Full credit is given to the above companies including the OS that this PDF file was generated!

Red Hat Enterprise Linux Release 9.2 Manual Pages on 'more.1p' command

$ man more.1p

MORE(1P) POSIX Programmer's Manual MORE(1P)

PROLOG

This manual page is part of the POSIX Programmer's Manual. The Linux implementation of this interface may differ (consult the corresponding Linux manual page for details of Linux behavior), or the interface may not be implemented on Linux.

NAME

more ? display files on a page-by-page basis

SYNOPSIS

more [-ceisu] [-n number] [-p command] [-t tagstring] [file...]

DESCRIPTION

The more utility shall read files and either write them to the terminal on a page-by-page basis or filter them to standard output. If standard output is not a terminal device, all input files shall be copied to standard output in their entirety, without modification, except as specified for the -s option. If standard output is a terminal device, the files shall be written a number of lines (one screenful) at a time under the control of user commands. See the EXTENDED DESCRIPTION sec? tion.

Certain block-mode terminals do not have all the capabilities necessary to support the complete more definition; they are incapable of accept? ing commands that are not terminated with a <newline>. Implementations that support such terminals shall provide an operating mode to more in

which all commands can be terminated with a <newline> on those termi?
nals. This mode:

* Shall be documented in the system documentation
* Shall, at invocation, inform the user of the terminal deficiency that requires the <newline> usage and provide instructions on how this warning can be suppressed in future invocations
* Shall not be required for implementations supporting only fully ca?pable terminals
* Shall not affect commands already requiring <newline> characters
* Shall not affect users on the capable terminals from using more as described in this volume of POSIX.1?2017

OPTIONS

The more utility shall conform to the Base Definitions volume of POSIX.1?2017, Section 12.2, Utility Syntax Guidelines, except that '+' may be recognized as an option delimiter as well as '-'.

The following options shall be supported:

-c If a screen is to be written that has no lines in common with the current screen, or more is writing its first screen, more shall not scroll the screen, but instead shall redraw each line of the screen in turn, from the top of the screen to the bottom. In addition, if more is writing its first screen, the screen shall be cleared. This option may be silently ignored on devices with insufficient terminal capabilities.

-e Exit immediately after writing the last line of the last file in the argument list; see the EXTENDED DESCRIPTION section.

-i Perform pattern matching in searches without regard to case; see the Base Definitions volume of POSIX.1?2017, Section 9.2, Regular Expression General Requirements.

-n number Specify the number of lines per screenful. The number argu?ment is a positive decimal integer. The -n option shall over?ride any values obtained from any other source.

-p command

Each time a screen from a new file is displayed or redis?

played (including as a result of more commands; for example, :p), execute the more command(s) in the command arguments in the order specified, as if entered by the user after the first screen has been displayed. No intermediate results shall be displayed (that is, if the command is a movement to a screen different from the normal first screen, only the screen resulting from the command shall be displayed.) If any of the commands fail for any reason, an informational message to this effect shall be written, and no further commands specified using the -p option shall be executed for this file.

-s Behave as if consecutive empty lines were a single empty line.

-t tagstring

Write the screenful of the file containing the tag named by the tagstring argument. See the ctags utility. The tags fea? ture represented by -t tagstring and the :t command is op? tional. It shall be provided on any system that also provides a conforming implementation of ctags; otherwise, the use of -t produces undefined results.

The filename resulting from the -t option shall be logically added as a prefix to the list of command line files, as if specified by the user. If the tag named by the tagstring ar? gument is not found, it shall be an error, and more shall take no further action.

If the tag specifies a line number, the first line of the display shall contain the beginning of that line. If the tag specifies a pattern, the first line of the display shall con? tain the beginning of the matching text from the first line of the file that contains that pattern. If the line does not exist in the file or matching text is not found, an informa? tional message to this effect shall be displayed, and more shall display the default screen as if -t had not been speci?

fied.

If both the -t tagstring and -p command options are given, the -t tagstring shall be processed first; that is, the file and starting line for the display shall be as specified by -t, and then the -p more command shall be executed. If the line (matching text) specified by the -t command does not ex? ist (is not found), no -p more command shall be executed for this file at any time.

-u Treat a <backspace> as a printable control character, dis? played as an implementation-defined character sequence (see the EXTENDED DESCRIPTION section), suppressing backspacing and the special handling that produces underlined or standout mode text on some terminal types. Also, do not ignore a <carriage-return> at the end of a line.

OPERANDS

The following operand shall be supported:

file A pathname of an input file. If no file operands are speci? fied, the standard input shall be used. If a file is '-', the standard input shall be read at that point in the sequence.

STDIN

The standard input shall be used only if no file operands are speci? fied, or if a file operand is '-'.

INPUT FILES

The input files being examined shall be text files. If standard output is a terminal, standard error shall be used to read commands from the user. If standard output is a terminal, standard error is not readable, and command input is needed, more may attempt to obtain user commands from the controlling terminal (for example, /dev/tty); otherwise, more shall terminate with an error indicating that it was unable to read user commands. If standard output is not a terminal, no error shall re? sult if standard error cannot be opened for reading.

ENVIRONMENT VARIABLES

The following environment variables shall affect the execution of more:

COLUMNS Override the system-selected horizontal display line size. See the Base Definitions volume of POSIX.1?2017, Chapter 8, Environment Variables for valid values and results when it is unset or null.

EDITOR Used by the v command to select an editor. See the EXTENDED DESCRIPTION section.

LANG Provide a default value for the internationalization vari? ables that are unset or null. (See the Base Definitions vol? ume of POSIX.1?2017, Section 8.2, Internationalization Vari? ables for the precedence of internationalization variables used to determine the values of locale categories.)

LC_ALL If set to a non-empty string value, override the values of all the other internationalization variables.

LC_COLLATE

Determine the locale for the behavior of ranges, equivalence classes, and multi-character collating elements within regu? lar expressions.

LC_CTYPE Determine the locale for the interpretation of sequences of bytes of text data as characters (for example, single-byte as opposed to multi-byte characters in arguments and input files) and the behavior of character classes within regular expressions.

LC_MESSAGES

Determine the locale that should be used to affect the format and contents of diagnostic messages written to standard error and informative messages written to standard output.

NLSPATH Determine the location of message catalogs for the processing of LC_MESSAGES.

LINES Override the system-selected vertical screen size, used as the number of lines in a screenful. See the Base Definitions volume of POSIX.1?2017, Chapter 8, Environment Variables for valid values and results when it is unset or null. The -n op? tion shall take precedence over the LINES variable for deter?

mining the number of lines in a screenful.

MORE Determine a string containing options described in the OP? TIONS section preceded with <hyphen-minus> characters and <blank>-separated as on the command line. Any command line options shall be processed after those in the MORE variable, as if the command line were:

more $MORE options operands

The MORE variable shall take precedence over the TERM and LINES variables for determining the number of lines in a screenful.

TERM Determine the name of the terminal type. If this variable is unset or null, an unspecified default terminal type is used.

ASYNCHRONOUS EVENTS

Default.

STDOUT

The standard output shall be used to write the contents of the input files.

STDERR

The standard error shall be used for diagnostic messages and user com? mands (see the INPUT FILES section), and, if standard output is a ter? minal device, to write a prompting string. The prompting string shall appear on the screen line below the last line of the file displayed in the current screenful. The prompt shall contain the name of the file currently being examined and shall contain an end-of-file indication and the name of the next file, if any, when prompting at the end-of-file. If an error or informational message is displayed, it is unspeci? fied whether it is contained in the prompt. If it is not contained in the prompt, it shall be displayed and then the user shall be prompted for a continuation character, at which point another message or the user prompt may be displayed. The prompt is otherwise unspecified. It is unspecified whether informational messages are written for other user commands.

OUTPUT FILES

None.

EXTENDED DESCRIPTION

The following section describes the behavior of more when the standard output is a terminal device. If the standard output is not a terminal device, no options other than -s shall have any effect, and all input files shall be copied to standard output otherwise unmodified, at which time more shall exit without further action.

The number of lines available per screen shall be determined by the -n option, if present, or by examining values in the environment (see the ENVIRONMENT VARIABLES section). If neither method yields a number, an unspecified number of lines shall be used.

The maximum number of lines written shall be one less than this number, because the screen line after the last line written shall be used to write a user prompt and user input. If the number of lines in the screen is less than two, the results are undefined. It is unspecified whether user input is permitted to be longer than the remainder of the single line where the prompt has been written.

The number of columns available per line shall be determined by examin? ing values in the environment (see the ENVIRONMENT VARIABLES section), with a default value as described in the Base Definitions volume of POSIX.1?2017, Chapter 8, Environment Variables.

Lines that are longer than the display shall be folded; the length at which folding occurs is unspecified, but should be appropriate for the output device. Folding may occur between glyphs of single characters that take up multiple display columns.

When standard output is a terminal and -u is not specified, more shall treat <backspace> and <carriage-return> characters specially:

* A character, followed first by a sequence of n <backspace> charac? ters (where n is the same as the number of column positions that the character occupies), then by n <underscore> characters ('_'), shall cause that character to be written as underlined text, if the terminal type supports that. The n <underscore> characters, fol? lowed first by n <backspace> characters, then any character with n

column positions, shall also cause that character to be written as underlined text, if the terminal type supports that.

* A sequence of n <backspace> characters (where n is the same as the number of column positions that the previous character occupies) that appears between two identical printable characters shall cause the first of those two characters to be written as emboldened text (that is, visually brighter, standout mode, or inverse-video mode), if the terminal type supports that, and the second to be discarded. Immediately subsequent occurrences of <backspace>/character pairs for that same character shall also be discarded. (For example, the sequence "a\ba\ba\ba" is interpreted as a single emboldened 'a'.)
* The more utility shall logically discard all other <backspace> characters from the line as well as the character which precedes them, if any.
* A <carriage-return> at the end of a line shall be ignored, rather than being written as a non-printable character, as described in the next paragraph.

It is implementation-defined how other non-printable characters are written. Implementations should use the same format that they use for the ex print command; see the OPTIONS section within the ed utility. It is unspecified whether a multi-column character shall be separated if it crosses a display line boundary; it shall not be discarded. The be? havior is unspecified if the number of columns on the display is less than the number of columns any single character in the line being dis? played would occupy.

When each new file is displayed (or redisplayed), more shall write the first screen of the file. Once the initial screen has been written, more shall prompt for a user command. If the execution of the user com? mand results in a screen that has lines in common with the current screen, and the device has sufficient terminal capabilities, more shall scroll the screen; otherwise, it is unspecified whether the screen is scrolled or redrawn.

For all files but the last (including standard input if no file was

specified, and for the last file as well, if the -e option was not specified), when more has written the last line in the file, more shall prompt for a user command. This prompt shall contain the name of the next file as well as an indication that more has reached end-of-file.

If the user command is f, <control>?F, <space>, j, <newline>, d, <con? trol>?D, or s, more shall display the next file. Otherwise, if display? ing the last file, more shall exit. Otherwise, more shall execute the user command specified.

Several of the commands described in this section display a previous screen from the input stream. In the case that text is being taken from a non-rewindable stream, such as a pipe, it is implementation-defined how much backwards motion is supported. If a command cannot be executed because of a limitation on backwards motion, an error message to this effect shall be displayed, the current screen shall not change, and the user shall be prompted for another command.

If a command cannot be performed because there are insufficient lines to display, more shall alert the terminal. If a command cannot be per? formed because there are insufficient lines to display or a / command fails: if the input is the standard input, the last screen in the file may be displayed; otherwise, the current file and screen shall not change, and the user shall be prompted for another command.

The interactive commands in the following sections shall be supported.

Some commands can be preceded by a decimal integer, called count in the following descriptions. If not specified with the command, count shall default to 1. In the following descriptions, pattern is a basic regular expression, as described in the Base Definitions volume of POSIX.1?2017, Section 9.3, Basic Regular Expressions. The term ``exam? ine'' is historical usage meaning ``open the file for viewing''; for example, more foo would be expressed as examining file foo.

In the following descriptions, unless otherwise specified, line is a line in the more display, not a line from the file being examined.

In the following descriptions, the current position refers to two things:

1. The position of the current line on the screen

2. The line number (in the file) of the current line on the screen

Usually, the line on the screen corresponding to the current position is the third line on the screen. If this is not possible (there are fewer than three lines to display or this is the first page of the file, or it is the last page of the file), then the current position is either the first or last line on the screen as described later.

Help

Synopsis:

h

Write a summary of these commands and other implementation-defined com? mands. The behavior shall be as if the more utility were executed with the -e option on a file that contained the summary information. The user shall be prompted as described earlier in this section when end-of-file is reached. If the user command is one of those specified to continue to the next file, more shall return to the file and screen state from which the h command was executed.

Scroll Forward One Screenful

Synopsis:

[count]f

[count]<control>-F

Scroll forward count lines, with a default of one screenful. If count is more than the screen size, only the final screenful shall be writ? ten.

Scroll Backward One Screenful

Synopsis:

[count]b

[count]<control>-B

Scroll backward count lines, with a default of one screenful (see the -n option). If count is more than the screen size, only the final screenful shall be written.

Scroll Forward One Line

Synopsis:

[count]<space>

[count]j

[count]<newline>

Scroll forward count lines. The default count for the <space> shall be one screenful; for j and <newline>, one line. The entire count lines shall be written, even if count is more than the screen size.

Scroll Backward One Line

Synopsis:

[count]k

Scroll backward count lines. The entire count lines shall be written, even if count is more than the screen size.

Scroll Forward One Half Screenful

Synopsis:

[count]d

[count]<control>-D

Scroll forward count lines, with a default of one half of the screen size. If count is specified, it shall become the new default for subse? quent d, <control>?D, and u commands.

Skip Forward One Line

Synopsis:

[count]s

Display the screenful beginning with the line count lines after the last line on the current screen. If count would cause the current posi? tion to be such that less than one screenful would be written, the last screenful in the file shall be written.

Scroll Backward One Half Screenful

Synopsis:

[count]u

[count]<control>-U

Scroll backward count lines, with a default of one half of the screen size. If count is specified, it shall become the new default for subse? quent d, <control>-D, u, and <control>-U commands. The entire count lines shall be written, even if count is more than the screen size.

Go to Beginning of File

Synopsis:

[count]g

Display the screenful beginning with line count.

Go to End-of-File

Synopsis:

[count]G

If count is specified, display the screenful beginning with the line count. Otherwise, display the last screenful of the file.

Refresh the Screen

Synopsis:

r

<control>-L

Refresh the screen.

Discard and Refresh

Synopsis:

R

Refresh the screen, discarding any buffered input. If the current file is non-seekable, buffered input shall not be discarded and the R com? mand shall be equivalent to the r command.

Mark Position

Synopsis:

mletter

Mark the current position with the letter named by letter, where letter represents the name of one of the lowercase letters of the portable character set. When a new file is examined, all marks may be lost.

Return to Mark

Synopsis:

'letter

Return to the position that was previously marked with the letter named by letter, making that line the current position.

Return to Previous Position

Synopsis:

''

Return to the position from which the last large movement command was executed (where a ``large movement'' is defined as any movement of more than a screenful of lines). If no such movements have been made, return to the beginning of the file.

Search Forward for Pattern

Synopsis:

[count]/[!]pattern<newline>

Display the screenful beginning with the countth line containing the pattern. The search shall start after the first line currently dis? played. The null regular expression ('/' followed by a <newline>) shall repeat the search using the previous regular expression, with a default count. If the character '!' is included, the matching lines shall be those that do not contain the pattern. If no match is found for the pattern, a message to that effect shall be displayed.

Search Backward for Pattern

Synopsis:

[count]?[!]pattern<newline>

Display the screenful beginning with the countth previous line contain? ing the pattern. The search shall start on the last line before the first line currently displayed. The null regular expression ('?' fol? lowed by a <newline>) shall repeat the search using the previous regu? lar expression, with a default count. If the character '!' is in? cluded, matching lines shall be those that do not contain the pattern. If no match is found for the pattern, a message to that effect shall be displayed.

Repeat Search

Synopsis:

[count]n

Repeat the previous search for countth line containing the last pattern (or not containing the last pattern, if the previous search was "/!" or "?!").

Repeat Search in Reverse

Synopsis:

[count]N

Repeat the search in the opposite direction of the previous search for the countth line containing the last pattern (or not containing the last pattern, if the previous search was "/!" or "?!").

Examine New File

Synopsis:

:e [filename]<newline>

Examine a new file. If the filename argument is not specified, the cur? rent file (see the :n and :p commands below) shall be re-examined. The filename shall be subjected to the process of shell word expansions (see Section 2.6, Word Expansions); if more than a single pathname re? sults, the effects are unspecified. If filename is a <number-sign> ('#'), the previously examined file shall be re-examined. If filename is not accessible for any reason (including that it is a non-seekable file), an error message to this effect shall be displayed and the cur? rent file and screen shall not change.

Examine Next File

Synopsis:

[count]:n

Examine the next file. If a number count is specified, the countth next file shall be examined. If filename refers to a non-seekable file, the results are unspecified.

Examine Previous File

Synopsis:

[count]:p

Examine the previous file. If a number count is specified, the countth previous file shall be examined. If filename refers to a non-seekable file, the results are unspecified.

Go to Tag

Synopsis:

:t tagstring<newline>

If the file containing the tag named by the tagstring argument is not

the current file, examine the file, as if the :e command was executed with that file as the argument. Otherwise, or in addition, display the screenful beginning with the tag, as described for the -t option (see the OPTIONS section). If the ctags utility is not supported by the sys? tem, the use of :t produces undefined results.

Invoke Editor

Synopsis:

v

Invoke an editor to edit the current file being examined. If standard input is being examined, the results are unspecified. The name of the editor shall be taken from the environment variable EDITOR, or shall default to vi. If the last pathname component in EDITOR is either vi or ex, the editor shall be invoked with a -c linenumber command line argument, where linenumber is the line number of the file line contain? ing the display line currently displayed as the first line of the screen. It is implementation-defined whether line-setting options are passed to editors other than vi and ex.

When the editor exits, more shall resume with the same file and screen as when the editor was invoked.

Display Position

Synopsis:

=

<control>-G

Write a message for which the information references the first byte of the line after the last line of the file on the screen. This message shall include the name of the file currently being examined, its number relative to the total number of files there are to examine, the line number in the file, the byte number and the total bytes in the file, and what percentage of the file precedes the current position. If more is reading from standard input, or the file is shorter than a single screen, the line number, the byte number, the total bytes, and the per? centage need not be written.

Quit

Synopsis:

q

:q

ZZ

Exit more.

EXIT STATUS

The following exit values shall be returned:

0 Successful completion.

>0 An error occurred.

CONSEQUENCES OF ERRORS

If an error is encountered accessing a file when using the :n command, more shall attempt to examine the next file in the argument list, but the final exit status shall be affected. If an error is encountered ac? cessing a file via the :p command, more shall attempt to examine the previous file in the argument list, but the final exit status shall be affected. If an error is encountered accessing a file via the :e com? mand, more shall remain in the current file and the final exit status shall not be affected.

The following sections are informative.

APPLICATION USAGE

When the standard output is not a terminal, only the -s filter-modifi? cation option is effective. This is based on historical practice. For example, a typical implementation of man pipes its output through more -s to squeeze excess white space for terminal users. When man is piped to lp, however, it is undesirable for this squeezing to happen.

EXAMPLES

The -p allows arbitrary commands to be executed at the start of each file. Examples are:

more -p G file1 file2

Examine each file starting with its last screenful.

more -p 100 file1 file2

Examine each file starting with line 100 in the current position (usually the third line, so line 98 would be the first line writ?

ten).

more -p /100 file1 file2

Examine each file starting with the first line containing the string "100" in the current position

RATIONALE

The more utility, available in BSD and BSD-derived systems, was chosen as the prototype for the POSIX file display program since it is more widely available than either the public-domain program less or than pg, a pager provided in System V. The 4.4 BSD more is the model for the features selected; it is almost fully upwards-compatible from the 4.3 BSD version in wide use and has become more amenable for vi users. Sev? eral features originally derived from various file editors, found in both less and pg, have been added to this volume of POSIX.1?2017 as they have proved extremely popular with users.

There are inconsistencies between more and vi that result from histori? cal practice. For example, the single-character commands h, f, b, and <space> are screen movers in more, but cursor movers in vi. These in? consistencies were maintained because the cursor movements are not ap? plicable to more and the powerful functionality achieved without the use of the control key justifies the differences.

The tags interface has been included in a program that is not a text editor because it promotes another degree of consistent operation with vi. It is conceivable that the paging environment of more would be su? perior for browsing source code files in some circumstances.

The operating mode referred to for block-mode terminals effectively adds a <newline> to each Synopsis line that currently has none. So, for example, d<newline> would page one screenful. The mode could be trig? gered by a command line option, environment variable, or some other method. The details are not imposed by this volume of POSIX.1?2017 be? cause there are so few systems known to support such terminals. Never? theless, it was considered that all systems should be able to support more given the exception cited for this small community of terminals because, in comparison to vi, the cursor movements are few and the com?

mand set relatively amenable to the optional <newline> characters.

Some versions of more provide a shell escaping mechanism similar to the ex ! command. The standard developers did not consider that this was necessary in a paginator, particularly given the wide acceptance of multiple window terminals and job control features. (They chose to re? tain such features in the editors and mailx because the shell interac? tion also gives an opportunity to modify the editing buffer, which is not applicable to more.)

The -p (position) option replaces the + command because of the Utility Syntax Guidelines. The +command option is no longer specified by POSIX.1?2008 but may be present in some implementations. In early pro? posals, it took a pattern argument, but historical less provided the more general facility of a command. It would have been desirable to use the same -c as ex and vi, but the letter was already in use.

The text stating ``from a non-rewindable stream ... implementations may limit the amount of backwards motion supported'' would allow an imple? mentation that permitted no backwards motion beyond text already on the screen. It was not possible to require a minimum amount of backwards motion that would be effective for all conceivable device types. The implementation should allow the user to back up as far as possible, within device and reasonable memory allocation constraints.

Historically, non-printable characters were displayed using the ARPA standard mappings, which are as follows:

1. Printable characters are left alone.
2. Control characters less than \177 are represented as followed by the character offset from the '@' character in the ASCII map; for example, \007 is represented as 'G'.
3. \177 is represented as followed by '?'.

The display of characters having their eighth bit set was less stan? dard. Existing implementations use hex (0x00), octal (\000), and a meta-bit display. (The latter displayed characters with their eighth bit set as the two characters "M-", followed by the seven-bit display as described previously.) The latter probably has the best claim to

historical practice because it was used with the -v option of 4 BSD and 4 BSD-derived versions of the cat utility since 1980.

No specific display format is required by POSIX.1?2008. Implementations are encouraged to conform to historic practice in the absence of any strong reason to diverge.

FUTURE DIRECTIONS

None.

SEE ALSO

Chapter 2, Shell Command Language, ctags, ed, ex, vi

The Base Definitions volume of POSIX.1?2017, Chapter 8, Environment Variables, Section 9.2, Regular Expression General Requirements, Section 9.3, Basic Regular Expressions, Section 12.2, Utility Syntax Guidelines

COPYRIGHT