NAME

nm – print name list of an object file

SYNOPSIS

/usr/ccs/bin/nm [**–AChlnPprRsTuVv**] [**–efox**] [**–g** | **–u**] [**–t** *format*] *file*. . .

/usr/xpg4/bin/nm [**–AChlnPprRsTuVv**] [**–efox**] [**–g** | **–u**] [**–t** *format*] *file*. . .

AVAILABILITY

/usr/ccs/bin/nm
SUNWbtool

/usr/xpg4/bin/nm
SUNWxcu4

DESCRIPTION

The **nm** command displays the symbol table of each ELF object file that is specified by *file*.

If no symbolic information is available for a valid input file, the **nm** utility will report that fact, but not consider it an error condition.

OPTIONS

The output of **nm** may be controlled using the following options:

–A Write the full path name or library name of an object on each line.

–C Demangle C++ symbol names before printing them out.

–e See **NOTES** below.

–f See **NOTES** below.

–g Write only external (global) symbol information.

–h Do not display the output heading data.

–l Distinguish between **WEAK** and **GLOBAL** symbols by appending a * to the key letter for **WEAK** symbols.

–n Sort external symbols by name before they are printed.

–o Print the value and size of a symbol in octal instead of decimal. (equivalent to **–t o**).

–p Produce easy to parse, terse output. Each symbol name is preceded by its value (blanks if undefined) and one of the letters:

A absolute symbol
B bss (uninitialized data space) symbol
D data object symbol
F file symbol.
N symbol has no type
S section symbol
T text symbol
U undefined

If the symbol's binding attribute is:

LOCAL the key letter is lower case

WEAK the key letter is upper case; if the **–l** modifier is specified, the upper case key letter is followed by a *

GLOBAL the key letter is upper case.

–P Write information in a portable output format, as specified in **Standard Output**.

–r Prepend the name of the object file or archive to each output line.

–R Print the archive name (if present), followed by the object file and symbol name. If the –**r** option is also specified, this option is ignored.

–s Print section name instead of section index.

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

d The offset is written in decimal (default).
o The offset is written in octal.
x The offset is written in hexadecimal.

–T See **NOTES** below.

/usr/ccs/bin/nm **–u** Print undefined symbols only.
/usr/xpg4/bin/nm **–u** Print long listing for each undefined symbol. See **OUTPUT** below.

–v Sort external symbols by value before they are printed.

–V Print the version of the **nm** command executing on the standard error output.

–x Print the value and size of a symbol in hexadecimal instead of decimal (equivalent to –**t x**).

Options may be used in any order, either singly or in combination, and may appear anywhere in the command line. When conflicting options are specified (such as –**v** and –**n**; and –**o** and –**x**) the first is taken and the second ignored with a warning message to the user. (See –**R** for exception.)

OPERANDS

The following operand is supported:

file A path name of an object file, executable file or object-file library.

OUTPUT

Standard Output

For each symbol, the following information will be printed:

Index The index of the symbol. (The index appears in brackets.)

Value The value of the symbol is one of the following:

- a section offset for defined symbols in a relocatable file
- alignment constraints for symbols whose section index is **SHN_COMMON**
- a virtual address in executable and dynamic library files.

Size The size in bytes of the associated object.

Type A symbol is of one of the following types:

NOTYPE no type was specified
OBJECT a data object such as an array or variable
FUNC a function or other executable code
SECTION a section symbol

FILE name of the source file.

Bind The symbol's binding attributes.

LOCAL symbols have a scope limited to the object file containing their definition

GLOBAL symbols are visible to all object files being combined

WEAK symbols are essentially global symbols with a lower precedence than **GLOBAL**.

Other A field reserved for future use, currently containing **0**.

Shndx Except for three special values, this is the section header table index in relation to which the symbol is defined. The following special values exist:

ABS indicates the symbol's value will not change through relocation

COMMON indicates an unallocated block and the value provides alignment constraints

UNDEF indicates an undefined symbol.

Name The name of the symbol

Object Name
The name of the object or library if −**A** is specified.

If the −**P** option is specified, the previous information is 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*> is formatted as follows:

- If −**A** is not specified, <*library/object name*> is 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 library. In this case, <*object file*> names 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** will write a line identifying the object containing the following symbols 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*> is the name of the file in the library containing the following symbols:

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

If –**P** is specified, but –**t** is not, the format is as if –**t x** had been specified.

ENVIRONMENT

See **environ**(5) for descriptions of the following environment variables that affect the execution of **nm**: **LC_COLLATE**, **LC_CTYPE**, **LC_MESSAGES**, and **NLSPATH**.

EXIT STATUS

The following exit values are returned:

0 Successful completion.
>0 An error occurred.

SEE ALSO

ar(1), **as**(1), **dump**(1), **ld**(1), **a.out**(4), **ar**(4), **environ**(5)

NOTES

The following options are obsolete because of changes to the object file format and will be deleted in a future release.

–**e** Print only external and static symbols. The symbol table now contains only static and external symbols. Automatic symbols no longer appear in the symbol table. They do appear in the debugging information produced by **cc** –**g**, which may be examined using **dump**(1).

–**f** Produce full output. Redundant symbols (such as .text, .data, and so forth). which existed previously do not exist and producing full output will be identical to the default output.

–**T** By default, **nm** prints the entire name of the symbols listed. Since symbol names have been moved to the last column, the problem of overflow is removed and it is no longer necessary to truncate the symbol name.