NAME

llvm-nm - list LLVM bitcode and object file’s symbol table

SYNOPSIS

llvm-nm [*options*] [*filenames...*]

DESCRIPTION

The **llvm-nm** utility lists the names of symbols from LLVM bitcode files, object files, and archives. Each symbol is listed along with some simple information about its provenance. If no filename is specified, *a.out* is used as the input. If - is used as a filename, **llvm-nm** will read a file from its standard input stream.

llvm-nm’s default output format is the traditional BSD **nm** output format. Each such output record consists of an (optional) 8-digit hexadecimal address, followed by a type code character, followed by a name, for each symbol. One record is printed per line; fields are separated by spaces. When the address is omitted, it is replaced by 8 spaces.

The supported type code characters are as follows. Where both lower and upper-case characters are listed for the same meaning, a lower-case character represents a local symbol, whilst an upper-case character represents a global (external) symbol:

a, A
: Absolute symbol.

b, B
: Uninitialized data (bss) object.

C
: Common symbol. Multiple definitions link together into one definition.

d, D
: Writable data object.

i, I
: COFF: .idata symbol or symbol in a section with IMAGE_SCN_LNK_INFO set.

n
: ELF: local symbol from non-alloc section.

: COFF: debug symbol.

N
ELF: debug section symbol, or global symbol from non-alloc section.

s, S
COFF: section symbol.

Mach-O: absolute symbol or symbol from a section other than __TEXT_EXEC __text, __TEXT __text, __DATA __data, or __DATA __bss.

r, R
Read-only data object.

t, T
Code (text) object.

u
ELF: GNU unique symbol.

U
Named object is undefined in this file.

v
ELF: Undefined weak object. It is not a link failure if the object is not defined.

V
ELF: Defined weak object symbol. This definition will only be used if no regular definitions exist in a link. If multiple weak definitions and no regular definitions exist, one of the weak definitions will be used.

w
Undefined weak symbol other than an ELF object symbol. It is not a link failure if the symbol is not defined.

W
Defined weak symbol other than an ELF object symbol. This definition will only be used if no regular definitions exist in a link. If multiple weak definitions and no regular definitions exist, one of the weak definitions will be used.

-
Mach-O: N_STAB symbol.

?
Something unrecognizable.

Because LLVM bitcode files typically contain objects that are not considered to have addresses until they are linked into an executable image or dynamically compiled "just-in-time", **llvm-nm** does not print an address for any symbol in an LLVM bitcode file, even symbols which are defined in the bitcode file.

OPTIONS

-B Use BSD output format. Alias for **--format=bsd**.

-X Specify the type of XCOFF object file, ELF object file, or IR object file input from command line or from archive files that llvm-nm should examine. The mode must be one of the following:

32 Process only 32-bit object files.

64 Process only 64-bit object files.

32_64
Process both 32-bit and 64-bit object files.

any Process all the supported object files.

--debug-syms, -a
Show all symbols, even those usually suppressed.

--defined-only, -U
Print only symbols defined in this file.

--demangle, -C
Demangle symbol names.

--dynamic, -D
Display dynamic symbols instead of normal symbols.

--export-symbols
Print sorted symbols with their visibility (if applicable), with duplicates removed.

--extern-only, -g

Print only symbols whose definitions are external; that is, accessible from other files.

--format=<format>, -f
Select an output format; *format* may be *sysv*, *posix*, *darwin*, *bsd* or *just-symbols*. The default is *bsd*.

--help, -h
Print a summary of command-line options and their meanings.

-j Print just the symbol names. Alias for *--format=just-symbols*‘.

-m Use Darwin format. Alias for **--format=darwin**.

--no-demangle
Don’t demangle symbol names. This is the default.

--no-llvm-bc
Disable the LLVM bitcode reader.

--no-sort, -p
Show symbols in the order encountered.

--no-weak, -W
Don’t print weak symbols.

--numeric-sort, -n, -v
Sort symbols by address.

--portability, -P
Use POSIX.2 output format. Alias for **--format=posix**.

--print-armap
Print the archive symbol table, in addition to the symbols.

--print-file-name, -A, -o
Precede each symbol with the file it came from.

--print-size, -S
Show symbol size as well as address (not applicable for Mach-O).

--quiet
Suppress ’no symbols’ diagnostic.

--radix=<RADIX>, -t
Specify the radix of the symbol address(es). Values accepted are *d* (decimal), *x* (hexadecimal) and *o* (octal).

--reverse-sort, -r
Sort symbols in reverse order.

--size-sort
Sort symbols by size.

--special-syms
Do not filter special symbols from the output.

--undefined-only, -u
Print only undefined symbols.

--version, -V
Display the version of the **llvm-nm** executable, then exit. Does not stack with other commands.

@<FILE>
Read command-line options from response file <*FILE*>.

MACH-O SPECIFIC OPTIONS

--add-dyldinfo
Add symbols from the dyldinfo, if they are not already in the symbol table. This is the default.

--add-inlinedinfo
Add symbols from the inlined libraries, TBD file inputs only.

--arch=<arch1[,arch2,...]>
Dump the symbols from the specified architecture(s).

--dyldinfo-only
Dump only symbols from the dyldinfo.

--no-dyldinfo

Do not add any symbols from the dyldinfo.

-s <section>
Dump only symbols from this segment and section name.

-x Print symbol entry in hex.

XCOFF SPECIFIC OPTIONS

--no-rsrc
Exclude resource file symbols (**__rsrc**) from export symbol list.

BUGS

- **llvm-nm** does not support the full set of arguments that GNU **nm** does.

EXIT STATUS

llvm-nm exits with an exit code of zero.

SEE ALSO

llvm-ar(1), **llvm-objdump(1)**, **llvm-readelf(1)**, **llvm-readobj(1)**

AUTHOR

Maintained by the LLVM Team (https://llvm.org/).

COPYRIGHT