

NAME

`dsymutil` - manipulate archived DWARF debug symbol files

SYNOPSIS

`dsymutil` [*options*] *executable*

DESCRIPTION

`dsymutil` links the DWARF debug information found in the object files for an executable *executable* by using debug symbols information contained in its symbol table. By default, the linked debug information is placed in a **.dSYM** bundle with the same name as the executable.

OPTIONS**--accelerator=<accelerator type>**

Specify the desired type of accelerator table. Valid options are 'Apple', 'Dwarf', 'Default' and 'None'.

--arch <arch>

Link DWARF debug information only for specified CPU architecture types. Architectures may be specified by name. When using this option, an error will be returned if any architectures can not be properly linked. This option can be specified multiple times, once for each desired architecture. All CPU architectures will be linked by default and any architectures that can't be properly linked will cause `dsymutil` to return an error.

--dump-debug-map

Dump the *executable*'s debug-map (the list of the object files containing the debug information) in YAML format and exit. No DWARF link will take place.

--flat, -f

Produce a flat dSYM file. A **.dwarf** extension will be appended to the executable name unless the output file is specified using the **-o** option.

--gen-reproducer

Generate a reproducer consisting of the input object files. Alias for `--reproducer=GenerateOnExit`.

--help, -h

Print this help output.

--keep-function-for-static

Make a static variable keep the enclosing function even if it would have been omitted otherwise.

--minimize, -z

When used when creating a dSYM file, this option will suppress the emission of the `.debug_inlines`, `.debug_pubnames`, and `.debug_pubtypes` sections since dsymutil currently has better equivalents: `.apple_names` and `.apple_types`. When used in conjunction with **--update** option, this option will cause redundant accelerator tables to be removed.

--no-odr

Do not use ODR (One Definition Rule) for uniquing C++ types.

--no-output

Do the link in memory, but do not emit the result file.

--no-swiftmodule-timestamp

Don't check the timestamp for swiftmodule files.

--num-threads <threads>, -j <threads>

Specifies the maximum number (**n**) of simultaneous threads to use when linking multiple architectures.

--object-prefix-map <prefix=remapped>

Remap object file paths (but no source paths) before processing. Use this for Clang objects where the module cache location was remapped using **-fdebug-prefix-map**; to help dsymutil find the Clang module cache.

--oso-prepend-path <path>

Specifies a **path** to prepend to all debug symbol object file paths.

--out <filename>, -o <filename>

Specifies an alternate **path** to place the dSYM bundle. The default dSYM bundle path is created by appending `.dSYM` to the executable name.

--papertrail

When running dsymutil as part of your build system, it can be desirable for warnings to be part of the end product, rather than just being emitted to the output stream. When enabled warnings are embedded in the linked DWARF debug information.

--remarks-output-format <format>

Specify the format to be used when serializing the linked remarks.

--remarks-prepend-path <path>

Specify a directory to prepend the paths of the external remark files.

--reproducer <mode>

Specify the reproducer generation mode. Valid options are 'GenerateOnExit', 'GenerateOnCrash', 'Use', 'Off'.

--statistics

Print statistics about the contribution of each object file to the linked debug info. This prints a table after linking with the object file name, the size of the debug info in the object file (in bytes) and the size contributed (in bytes) to the linked dSYM. The table is sorted by the output size listing the object files with the largest contribution first.

--symbol-map <bcsymbolmap>

Update the existing dSYMs inplace using symbol map specified.

-s, --symtab

Dumps the symbol table found in *executable* or object file(s) and exits.

-S Output textual assembly instead of a binary dSYM companion file.

--toolchain <toolchain>

Embed the toolchain in the dSYM bundle's property list.

-u, --update

Update an existing dSYM file to contain the latest accelerator tables and other DWARF optimizations. This option will rebuild the '.apple_names' and '.apple_types' hashed accelerator tables.

--use-reproducer <path>

Use the object files from the given reproducer path. Alias for --reproducer=Use.

--verbose

Display verbose information when linking.

--verify

Run the DWARF verifier on the linked DWARF debug info.

-v, --version

Display the version of the tool.

-y Treat *executable* as a YAML debug-map rather than an executable.

EXIT STATUS

dsymutil returns 0 if the DWARF debug information was linked successfully. Otherwise, it returns 1.

SEE ALSO

llvm-dwarfdump(1)

AUTHOR

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

COPYRIGHT

2003-2024, LLVM Project