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 'm4.1p' command

$ man m4.1p

M4(1P) POSIX Programmer's Manual M4(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

m4 ? macro processor

SYNOPSIS

m4 [-s] [-D name[=val]]... [-U name]... file...

DESCRIPTION

The m4 utility is a macro processor that shall read one or more text files, process them according to their included macro statements, and write the results to standard output.

OPTIONS

The m4 utility shall conform to the Base Definitions volume of POSIX.1?2017, Section 12.2, Utility Syntax Guidelines, except that the order of the -D and -U options shall be significant, and options can be interspersed with operands.

The following options shall be supported:

-s Enable line synchronization output for the c99 preprocessor phase (that is, #line directives).

-D name[=val]

Define name to val or to null if =val is omitted.

-U name Undefine name.

OPERANDS

The following operand shall be supported:

file A pathname of a text file to be processed. If no file is given, or if it is '-', the standard input shall be read.

STDIN

The standard input shall be a text file that is used if no file operand is given, or if it is '-'.

INPUT FILES

The input file named by the file operand shall be a text file.

ENVIRONMENT VARIABLES

The following environment variables shall affect the execution of m4:

LANG Provide a default value for the internationalization variables that are unset or null. (See the Base Definitions volume of POSIX.1?2017, Section 8.2, Internationalization Variables 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_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).

LC_MESSAGES

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

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

ASYNCHRONOUS EVENTS

Default.

STDOUT

The standard output shall be the same as the input files, after being processed for macro expansion.

STDERR

The standard error shall be used to display strings with the errprint macro, macro tracing enabled by the traceon macro, the defined text for macros written by the dumpdef macro, or for diagnostic messages.

OUTPUT FILES

None.

EXTENDED DESCRIPTION

The m4 utility shall compare each token from the input against the set of built-in and user-defined macros. If the token matches the name of a macro, then the token shall be replaced by the macro's defining text, if any, and rescanned for matching macro names. Once no portion of the token matches the name of a macro, it shall be written to standard out? put. Macros may have arguments, in which case the arguments shall be substituted into the defining text before it is rescanned.

Macro calls have the form:

name(arg1, arg2, ..., argn)

Macro names shall consist of letters, digits, and underscores, where the first character is not a digit. Tokens not of this form shall not be treated as macros.

The application shall ensure that the <left-parenthesis> immediately follows the name of the macro. If a token matching the name of a macro is not followed by a <left-parenthesis>, it is handled as a use of that macro without arguments.

If a macro name is followed by a <left-parenthesis>, its arguments are the <comma>-separated tokens between the <left-parenthesis> and the matching <right-parenthesis>. Unquoted white-space characters preced? ing each argument shall be ignored. All other characters, including trailing white-space characters, are retained. <comma> characters en? closed between <left-parenthesis> and <right-parenthesis> characters do not delimit arguments.

Arguments are positionally defined and referenced. The string "$1" in

the defining text shall be replaced by the first argument. Systems shall support at least nine arguments; only the first nine can be ref? erenced, using the strings "$1" to "$9", inclusive. The string "$0" is replaced with the name of the macro. The string "$#" is replaced by the number of arguments as a string. The string "$*" is replaced by a list of all of the arguments, separated by <comma> characters. The string "$@" is replaced by a list of all of the arguments separated by <comma> characters, and each argument is quoted using the current left and right quoting strings. The string "${" produces unspecified behavior.

If fewer arguments are supplied than are in the macro definition, the omitted arguments are taken to be null. It is not an error if more ar? guments are supplied than are in the macro definition.

No special meaning is given to any characters enclosed between matching left and right quoting strings, but the quoting strings are themselves discarded. By default, the left quoting string consists of a grave ac? cent (backquote) and the right quoting string consists of an acute ac? cent (single-quote); see also the changequote macro.

Comments are written but not scanned for matching macro names; by de? fault, the begin-comment string consists of the <number-sign> character and the end-comment string consists of a <newline>. See also the changecom and dnl macros.

The m4 utility shall make available the following built-in macros. They can be redefined, but once this is done the original meaning is lost. Their values shall be null unless otherwise stated. In the descriptions below, the term defining text refers to the value of the macro: the second argument to the define macro, among other things. Except for the first argument to the eval macro, all numeric arguments to built-in macros shall be interpreted as decimal values. The string values pro? duced as the defining text of the decr, divnum, incr, index, len, and sysval built-in macros shall be in the form of a decimal-constant as defined in the C language.

changecom The changecom macro shall set the begin-comment and end-com? ment strings. With no arguments, the comment mechanism shall

be disabled. With a single non-null argument, that argument shall become the begin-comment and the <newline> shall become the end-comment string. With two non-null arguments, the first argument shall become the begin-comment string and the second argument shall become the end-comment string. The behavior is unspecified if either argument is provided but null. Systems shall support comment strings of at least five characters.

changequote

The changequote macro shall set the begin-quote and end-quote strings. With no arguments, the quote strings shall be set to the default values (that is, `'). The behavior is unspecified if there is a single argument or either argument is null. With two non-null arguments, the first argument shall become the begin-quote string and the second argument shall become the end-quote string. Systems shall support quote strings of at least five characters.

decr The defining text of the decr macro shall be its first argument decremented by 1. It shall be an error to specify an argument containing any non-numeric characters. The behavior is unspecified if decr is not immediately followed by a <left-parenthesis>.

define The second argument shall become the defining text of the macro whose name is the first argument. It is unspecified whether the define macro deletes all prior definitions of the macro named by its first argument or preserves all but the current definition of the macro. The behavior is unspecified if define is not immediately followed by a <left-parenthesis>.

defn The defining text of the defn macro shall be the quoted definition (using the current quoting strings) of its arguments. The behavior is unspecified if defn is not immediately followed by a <left-parenthesis>.

divert The m4 utility maintains nine temporary buffers, numbered 1 to 9, inclusive. When the last of the input has been pro? cessed, any output that has been placed in these buffers shall be written to standard output in buffer-numerical or? der. The divert macro shall divert future output to the buf? fer specified by its argument. Specifying no argument or an argument of 0 shall resume the normal output process. Output diverted to a stream with a negative number shall be dis? carded. Behavior is implementation-defined if a stream number larger than 9 is specified. It shall be an error to specify an argument containing any non-numeric characters.

divnum The defining text of the divnum macro shall be the number of the current output stream as a string.

dnl The dnl macro shall cause m4 to discard all input characters up to and including the next <newline>.

dumpdef The dumpdef macro shall write the defined text to standard error for each of the macros specified as arguments, or, if no arguments are specified, for all macros.

errprint The errprint macro shall write its arguments to standard er? ror. The behavior is unspecified if errprint is not immedi? ately followed by a <left-parenthesis>.

eval The eval macro shall evaluate its first argument as an arith? metic expression, using signed integer arithmetic with at least 32-bit precision. At least the following C-language op? erators shall be supported, with precedence, associativity, and behavior as described in Section 1.1.2.1, Arithmetic Pre? cision and Operations:

()

unary +

unary -

~

!

binary *

/

%

binary +

binary -

<<

>>

<

<=

>

>=

==

!=

binary &

^

|

&&

||

Systems shall support octal and hexadecimal numbers as in the ISO C standard. The second argument, if specified, shall set the radix for the result; if the argument is blank or unspec? ified, the default is 10. Behavior is unspecified if the radix falls outside the range 2 to 36, inclusive. The third argument, if specified, sets the minimum number of digits in the result. Behavior is unspecified if the third argument is less than zero. It shall be an error to specify the second or third argument containing any non-numeric characters. The be? havior is unspecified if eval is not immediately followed by a <left-parenthesis>.

ifdef If the first argument to the ifdef macro is defined, the defining text shall be the second argument. Otherwise, the defining text shall be the third argument, if specified, or the null string, if not. The behavior is unspecified if ifdef is not immediately followed by a <left-parenthesis>.

ifelse The ifelse macro takes three or more arguments. If the first two arguments compare as equal strings (after macro expansion of both arguments), the defining text shall be the third argument. If the first two arguments do not compare as equal strings and there are three arguments, the defining text shall be null. If the first two arguments do not compare as equal strings and there are four or five arguments, the defining text shall be the fourth argument. If the first two arguments do not compare as equal strings and there are six or more arguments, the first three arguments shall be discarded and processing shall restart with the remaining arguments. The behavior is unspecified if ifelse is not immediately followed by a <left-parenthesis>.

include The defining text for the include macro shall be the contents of the file named by the first argument. It shall be an error if the file cannot be read. The behavior is unspecified if include is not immediately followed by a <left-parenthesis>.

incr The defining text of the incr macro shall be its first argument incremented by 1. It shall be an error to specify an argument containing any non-numeric characters. The behavior is unspecified if incr is not immediately followed by a <left-parenthesis>.

index The defining text of the index macro shall be the first character position (as a string) in the first argument where a string matching the second argument begins (zero origin), or -1 if the second argument does not occur. The behavior is unspecified if index is not immediately followed by a <left-parenthesis>.

len The defining text of the len macro shall be the length (as a string) of the first argument. The behavior is unspecified if len is not immediately followed by a <left-parenthesis>.

m4exit Exit from the m4 utility. If the first argument is specified, it shall be the exit code. If no argument is specified, the

exit code shall be zero. It shall be an error to specify an argument containing any non-numeric characters. If the first argument is zero or no argument is specified, and an error has previously occurred (for example, a file operand that could not be opened), it is unspecified whether the exit sta? tus is zero or non-zero.

m4wrap The first argument shall be processed when EOF is reached. If the m4wrap macro is used multiple times, the arguments speci? fied shall be processed in the order in which the m4wrap macros were processed. The behavior is unspecified if m4wrap is not immediately followed by a <left-parenthesis>.

maketemp The defining text shall be the first argument, with any trailing 'X' characters replaced with the current process ID as a string. The behavior is unspecified if maketemp is not immediately followed by a <left-parenthesis>.

mkstemp The defining text shall be as if it were the resulting path? name after a successful call to the mkstemp() function de? fined in the System Interfaces volume of POSIX.1?2017 called with the first argument to the macro invocation. If a file is created, that file shall be closed. If a file could not be created, the m4 utility shall write a diagnostic message to standard error and shall continue processing input but its final exit status shall be non-zero; the defining text of the macro shall be the empty string. The behavior is unspecified if mkstemp is not immediately followed by a <left-parenthe? sis>.

popdef The popdef macro shall delete the current definition of its arguments, replacing that definition with the previous one. If there is no previous definition, the macro is undefined. The behavior is unspecified if popdef is not immediately fol? lowed by a <left-parenthesis>.

pushdef The pushdef macro shall be equivalent to the define macro with the exception that it shall preserve any current defini?

tion for future retrieval using the popdef macro. The behav? ior is unspecified if pushdef is not immediately followed by a <left-parenthesis>.

shift The defining text for the shift macro shall be a comma-sepa? rated list of its arguments except the first one. Each argu? ment shall be quoted using the current quoting strings. The behavior is unspecified if shift is not immediately followed by a <left-parenthesis>.

sinclude The sinclude macro shall be equivalent to the include macro, except that it shall not be an error if the file is inacces? sible. The behavior is unspecified if sinclude is not imme? diately followed by a <left-parenthesis>.

substr The defining text for the substr macro shall be the substring of the first argument beginning at the zero-offset character position specified by the second argument. The third argu? ment, if specified, shall be the number of characters to se? lect; if not specified, the characters from the starting point to the end of the first argument shall become the defining text. It shall not be an error to specify a starting point beyond the end of the first argument and the defining text shall be null. It shall be an error to specify an argu? ment containing any non-numeric characters. The behavior is unspecified if substr is not immediately followed by a <left-parenthesis>.

syscmd The syscmd macro shall interpret its first argument as a shell command line. The defining text shall be the string re? sult of that command. The string result shall not be res? canned for macros while setting the defining text. No output redirection shall be performed by the m4 utility. The exit status value from the command can be retrieved using the sys? val macro. The behavior is unspecified if syscmd is not imme? diately followed by a <left-parenthesis>.

sysval The defining text of the sysval macro shall be the exit value

of the utility last invoked by the syscmd macro (as a string).

traceon The traceon macro shall enable tracing for the macros speci? fied as arguments, or, if no arguments are specified, for all macros. The trace output shall be written to standard error in an unspecified format.

traceoff The traceoff macro shall disable tracing for the macros spec? ified as arguments, or, if no arguments are specified, for all macros.

translit The defining text of the translit macro shall be the first argument with every character that occurs in the second argu? ment replaced with the corresponding character from the third argument. If no replacement character is specified for some source character because the second argument is longer than the third argument, that character shall be deleted from the first argument in translit's defining text. The behavior is unspecified if the '-' character appears within the second or third argument anywhere besides the first or last character. The behavior is unspecified if the same character appears more than once in the second argument. The behavior is un? specified if translit is not immediately followed by a <left-parenthesis>.

undefine The undefine macro shall delete all definitions (including those preserved using the pushdef macro) of the macros named by its arguments. The behavior is unspecified if undefine is not immediately followed by a <left-parenthesis>.

undivert The undivert macro shall cause immediate output of any text in temporary buffers named as arguments, or all temporary buffers if no arguments are specified. Buffers can be undi? verted into other temporary buffers. Undiverting shall dis? card the contents of the temporary buffer. The behavior is unspecified if an argument contains any non-numeric charac? ters.

EXIT STATUS

The following exit values shall be returned:

0 Successful completion.

>0 An error occurred

If the m4exit macro is used, the exit value can be specified by the in?

put file.

CONSEQUENCES OF ERRORS

Default.

The following sections are informative.

APPLICATION USAGE

The defn macro is useful for renaming macros, especially built-ins.

Since eval defers to the ISO C standard, some operations have undefined behavior. In some implementations, division or remainder by zero cause a fatal signal, even if the division occurs on the short-circuited branch of "&&" or "||". Any operation that overflows in signed arith? metic produces undefined behavior. Likewise, using the shift operators with a shift amount that is not positive and smaller than the precision is undefined, as is shifting a negative number to the right. Histori? cally, not all implementations obeyed C-language precedence rules: '~' and '!' were lower than '=='; '==' and '!=' were not lower than '<'; and '|' was not lower than '^'; the liberal use of "()" can force the desired precedence even with these non-compliant implementations. Fur? thermore, some traditional implementations treated '^' as an exponenti? ation operator, although most implementations now use "**" as an exten? sion for this purpose.

When a macro has been multiply defined via the pushdef macro, it is un? specified whether the define macro will alter only the most recent def? inition (as though by popdef and pushdef), or replace the entire stack of definitions with a single definition (as though by undefine and pushdef). An application desiring particular behavior for the define macro in this case can redefine it accordingly.

Applications should use the mkstemp macro instead of the obsolescent maketemp macro for creating temporary files.

EXAMPLES

If the file m4src contains the lines:

```
The value of `VER' is "VER".
ifdef(`VER', ``VER'' is defined to be VER., VER is not defined.)
ifelse(VER, 1, ``VER'' is `VER'.)
ifelse(VER, 2, ``VER'' is `VER'., ``VER'' is not 2.)
end
```

then the command

```
m4 m4src
```

or the command:

```
m4 -U VER m4src
```

produces the output:

```
The value of VER is "VER".
VER is not defined.
VER is not 2.
end
```

The command:

```
m4 -D VER m4src
```

produces the output:

```
The value of VER is "".
VER is defined to be .
VER is not 2.
end
```

The command:

```
m4 -D VER=1 m4src
```

produces the output:

```
The value of VER is "1".
VER is defined to be 1.
VER is 1.
VER is not 2.
end
```

The command:

```
m4 -D VER=2 m4src
```

produces the output:

```
The value of VER is "2".
VER is defined to be 2.
VER is 2.
end
```

RATIONALE

Historic System V-based behavior treated "${" in a macro definition as two literal characters. However, this sequence is left unspecified so that implementations may offer extensions such as "${11}" meaning the eleventh positional parameter. Macros can still be defined with appro? priate uses of nested quoting to result in a literal "${" in the output after rescanning removes the nested quotes.

In the translit built-in, historic System V-based behavior treated '-' as a literal; GNU behavior treats it as a range. This version of the standard allows either behavior.

FUTURE DIRECTIONS

None.

SEE ALSO

c99

The Base Definitions volume of POSIX.1?2017, Chapter 8, Environment Variables, Section 12.2, Utility Syntax Guidelines

files to man page format. To report such errors, see https://www.ker?

nel.org/doc/man-pages/reporting_bugs.html .