

## CHANGES FOR 3.0

The **MOTD** application no longer updates the `/usr/adm/wtmp` and `/usr/adm/lastlog` files. This deficiency in **Workspace** has been corrected in 3.0. Thus, there is no longer a *LogoutHook* component. You'll want to:

```
dremove loginwindow LogoutHook
```

if you were previously using the `/usr/local/bin/LogoutHook` component. You should also do

```
dwrite loginwindow LoginHook /LocalApps/MOTD.app/MOTD
```

to change the *LoginHook* from `'LocalApps/MOTD'` to `'/LocalApps/MOTD.app/MOTD'`.

**MOTD** now handles `*.rtfd` files but it no longer assumes that if the first character of a file is `{`, then it is in RTF format. It now only looks at file name extensions (`*.rtf` & `*.rtfd`) to determine if a file contains RTF.

**MOTD** no longer uses the files `mail.snd` and `newmail.snd` on the `/LocalLibrary/Sounds` directory (these `*.snd` files contained the "You have mail" and "You have new mail" voice alerts). Instead, it contains localizable sound files in the `MOTD.app` application bundle itself.

The system defaults have been moved from the `MOTD.h` to the file `Defaults.strings` in the `MOTD.app` application bundle.

## DESCRIPTION

The **MOTD** application (designed to run as a *LoginHook* under `loginwindow` or be *launched*) displays the contents of the file `/etc/motd` (or any file you choose) in a window when a user logs on. By default, the application checks the modification date of the file against the user's last login time (as recorded in `/usr/adm/wtmp`) and does nothing if the file hasn't changed.

The **MOTD** window has three buttons, one to *hold* the window on the screen (it goes away automatically in 25 seconds--the *hold* button displays the count down), another to *quit* (and proceed with the login) and a third to *cancel* the login (only enabled if **MOTD** is run as a *LoginHook*). **MOTD** also implements a *Mail Alert* panel which appears if there is unread mail. The panel automatically goes away (in a couple of seconds) if there isn't a new `/etc/motd` message.

To get an idea of what **MOTD** does (without installing it), as *root* do:

```
#touch /etc/motd
#MOTD
```

When used as a *LoginHook*, the window appears over a blank screen just as the user logs in. **MOTD** supports RTF.

**INSTALLATION**

To install **MOTD** as a *LoginHook*, put it on /LocalApps, put **LogoutHook** on /usr/local/bin and as *root* do:

```
dwrite loginwindow LoginHook /LocalApps/MOTD.app/MOTD
```

alternatively, the file /etc/ttys can be modified to include the -LoginHook arguments to loginwindow.

You may want to consider commenting out the following lines in /usr/adm/monthly that trim the wtmp and lastlog files or else users may see messages they've read over again and the logging information won't be correct:

```
# Trim the lastlog
# cp -p /usr/adm/lastlog /usr/adm/lastlog.old
# cat /dev/null > /usr/adm/lastlog

# Trim wtmp
# cp -p /usr/adm/wtmp /usr/adm/wtmp.old
# cat /dev/null > /usr/adm/wtmp
```

You may want to come up with some less frequent scheme for trimming the wtmp file. A program that can do this, trimwtmp.c, is included and can be added to the /usr/adm/monthly cleanup. See the trimwtmp.c file for details. Finally, restart the window server (login as *EXIT* or reboot) for this to take effect. Update /etc/motd as appropriate.

If you don't want to make **MOTD** a requirement for users, then don't make it a *LoginHook* but instead have interested users add it to their dock and set it launch at startup using the '*Preferences...*' panel of **Workspace**.

## DEFAULTS

**MOTD** has the following defaults that can be changed (if used as a *LoginHook*, they have to be changed in *root's* defaults database):

- File** -- The file to display, defaults to `/etc/motd`.
- Wait** -- The amount of time to display *File*, defaults to 25 seconds.
- MailWait** -- The amount of time to display mail alert, defaults to 2 seconds.
- Hold** -- Whether or not (Yes or No) to come up *held*, No by default. If Yes then the Hold button won't appear.
- OnlyOnce** -- Whether or not (Yes or No) to check file date, Yes by default. If No, displays file on every login.

The defaults can be changed via *root's* defaults database or by editing their values in the `Defaults.strings` file in the `MOTD.app` application bundle.

**MOTD** also uses the *SystemAlert* default of the *System* entry in the defaults database to control voice messages on the mail alerts. This value can be set (by *root* if **MOTD** is a *LoginHook*) via the '*Use Voice Alerts*' checkbox in the **Preferences** application. You may need to force an entry for *SystemAlert* by cycling the '*Use Voice Alerts*' checkbox (turning off and on the check mark) as *root's* defaults database may not contain an entry for *SystemAlert* even though the checkbox is selected.

## FILE FORMAT

The *File* is assumed to be any plain text format acceptable to a *ScrollView*. **MOTD** also recognizes the '`rtf`' and '`rtfd`' file extensions as *RTF* (rich text format) files and displays them as such. If you use *RTF* files with **MOTD**, it is probably best to create separate files, `motd` and `motd.rtf`, and change the *File* default above to use the `motd.rtf` (or `motd.rtfd`) file (*dwrite* the new *File* default as *root* if **MOTD** is a *LoginHook*):

```
dwrite MOTD File /etc/motd.rtf
```

This prevents network logins being presented with the *RTF* structured text.

## DIAGNOSTICS

**MOTD** is designed to run as a *LoginHook* for the `loginwindow` program and thus always tries to return *success* (0), even on error, so that the user can still log in if the program runs into trouble. The '*Cancel Login*' button causes the program to exit with *failure* (1) and abort the login.

## BUGS

**MOTD** determines that a file is in *RTF* format based on its extension and doesn't look at content.

An affirmative database value for *Hold* and *OnlyOnce* can be '*Yes*', *YES*' or '*yes*' but not '*true*'. Anything not starting with Y/y is considered '*No*'.

The *console* device that **MOTD** enters into the `wtmp` file is hardcoded in `MOTD_main.m` since `basename (ttyname (0))` didn't work in the context of a *LoginHook*.

The timeout should probably be a function of the number of lines of text in the file being displayed.

## AUTHORS

Christopher Lane (`lane@sumex-aim.stanford.edu`)  
Symbolic Systems Resources Group  
Knowledge Systems Laboratory  
Stanford University

Izumi Ohzawa (`izumi@violet.berkeley.edu`)  
Group in Neurobiology/School of Optometry  
University of California, Berkeley

Steve Hayman (`sahayman@cs.indiana.edu`)

Computer Science Department Workstation Manager