

NAME

tblgen - Description to C++ Code

SYNOPSIS

clang-tblgen [*options*] [*filename*]

lldb-tblgen [*options*] [*filename*]

llvm-tblgen [*options*] [*filename*]

mlir-tblgen [*options*] [*filename*]

DESCRIPTION

***-tblgen** is a family of programs that translates target description (**.td**) files into C++ code and other output formats. Most users of LLVM will not need to use this program. It is used only for writing parts of the compiler, debugger, and LLVM target backends.

The details of the input and output of the ***-tblgen** programs is beyond the scope of this short introduction; please see the *TableGen Overview* for an introduction and for references to additional TableGen documents.

The *filename* argument specifies the name of the Target Description (**.td**) file that TableGen processes.

OPTIONS**General Options****-help**

Print a description of the command line options.

-help-list

Print a description of the command line options in a simple list format.

-D=macroname

Specify the name of a macro to be defined. The name is defined, but it has no particular value.

-d=filename

Specify the name of the dependency filename.

-debug

Enable debug output.

-dump-json

Print a JSON representation of all records, suitable for further automated processing.

-I directory

Specify where to find other target description files for inclusion. The **directory** value should be a full or partial path to a directory that contains target description files.

-null-backend

Parse the source files and build the records, but do not run any backend. This is useful for timing the frontend.

-o filename

Specify the output file name. If **filename** is -, then ***-tblgen** sends its output to standard output.

-print-records

Print all classes and records to standard output (default backend option).

-print-detailed-records

Print a detailed report of all global variables, classes, and records to standard output.

-stats

Print a report with any statistics collected by the backend.

-time-phases

Time the parser and backend phases and print a report.

-version

Show the version number of the program.

-write-if-changed

Write the output file only if it is new or has changed.

clang-tblgen Options**-gen-clang-attr-classes**

Generate Clang attribute classes.

-gen-clang-attr-parser-string-switches

Generate all parser-related attribute string switches.

-gen-clang-attr-subject-match-rules-parser-string-switches

Generate all parser-related attribute subject match rule string switches.

-gen-clang-attr-impl

Generate Clang attribute implementations.

-gen-clang-attr-list''

Generate a Clang attribute list.

-gen-clang-attr-subject-match-rule-list

Generate a Clang attribute subject match rule list.

-gen-clang-attr-pch-read

Generate Clang PCH attribute reader.

-gen-clang-attr-pch-write

Generate Clang PCH attribute writer.

-gen-clang-attr-has-attribute-impl

Generate a Clang attribute spelling list.

-gen-clang-attr-spelling-index

Generate a Clang attribute spelling index.

-gen-clang-attr-ast-visitor

Generate a recursive AST visitor for Clang attributes.

-gen-clang-attr-template-instantiate

Generate a Clang template instantiate code.

-gen-clang-attr-parsed-attr-list

Generate a Clang parsed attribute list.

-gen-clang-attr-parsed-attr-impl

Generate the Clang parsed attribute helpers.

-gen-clang-attr-parsed-attr-kinds

Generate a Clang parsed attribute kinds.

-gen-clang-attr-text-node-dump

Generate Clang attribute text node dumper.

-gen-clang-attr-node-traverse

Generate Clang attribute traverser.

-gen-clang-diags-defs

Generate Clang diagnostics definitions.

-clang-component component

Only use warnings from specified component.

-gen-clang-diag-groups

Generate Clang diagnostic groups.

-gen-clang-diags-index-name

Generate Clang diagnostic name index.

-gen-clang-basic-reader

Generate Clang BasicReader classes.

-gen-clang-basic-writer

Generate Clang BasicWriter classes.

-gen-clang-comment-nodes

Generate Clang AST comment nodes.

-gen-clang-decl-nodes

Generate Clang AST declaration nodes.

-gen-clang-stmt-nodes

Generate Clang AST statement nodes.

-gen-clang-type-nodes

Generate Clang AST type nodes.

-gen-clang-type-reader

Generate Clang AbstractTypeReader class.

-gen-clang-type-writer

Generate Clang AbstractTypeWriter class.

-gen-clang-opcodes

Generate Clang constexpr interpreter opcodes.

-gen-clang-sa-checkers

Generate Clang static analyzer checkers.

-gen-clang-comment-html-tags

Generate efficient matchers for HTML tag names that are used in documentation comments.

-gen-clang-comment-html-tags-properties

Generate efficient matchers for HTML tag properties.

-gen-clang-comment-html-named-character-references

Generate function to translate named character references to UTF-8 sequences.

-gen-clang-comment-command-info

Generate command properties for commands that are used in documentation comments.

-gen-clang-comment-command-list

Generate list of commands that are used in documentation comments.

-gen-clang-ocl-builtins

Generate OpenCL builtin declaration handlers.

-gen-arm-neon

Generate **arm_neon.h** for Clang.

-gen-arm-fp16

Generate **arm_fp16.h** for Clang.

-gen-arm-bf16

Generate **arm_bf16.h** for Clang.

-gen-arm-neon-sema

Generate ARM NEON sema support for Clang.

-gen-arm-neon-test

Generate ARM NEON tests for Clang.

-gen-arm-sve-header

Generate **arm_sve.h** for Clang.

-gen-arm-sve-builtins

Generate **arm_sve_builtins.inc** for Clang.

-gen-arm-sve-builtin-codegen

Generate **arm_sve_builtin_cg_map.inc** for Clang.

-gen-arm-sve-typeflags

Generate **arm_sve_typeflags.inc** for Clang.

-gen-arm-sve-sema-rangechecks

Generate **arm_sve_sema_rangechecks.inc** for Clang.

-gen-arm-mve-header

Generate **arm_mve.h** for Clang.

-gen-arm-mve-builtin-def

Generate ARM MVE builtin definitions for Clang.

-gen-arm-mve-builtin-sema

Generate ARM MVE builtin sema checks for Clang.

-gen-arm-mve-builtin-codegen

Generate ARM MVE builtin code-generator for Clang.

-gen-arm-mve-builtin-aliases

Generate list of valid ARM MVE builtin aliases for Clang.

-gen-arm-cde-header

Generate **arm_cde.h** for Clang.

-gen-arm-cde-builtin-def

Generate ARM CDE builtin definitions for Clang.

-gen-arm-cde-builtin-sema

Generate ARM CDE builtin sema checks for Clang.

-gen-arm-cde-builtin-codegen

Generate ARM CDE builtin code-generator for Clang.

-gen-arm-cde-builtin-aliases

Generate list of valid ARM CDE builtin aliases for Clang.

-gen-riscv-vector-header

Generate **riscv_vector.h** for Clang.

-gen-riscv-vector-builtins

Generate **riscv_vector_builtins.inc** for Clang.

-gen-riscv-vector-builtin-codegen

Generate **riscv_vector_builtin_cg.inc** for Clang.

-gen-attr-docs

Generate attribute documentation.

-gen-diag-docs

Generate diagnostic documentation.

-gen-opt-docs

Generate option documentation.

-gen-clang-data-collectors

Generate data collectors for AST nodes.

-gen-clang-test-pragma-attribute-supported-attributes

Generate a list of attributes supported by **#pragma** Clang attribute for testing purposes.

lldb-tblgen Options**gen-lldb-option-defs**

Generate lldb OptionDefinition values.

gen-lldb-property-defs

Generate lldb PropertyDefinition values.

gen-lldb-property-enum-defs

Generate lldb PropertyDefinition enum values.

llvm-tblgen Options

-gen-asm-matcher

Generate assembly instruction matcher.

-match-prefix=prefix

Make -gen-asm-matcher match only instructions with the given *prefix*.

-gen-asm-parser

Generate assembly instruction parser.

-asmparsernum=n

Make -gen-asm-parser emit assembly parser number *n*.

-gen-asm-writer

Generate assembly writer.

-asmwriternum=n

Make -gen-asm-writer emit assembly writer number *n*.

-gen-attrs

Generate attributes.

-gen-automata

Generate generic automata.

-gen-callingconv

Generate calling convention descriptions.

-gen-compress-inst-emitter

Generate RISC-V compressed instructions.

-gen-ctags

Generate ctags-compatible index.

-gen-dag-isel

Generate a DAG (directed acyclic graph) instruction selector.

-instrument-coverage

Make -gen-dag-isel generate tables to help identify the patterns matched.

-omit-comments

Make -gen-dag-isel omit comments. The default is false.

-gen-dfa-packetizer

Generate DFA Packetizer for VLIW targets.

-gen-directive-decl

Generate directive related declaration code (header file).

-gen-directive-gen

Generate directive related implementation code part.

-gen-directive-impl

Generate directive related implementation code.

-gen-disassembler

Generate disassembler.

-gen-emitter

Generate machine code emitter.

-gen-exegesis

Generate llvm-exegesis tables.

-gen-fast-isel

Generate a "fast" instruction selector.

-gen-global-isel

Generate GlobalISel selector.

-gisel-coverage-file=filename

Specify the file from which to retrieve coverage information.

-instrument-gisel-coverage

Make -gen-global-isel generate coverage instrumentation.

-optimize-match-table

Make -gen-global-isel generate an optimized version of the match table.

-warn-on-skipped-patterns

Make -gen-global-isel explain why a pattern was skipped for inclusion.

-gen-global-isel-combiner

Generate GlobalISel combiner.

-combiners=list

Make -gen-global-isel-combiner emit the specified combiners.

-gicombiner-show-expansions

Make -gen-global-isel-combiner use C++ comments to indicate occurrences of code expansion.

-gicombiner-stop-after-build

Make -gen-global-isel-combiner stop processing after building the match tree.

-gicombiner-stop-after-parse

Make -gen-global-isel-combiner stop processing after parsing rules and dump state.

-gen-instr-info

Generate instruction descriptions.

-gen-instr-docs

Generate instruction documentation.

-gen-intrinsic-enums

Generate intrinsic enums.

-intrinsic-prefix=prefix

Make -gen-intrinsic-enums generate intrinsics with this target *prefix*.

-gen-intrinsic-impl

Generate intrinsic information.

-gen-opt-parser-defs

Generate options definitions.

-gen-opt-rst

Generate option RST.

-gen-pseudo-lowering

Generate pseudo instruction lowering.

-gen-register-bank

Generate register bank descriptions.

-gen-register-info

Generate registers and register classes info.

-register-info-debug

Make -gen-register-info dump register information for debugging.

-gen-searchable-tables

Generate generic searchable tables. See *TableGen BackEnds* for a detailed description.

-gen-subtarget

Generate subtarget enumerations.

-gen-x86-EVEX2VEX-tables

Generate X86 EVEX to VEX compress tables.

-gen-x86-fold-tables

Generate X86 fold tables.

-long-string-literals

When emitting large string tables, prefer string literals over comma-separated char literals. This can be a readability and compile-time performance win, but upsets some compilers.

-print-enums

Print enumeration values for a class.

-class=classname

Make -print-enums print the enumeration list for the specified class.

-print-sets

Print expanded sets for testing DAG exprs.

mlir-tblgen Options

-gen-avail-interface-decls

Generate availability interface declarations.

-gen-avail-interface-defs

Generate op interface definitions.

-gen-dialect-doc

Generate dialect documentation.

-dialect

The dialect to generate.

-gen-directive-decl

Generate declarations for directives (OpenMP, etc.).

-gen-enum-decls

Generate enum utility declarations.

-gen-enum-defs

Generate enum utility definitions.

-gen-enum-from-llvmir-conversions

Generate conversions of EnumAttrs from LLVM IR.

-gen-enum-to-llvmir-conversions

Generate conversions of EnumAttrs to LLVM IR.

-gen-llvmir-conversions

Generate LLVM IR conversions.

-gen-llvmir-intrinsics

Generate LLVM IR intrinsics.

-llvmir-intrinsics-filter

Only keep the intrinsics with the specified substring in their record name.

-dialect-opclass-base

The base class for the ops in the dialect we are to emit.

-gen-op-decls

Generate operation declarations.

-gen-op-defs

Generate operation definitions.

-asmformat-error-is-fatal

Emit a fatal error if format parsing fails.

-op-exclude-regex

Regular expression of name of ops to exclude (no filter if empty).

-op-include-regex

Regular expression of name of ops to include (no filter if empty).

-gen-op-doc

Generate operation documentation.

-gen-pass-decls

Generate operation documentation.

-name namestring

The name of this group of passes.

-gen-pass-doc

Generate pass documentation.

-gen-rewriters

Generate pattern rewriters.

-gen-spirv-avail-impls

Generate SPIR-V operation utility definitions.

-gen-spirv-capability-implication

Generate utility function to return implied capabilities for a given capability.

-gen-spirv-enum-avail-decls

Generate SPIR-V enum availability declarations.

-gen-spirv-enum-avail-defs

Generate SPIR-V enum availability definitions.

-gen-spirv-op-utils

Generate SPIR-V operation utility definitions.

-gen-spirv-serialization

Generate SPIR-V (de)serialization utilities and functions.

-gen-struct-attr-decls

Generate struct utility declarations.

-gen-struct-attr-defs

Generate struct utility definitions.

-gen-typedef-decls

Generate TypeDef declarations.

-gen-typedef-defs

Generate TypeDef definitions.

-typedefs-dialect name

Generate types for this dialect.

EXIT STATUS

If ***-tblgen** succeeds, it will exit with 0. Otherwise, if an error occurs, it will exit with a non-zero value.

AUTHOR

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

COPYRIGHT

2003-2023, LLVM Project