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 'nm.1p' command

$ man nm.1p

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

nm ? write the name list of an object file (DEVELOPMENT)

SYNOPSIS

nm [-APv] [-g|-u] [-t format] file...

nm [-APv] [-efox] [-g|-u] [-t format] file...

DESCRIPTION

The nm utility shall display symbolic information appearing in the ob? ject file, executable file, or object-file library named by file. If no symbolic information is available for a valid input file, the nm utility shall report that fact, but not consider it an error condition.

The default base used when numeric values are written is unspecified. On XSI-conformant systems, it shall be decimal if the -P option is not specified.

OPTIONS

The nm utility shall conform to the Base Definitions volume of POSIX.1?2017, Section 12.2, Utility Syntax Guidelines.

The following options shall be supported:

-A Write the full pathname or library name of an object on each line.

-e Write only external (global) and static symbol information.

-f Produce full output. Write redundant symbols (.text, .data, and .bss), normally suppressed.

-g Write only external (global) symbol information.

-o Write numeric values in octal (equivalent to -t o).

-P Write information in a portable output format, as specified in the STDOUT section.

-t format Write each numeric value in the specified format. The format shall be dependent on the single character used as the format option-argument:

d decimal (default if -P is not specified).

o octal.

x hexadecimal (default if -P is specified).

-u Write only undefined symbols.

-v Sort output by value instead of by symbol name.

-x Write numeric values in hexadecimal (equivalent to -t x).

OPERANDS

The following operand shall be supported:

file A pathname of an object file, executable file, or object-file library.

STDIN

See the INPUT FILES section.

INPUT FILES

The input file shall be an object file, an object-file library whose format is the same as those produced by the ar utility for link edit? ing, or an executable file. The nm utility may accept additional imple? mentation-defined object library formats for the input file.

ENVIRONMENT VARIABLES

The following environment variables shall affect the execution of nm:

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 character collation information for the symbol-name and symbol-value collation sequences.

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).

LC_MESSAGES
Determine the locale that should be used to affect the format and contents of diagnostic messages written to standard er?
ror.

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

ASYNCHRONOUS EVENTS

Default.

STDOUT

If symbolic information is present in the input files, then for each file or for each member of an archive, the nm utility shall write the following information to standard output. By default, the format is un?
specified, but the output shall be sorted by symbol name according to the collation sequence in the current locale.

* Library or object name, if -A is specified
* Symbol name
* Symbol type, which shall either be one of the following single characters or an implementation-defined type represented by a sin?
gle character:
  A Global absolute symbol.
  a Local absolute symbol.
  B Global ``bss'' (that is, uninitialized data space) symbol.

b Local bss symbol.

D Global data symbol.

d Local data symbol.

T Global text symbol.

t Local text symbol.

U Undefined symbol.

* Value of the symbol

* The size associated with the symbol, if applicable

This information may be supplemented by additional information specific to the implementation.

If the -P option is specified, the previous information shall be dis? played using the following portable format. The three versions differ depending on whether -t d, -t o, or -t x was specified, respectively:

"%s%s %s %d %d\n", <library/object name>, <name>, <type>, <value>, <size>

"%s%s %s %o %o\n", <library/object name>, <name>, <type>, <value>, <size>

"%s%s %s %x %x\n", <library/object name>, <name>, <type>, <value>, <size>

where <library/object name> shall be formatted as follows:

* If -A is not specified, <library/object name> shall be an empty string.

* If -A is specified and the corresponding file operand does not name a library:

"%s: ", <file>

* If -A is specified and the corresponding file operand names a li? brary. In this case, <object file> shall name the object file in the library containing the symbol being described:

"%s[%s]: ", <file>, <object file>

If -A is not specified, then if more than one file operand is specified or if only one file operand is specified and it names a library, nm shall write a line identifying the object containing the following sym? bols before the lines containing those symbols, in the form:

* If the corresponding file operand does not name a library:

"%s:\n", <file>

* If the corresponding file operand names a library; in this case, <object file> shall be the name of the file in the library contain? ing the following symbols:

"%s[%s]:\n", <file>, <object file>

If -P is specified, but -t is not, the format shall be as if -t x had been specified.

STDERR

The standard error shall be used only for diagnostic messages.

OUTPUT FILES

None.

EXTENDED DESCRIPTION

None.

EXIT STATUS

The following exit values shall be returned:

0 Successful completion.

>0 An error occurred.

CONSEQUENCES OF ERRORS

Default.

The following sections are informative.

APPLICATION USAGE

Mechanisms for dynamic linking make this utility less meaningful when applied to an executable file because a dynamically linked executable may omit numerous library routines that would be found in a statically linked executable.

EXAMPLES

None.

RATIONALE

Historical implementations of nm have used different bases for numeric output and supplied different default types of symbols that were re? ported. The -t format option, similar to that used in od and strings, can be used to specify the numeric base; -g and -u can be used to re?

strict the amount of output or the types of symbols included in the output.

The compromise of using -t format versus using -d, -o, and other simi? lar options was necessary because of differences in the meaning of -o between implementations. The -o option from BSD has been provided here as -A to avoid confusion with the -o from System V (which has been pro? vided here as -t and as -o on XSI-conformant systems).

The option list was significantly reduced from that provided by histor? ical implementations.

The nm description is a subset of both the System V and BSD nm utili? ties with no specified default output.

It was recognized that mechanisms for dynamic linking make this utility less meaningful when applied to an executable file (because a dynami? cally linked executable file may omit numerous library routines that would be found in a statically linked executable file), but the value of nm during software development was judged to outweigh other limita? tions.

The default output format of nm is not specified because of differences in historical implementations. The -P option was added to allow some type of portable output format. After a comparison of the different formats used in SunOS, BSD, SVR3, and SVR4, it was decided to create one that did not match the current format of any of these four systems. The format devised is easy to parse by humans, easy to parse in shell scripts, and does not need to vary depending on locale (because no Eng? lish descriptions are included). All of the systems currently have the information available to use this format.

The format given in nm STDOUT uses <space> characters between the fields, which may be any number of <blank> characters required to align the columns. The single-character types were selected to match histori? cal practice, and the requirement that implementation additions also be single characters made parsing the information easier for shell scripts.

None.

SEE ALSO

ar, c99

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