IEEE P1003.2a Draft 8 – December 1991

5.21 nm — Write the name list of an object file

This utility shall be provided on systems that support both the User Portability 7
Utilities Option and the Software Development Utilities Option. On other sys- 7
tems it is optional. 7

5.21.1 Synopsis

```
nm [−APv] [ −g | −u ] [−t format ] file ...
```

5.21.2 Description

The nm utility shall display symbolic information appearing in the object 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 that an error condition.

The default base used when numeric values are written is unspecified.

5.21.3 Options

The nm utility shall conform to the utility argument syntax guidelines described in 2.10.2.

The following options shall be supported by the implementation:

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

−g Write only external (global) symbol information.

−P Write information in a portable output format, as specified in 5.21.6.1.

−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 The offset shall be written in decimal.
- o The offset shall be written in octal.
- x The offset shall be written in hexadecimal.

−u Write only undefined symbols.

−v Sort output by value instead of alphabetically.

5.21.4 Operands

The following operand shall be supported by the implementation:

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

5.21.5 External Influences

5.21.5.1 Standard Input

See Input Files.

5.21.5.2 Input Files

The input file shall be an object file, an object-file library whose format is the 7
same as those produced by the `ar` utility for link editing (see 6.1), or an execut- 7
able file. The *nm* utility may accept additional implementation-defined object 7
library formats for the input file. 7

5.21.5.3 Environment Variables

The following environment variables shall affect the execution of `nm`:

LANG This variable shall determine the locale to use for the locale categories when both **LC_ALL** and the corresponding environment variable (beginning with **LC_**) do not specify a locale. See 2.6.

LC_ALL This variable shall determine the locale to be used to override any values for locale categories specified by the settings of **LANG** or any environment variables beginning with **LC_**.

LC_COLLATE This variable shall determine the locale for character collation information for the symbol-name and symbol-value collation sequences.

LC_CTYPE This variable shall determine the locale for the interpretation of sequences of bytes of text data as characters (e.g., single- versus multibyte characters in arguments). 7

LC_MESSAGES This variable shall determine the language in which messages should be written.

5.21.5.4 Asynchronous Events

Default.

5.21.6 External Effects

5.21.6.1 Standard Output

If symbolic information is present in the input file(s), 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 unspecified, but the output shall be sorted alphabetically by symbol name.

— 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 single character:

`A`	Global absolute symbol
`a`	Local absolute symbol
`B`	Global ‘‘bss’’(i.e., 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 displayed 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>* and its trailing colon and `<blank>`s are omitted when `−A` is not specified and *<size>* and its preceding `<blank>`s are omitted when inapplicable to the symbol type.

5.21.6.2 Standard Error

Used only for diagnostic messages.

5.21.6.3 Output Files

None.

5.21.7 Extended Description

None.

5.21.8 Exit Status

The `nm` utility shall exit with one of the following values:

0	Successful completion.
>0	An error occurred.

5.21.9 Consequences of Errors

Default.

5.21.10 Rationale. *(This subclause is not a part of P1003.2a)*

Usage, Examples

Historical implementations of `nm` have used different bases for numeric output and supplied different default types of symbols that were reported. 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 restrict the amount of output or the types of symbols included in the output.

History of Decisions Made

The option list was significantly reduced from that provided by historical implementations.

The `nm` description is a subset of both the System V and BSD `nm` utilities with no specified default output. 7

It was recognized that mechanisms for dynamic linking make this utility less meaningful when applied to a library, but the value of `nm` during software development was judged to outweigh other limitations.

The compromise of using `−t` *format* versus using `−d`, `−o`, and other similar 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 collision with the −o from System V (which has been provided here as `-t o`).

The default output format of `nm` is not specified because of differences in historical implementations. The `P` option was added to satisfy balloting objections relating to the use of `nm` in scripts as well as interactively. Some type of portable format was required. The format given in 5.21.6.1 uses spaces between the fields, which may be any number of <blank>s required to get the columns to line up. The single-character types were selected to match historical practice and the requirement that implementation additions also be single characters make parsing the information easier for shell scripts.

5.22 `patch` — Apply changes to files

5.22.1 Synopsis

`patch` [−blNR] [−c | −e | −n] [−d *dir*] [−D *define*] [−i *patchfile*] [−o *outfile*] 8
[−r *rejectfile*] [*file*] 8

5.22.2 Description

The `patch` utility shall read a source (‘‘patch’’) file containing any of the three forms of difference (‘‘diff’’) listings produced by the `diff` (see 4.17) utility (normal, context, or in the style of `ed`; see 4.20) and apply those differences to a file. By default, `patch` shall read from the standard input.

The `patch` utility shall attempt to determine the type of the diff listing, unless overruled by a −c, −e, or −n option.

If the patch file contains more than one patch, `patch` shall attempt to apply each
of them as if they came from separate patch files. (In this case the name of the 7
patch file must be determinable for each `diff` listing.) 7

5.22.3 Options

The `patch` utility shall conform to the utility argument syntax guidelines described in 2.10.2.

The following options shall be supported by the implementation:

−b Save a copy of the original contents of each modified file, before 7
the differences are applied, in a file of the same name with the 7
suffix `.orig` appended to it. If the file already exists, it shall be 7
overwritten; if multiple patches are applied to the same file, the 7
`.orig` file shall be written only for the first patch. When the 8
−o *outfile* option is also specified, *file* `.orig` shall not be created 7
but, if *outfile* already exists, *outfile* `.orig` shall be created. 7