



**Full credit is given to the above companies including the Operating System (OS) that this PDF file was generated!**

### ***Rocky Enterprise Linux 9.2 Manual Pages on command 'update-motd.5'***

#### ***\$ man update-motd.5***

update-motd(5)                      File Formats Manual                      update-motd(5)

#### **NAME**

update-motd - dynamic MOTD generation

#### **SYNOPSIS**

/etc/update-motd.d/\*

#### **DESCRIPTION**

UNIX/Linux system administrators often communicate important information to console and remote users by maintaining text in the file /etc/motd, which is displayed by the pam\_motd(8) module on interactive shell logins.

Traditionally, this file is static text, typically installed by the distribution and only updated on release upgrades, or overwritten by the local administrator with pertinent information.

Ubuntu introduced the update-motd framework, by which the motd(5) is dynamically assembled from a collection of scripts at login.

Executable scripts in /etc/update-motd.d/\* are executed by pam\_motd(8) as the root user at each login, and this information is concatenated in /run/motd.dynamic. The order of script execution is determined by the run-parts(8) --lsbysinit option (basically alphabetical order, with a few caveats).

On Ubuntu systems, /etc/motd is typically a symbolic link to /run/motd.dynamic.

#### **BEST PRACTICES**

MOTD fragments must be scripts in /etc/update-motd.d, must be executable, and must emit information on standard out.

Scripts should be named NN-xxxxxx where NN is a two digit number indicating their

position in the MOTD, and xxxxxx is an appropriate name for the script.

Scripts must not have filename extensions, per run-parts(8) --lsbysinit instructions.

Packages should add scripts directly into /etc/update-motd.d, rather than symlinks to other scripts, such that administrators can modify or remove these scripts and upgrades will not wipe the local changes. Consider using a simple shell script that simply calls exec on the external utility.

Long running operations (such as network calls) or resource intensive scripts should cache output, and only update that output if it is deemed expired. For instance:

```
/etc/update-motd.d/50-news

#!/bin/sh

out=/run/foo

script="w3m -dump http://news.google.com/"

if [ -f "$out" ]; then

    # Output exists, print it

    echo

    cat "$out"

    # See if it's expired, and background update

    lastrun=$(stat -c %Y "$out") || lastrun=0

    expiration=$(expr $lastrun + 86400)

    if [ $(date +%s) -ge $expiration ]; then

        $script > "$out" &

    fi

else

    # No cache at all, so update in the background

    $script > "$out" &

fi
```

Scripts should emit a blank line before output, and end with a newline character. For instance:

```
/etc/update-motd/05-lsb-release

#!/bin/sh

echo

lsb-release -a
```

/etc/motd, /run/motd.dynamic, /etc/update-motd.d

## SEE ALSO

motd(5), pam\_motd(8), run-parts(8)

## AUTHOR

This manpage and the update-motd framework was written by Dustin Kirkland <kirkland@canonical.com> for Ubuntu systems (but may be used by others). Permission is granted to copy, distribute and/or modify this document under the terms of the GNU General Public License, Version 3 published by the Free Software Foundation.

On Debian systems, the complete text of the GNU General Public License can be found in /usr/share/common-licenses/GPL.

update-motd

13 April 2010

update-motd(5)