

For many years, Windows Notepad only supported text documents containing Windows End of Line (EOL) characters - Carriage Return (CR) & Line Feed (LF). This means that Notepad was unable to correctly display the contents of text files created in Unix, Linux and macOS.

For example, here's a screenshot of Notepad trying to display the contents of a Linux `.bashrc` text file, which only contains Unix LF EOL characters:



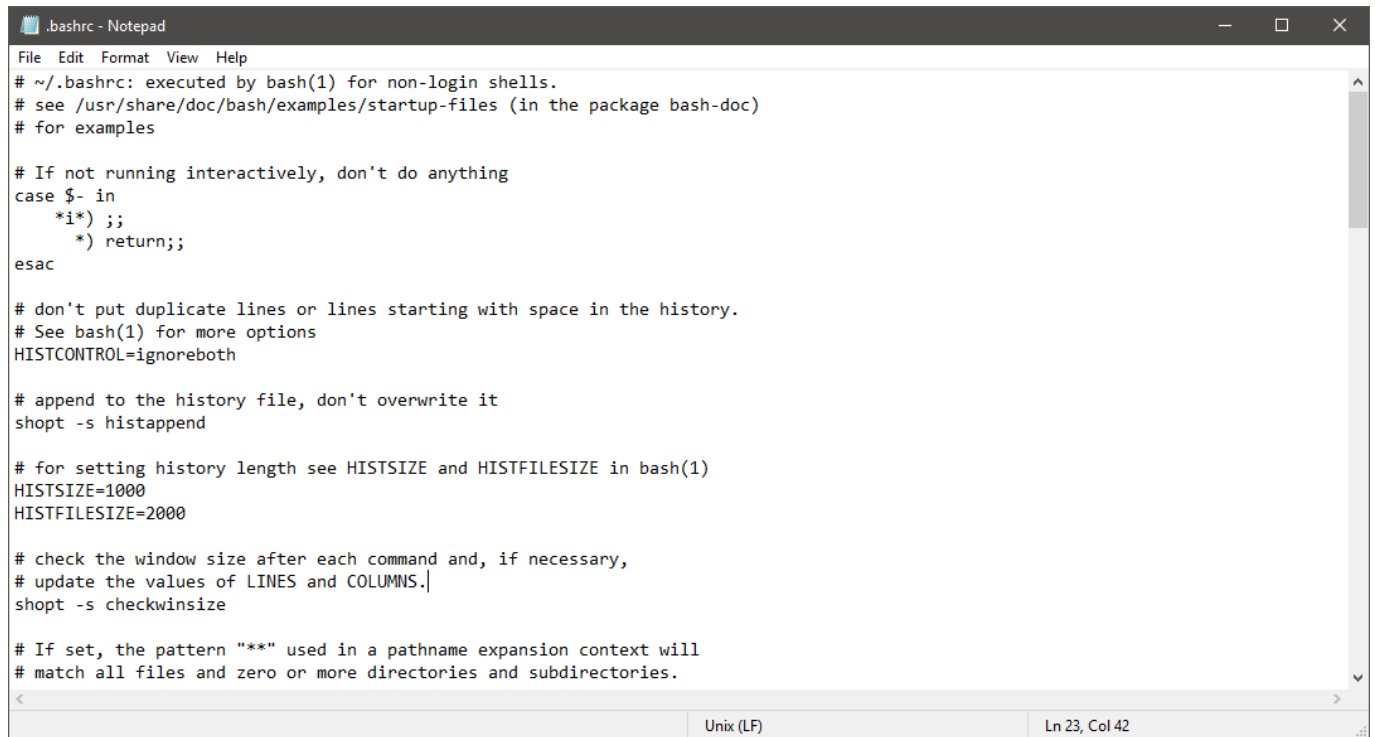
```
# ~/.bashrc: executed by bash(1) for non-login shells. # see /usr/share/doc/bash/examples/startup-files (in the package bash-doc) #
for examples # If not running interactively, don't do anything case $- in
    *) return;; esac # don't put duplicate lines
or lines starting with space in the history. # See bash(1) for more options HISTCONTROL=ignoreboth # append to the history file, don't
overwrite it shopt -s histappend # for setting history length see HISTSIZE and HISTFILESIZE in bash(1) HISTSIZE=1000 HISTFILESIZE=2000 #
check the window size after each command and, if necessary, # update the values of LINES and COLUMNS. shopt -s checkwinsize # If set,
the pattern "***" used in a pathname expansion context will # match all files and zero or more directories and subdirectories. # shopt
-s globstar # make less more friendly for non-text input files, see lesspipe(1) # [ -x /usr/bin/lesspipe ] && eval "$(SHELL=/bin/sh
lesspipe)" # set variable identifying the chroot you work in (used in the prompt below) if [ -z "${debian_chroot:-}" ] && [ -r
/etc/debian_chroot ]; then
    debian_chroot=$(cat /etc/debian_chroot) fi # set a fancy prompt (non-color, unless we know we "want"
color) case "$TERM" in
    xterm-color | *-256color) color_prompt=yes;; esac # uncomment for a colored prompt, if the terminal has the
capability; turned # off by default to not distract the user: the focus in a terminal window # should be on the output of commands,
not on the prompt # force_color_prompt=yes if [ -n "$force_color_prompt" ]; then
    if [ -x /usr/bin/tput ] && tput setaf 1
    >&/dev/null; then
        # We have color support; assume it's compliant with Ecma-48
        # (ISO/IEC-6429). (Lack of such support is
        extremely rare, and such
        # a case would tend to support setf rather than setaf.) color_prompt=yes
    else
        color_prompt=
    fi if [ "$color_prompt" = yes ]; then
        PS1='${debian_chroot:+($debian_chroot)}\[\033[01;32m\]\u@\h\[\033[00m
\]:\[\033[01;34m\]\w\[\033[00m\]\$ '
    else
        PS1='${debian_chroot:+($debian_chroot)}\u@\h:\w\$ '
    fi unset color_prompt
    force_color_prompt # If this is an xterm set the title to user@host:dir case "$TERM" in
    xterm* | rxvt*)
        PS1="\[\e\];${debian_chroot:+($debian_chroot)}\u@\h: \w\a\]$PS1"
        ;;
    *)
        ;;
    esac # enable color support of ls and also add handy aliases if [ -x
/usr/bin/dircolors ]; then
    test -r ~/.dircolors && eval "$(dircolors -b ~/.dircolors)" || eval "$(dircolors -b)"
    alias ls='ls
--color=auto'
    #alias dir='dir --color=auto'
    #alias vdir='vdir --color=auto'
    #alias grep='grep --color=auto'
    #alias
fgrep='fgrep --color=auto'
    #alias egrep='egrep --color=auto' fi # colored GCC warnings and errors # export
GCC_COLORS='error=01;31:warning=01;35:note=01;36:caret=01;32:locus=01:quote=01' # some more ls aliases # alias ll='ls -l' # alias la='ls
-A' # alias l='ls -CF' # Alias definitions. # You may want to put all your additions into a separate file like # ~/.bash_aliases,
instead of adding them here directly. # See /usr/share/doc/bash-doc/examples in the bash-doc package. if [ -f ~/.bash_aliases ]; then
. ~/.bash_aliases fi # enable programmable completion features (you don't need to enable # this, if it's already enabled in
/etc/bash.bashrc and /etc/profile # sources /etc/bash.bashrc). if ! shopt -oq posix; then
    if [ -f /usr/share/bash-completion/bash_completion ]; then
        . /usr/share/bash-completion/bash_completion
    elif [ -f /etc/bash_completion ]; then
        . /etc/bash_completion
    fi fi
```

As you can see, Notepad is incorrectly displaying the file's contents, making the file look garbled. This has been a major annoyance for developers, IT Pros, administrators, and end users throughout the community.

Today, we're excited to announce that we have fixed this issue!

Starting with the current Windows 10 Insider build, Notepad will support Unix/Linux line endings (LF), Macintosh line endings (CR), and Windows Line endings (CRLF) as usual. New files created within Notepad will use Windows line ending (CRLF) by default, but it will now be possible to view, edit, and print existing files, correctly maintaining the file's current line ending format.

Here's a screenshot of the newly updated Notepad displaying the contents of the same Unix/Linux `.bashrc` file we saw earlier:



```

.bashrc - Notepad
File Edit Format View Help
# ~/.bashrc: executed by bash(1) for non-login shells.
# see /usr/share/doc/bash/examples/startup-files (in the package bash-doc)
# for examples

# If not running interactively, don't do anything
case $- in
  *(*) ;;
  *) return;;
esac

# don't put duplicate lines or lines starting with space in the history.
# See bash(1) for more options
HISTCONTROL=ignoreboth

# append to the history file, don't overwrite it
shopt -s histappend

# for setting history length see HISTSIZE and HISTFILESIZE in bash(1)
HISTSIZE=1000
HISTFILESIZE=2000

# check the window size after each command and, if necessary,
# update the values of LINES and COLUMNS.
shopt -s checkwinsize

# If set, the pattern "*" used in a pathname expansion context will
# match all files and zero or more directories and subdirectories.

```

Also note that the status bar indicates the detected EOL format of the currently open file.

As with any change to a long-established tool, there's a chance that this new behavior may not work for your scenarios, or you may prefer to disable this new behavior and return to Notepad's original behavior. To do this, you can change the following registry keys in the following location to tweak how Notepad handles pasting of text, and which EOL character to use when Enter/Return is hit:

[HKEY_CURRENT_USER\Software\Microsoft\Notepad]

Value	Enabled value	Disabled value
fPasteOriginalEOL Default: 0	0 Modifies the EOL character of the text being pasted into Notepad to the EOL character of the currently open document	1 EOL characters in text pasted into Notepad are NOT modified
fWindowsOnlyEOL Default: 0	0 Insert the currently open document's detected EOL character when Return/Enter key is hit	1 Force Windows CRLF line endings when Return/Enter key is hit <i>In this mode, Notepad will behave as it always has done and will not correctly render Linux/Mac text files (see 1st screenshot above)</i>

We hope that you find this change useful and look forward to hearing your feedback.