead.

10. Save the modified file and exit the editor.
11. Reflect the new setting by either rebooting the system or running the following command:

```
# /etc/rc.d/syscons start
Configuring syscons: keymap keyrate font8x16 font8x14 font8x8 blanktime
screensaver.
#
```

(At least the word "keymap" should be shown.)

Alternative Workaround – For VMware Fusion 1.x and 2.x

Versions 1.x and 2.x of VMware Fusion mis-implemented the keyboard emulation, which delays transmission of the down scan code for the (mapped) Command key until the key is released, at which point both the down and the up scan codes are sent simultaneously. Furthermore, when the Command key is used as part of a VMware keyboard shortcut, e.g. Ctrl-Command, the VMware does not transmit the scan codes for the Command key at all to the guest operating system.

One can use this fact to work around the Escape key problem, by releasing the other key (e.g. Ctrl) in the shortcut before releasing the Command key.

To enumerate, there are four possible keystroke combinations for Ctrl-Command:

- Ctrl down, Command down, Ctrl up, Command up
- Command down, Ctrl down, Ctrl up, Command up
- Ctrl down, Command down, Command up, Ctrl up
- Command down, Ctrl down, Command up, Ctrl up

Useful out of these four are the first two combinations, which release the Command key before the other (Ctrl) key.

Contact Author

Please direct further questions or comments to the author, Eugene M. Kim, at gene@nttmcl.com.