NAME

v8 −− Google's JavaScript engine

SYNOPSIS

v8 [*flags*] [**−−help** | **−−shell**] [**−e** *string*] [*file*...]
d8 [*flags*] [**−−help** | **−−shell**] [*file*...]

OPTIONS

−e *string*
Execute *string* in V8.

−−shell
Run an interactive JavaScript shell.

−−help
Print flags and usage message to stdout, then exit.

V8 FLAGS

Option names beginning with "**no-**" pertain to a boolean V8 flag enabled by default.

Harmony Features

−−harmony
Enable all completed harmony features.

−−harmony_object_observe
Enable `Object.observe`.

−−harmony_function_sent
Enable `function.sent`.

−−harmony_sharedarraybuffer
In progress.

Enable `SharedArrayBuffer`.

−−harmony_simd
In progress.

Enable `SIMD`.

−−harmony_do_expressions
In progress.

Enable `do` expressions.

−−harmony_regexp_property
In progress.

Enable RegExp Unicode property classes.

--harmony_string_padding
In progress.

Enable String-padding methods.

--harmony_regexp_lookbehind
Enable RegExp lookbehinds.

--harmony_tailcalls
Enable tail-call optimisation.

--harmony_object_values_entries
Enable `Object.values` and `Object.entries`.

--harmony_object_own_property_descriptors
Enable `Object.getOwnPropertyDescriptors()`.

--harmony_exponentiation_operator
Enable the exponentiation operator `**`.

--no-harmony_array_prototype_values
Disable `Array.prototype.values`.

--no-harmony_function_name
Disable function name inference.

--no-harmony_instanceof
Disable `instanceof` support.

--no-harmony_iterator_close
Disable iterator finalisation.

--no-harmony_unicode_regexps
Disable Unicode `/u` modifiers in regular expressions.

--no-harmony_regexp_exec
Disable RegExp `exec` override behaviour.

--no-harmony_sloppy
Disable harmony features in sloppy mode.

--no-harmony_sloppy_let
Disable `let` in sloppy mode.

--no-harmony_sloppy_function
Disable block-scoping in sloppy mode.

--no-harmony_regexp_subclass
Disable RegExp subclassing.

--no-harmony_restrictive_declarations
Disable limitations on sloppy mode function declarations.

--no-harmony_species
Disable `Symbol.species`.

Ignition

--ignition
Use ignition interpreter.

--no-ignition_eager
Disable eager compilation and parsing with ignition.

--ignition_filter
Type: String
Default: `"*"`

Filter for ignition interpreter.

--print_bytecode
Print bytecode generated by ignition interpreter.

--trace_ignition
Trace the bytecodes executed by the ignition interpreter.

--trace_ignition_codegen
Trace the codegen of ignition interpreter bytecode handlers.

TurboFan Compiler

--turbo
Enable TurboFan compiler.

--no-turbo_shipping
Disable TurboFan compiler on subset.

--turbo_greedy_regalloc
Use the greedy register allocator.

--turbo_sp_frame_access
Use stack pointer-relative access to frame wherever possible.

--no-turbo_preprocess_ranges
Disable pre-register allocation heuristics.

--no-turbo_loop_stackcheck
Disable stack checks in loops.

--turbo_filter
Type: String
Default: `"~~"`

Optimisation filter for TurboFan compiler.

--trace_turbo
Trace generated TurboFan IR.

--trace_turbo_graph
Trace generated TurboFan graphs.

--trace_turbo_cfg_file
Type: String
Default: `NULL`

Trace turbo cfg graph (for C1 visualiser) to a given filename.

--no-trace_turbo_types
Don't trace TurboFan's types.

--trace_turbo_scheduler
Trace TurboFan's scheduler.

--trace_turbo_reduction
Trace TurboFan's various reducers.

--trace_turbo_jt
Trace TurboFan's jump threading.

--trace_turbo_ceq
Trace TurboFan's control equivalence.

--no-turbo_asm
Disable TurboFan for asm.js code.

--turbo_asm_deoptimization
Enable deoptimisation in TurboFan for asm.js code.

--turbo_verify
Verify TurboFan graphs at each phase.

−−turbo_stats
Print TurboFan statistics.

−−no-turbo_splitting
Don’t split nodes during scheduling in TurboFan.

−−no-turbo_types
Disable typed lowering in TurboFan.

−−turbo_source_positions
Track source code positions when building TurboFan IR.

−−function_context_specialization
Enable function context specialisation in TurboFan.

−−no-native_context_specialization
Disable native context specialisation in TurboFan.

−−no-turbo_inlining
Disable inlining in TurboFan.

−−trace_turbo_inlining
Trace TurboFan inlining.

−−no-loop_assignment_analysis
Disable loop assignment analysis.

−−turbo_profiling
Enable profiling in TurboFan.

−−turbo_verify_allocation
Verify register allocation in TurboFan.

−−no-turbo_move_optimization
Don’t optimise gap moves in TurboFan.

−−no-turbo_jt
Disable jump threading in TurboFan.

−−no-turbo_osr
Disable OSR in TurboFan.

−−turbo_stress_loop_peeling
Stress loop peeling optimisation.

−−no-turbo_cf_optimization
Don’t optimise control flow in TurboFan.

--no-turbo_frame_elision
Don't elide frames in TurboFan.

--no-turbo_cache_shared_code
Don't cache context-independent code.

--turbo_preserve_shared_code
Keep context-independent code.

--turbo_escape
Enable escape analysis.

--turbo_instruction_scheduling
Enable instruction scheduling in TurboFan.

--turbo_stress_instruction_scheduling
Randomly schedule instructions to stress dependency tracking.

WebAssembly

--expose_wasm
Expose WASM interface to JavaScript.

--trace_wasm_encoder
Trace encoding of WASM code.

--trace_wasm_decoder
Trace decoding of WASM code.

--trace_wasm_decode_time
Trace decoding time of WASM code.

--trace_wasm_compiler
Trace compiling of WASM code.

--trace_wasm_ast_start
Type: Int
Default: `0`

Start function for WASM AST trace (inclusive).

--trace_wasm_ast_end
Type: Int
Default: `0`

End function for WASM AST trace (exclusive).

--skip_compiling_wasm_funcs
Type: Int
Default: `0`

Start compiling at function `N`

--wasm_break_on_decoder_error
Debug break when WASM decoder encounters an error.

--no-wasm_loop_assignment_analysis
Disable loop assignment analysis for WASM.

--enable_simd_asmjs
Enable SIMD.js in asm.js stdlib.

--dump_wasm_module
Dump WASM module bytes.

--dump_wasm_module_path
Type: String
Default: `NULL`

Directory to dump WASM modules to.

Other

--experimental_extras
Enable code compiled in via `v8_experimental_extra_library_files`.

--use_strict
Enforce strict mode.

--es_staging
Internal use only.

Enable test-worthy harmony features.

--compiled_keyed_generic_loads
Use optimising compiler to generate keyed generic load stubs.

--no-allocation_site_pretenuring
Disable pretenuring with allocation sites.

--trace_pretenuring
Trace pretenuring decisions of HAllocate instructions.

--trace_pretenuring_statistics
Trace allocation site pretenuring statistics.

--no-track_fields
Disable tracking of fields with only SMI values.

--no-track_double_fields
Disable tracking of fields with double values.

--no-track_heap_object_fields
Disable tracking of fields with heap values.

--no-track_computed_fields
Disable tracking of computed boilerplate fields.

--no-harmony_instanceof_opt
Disable optimisation for ES6 `instanceof` support.

--no-track_field_types
Disable tracking of field types.

--no-smi_binop
Disable support for SMI representation in binary operations.

--optimize_for_size
Enable optimisations which favour memory size over execution speed.

--no-unbox_double_arrays
Don't automatically unbox arrays of doubles.

--no-string_slices
Disable string slices.

--no-crankshaft
Disable Crankshaft.

--hydrogen_filter
Type: String
Default: `"*"`

Optimisation filter.

--no-use_gvn
Disable hydrogen global value numbering.

--gvn_iterations
Type: Int
Default: `3`

Maximum number of GVN fix-point iterations.

--no-use_canonicalizing
Disable hydrogen instruction canonicalising.

--no-use_inlining
Disable function inlining.

--no-use_escape_analysis
Disable hydrogen escape analysis.

--no-use_allocation_folding
Disable allocation folding.

--use_local_allocation_folding
Only fold in basic blocks.

--no-use_write_barrier_elimination
Don't eliminate write barriers targeting allocations in optimised code.

--max_inlining_levels
Type: Int
Default: `5`

Maximum number of inlining levels.

--max_inlined_source_size
Type: Int
Default: `600`

Maximum source size (in bytes) considered for a single inlining.

--max_inlined_nodes
Type: Int
Default: `196`

Maximum number of AST nodes considered for a single inlining.

--max_inlined_nodes_cumulative
Type: Int
Default: `400`

Maximum cumulative number of AST nodes considered for inlining.

--no-loop_invariant_code_motion
Disable loop invariant code motion.

--no-fast_math
Disable faster (but potentially less accurate) math functions.

--collect_megamorphic_maps_from_stub_cache
Allow Crankshaft to harvest type feedback from stub cache.

--hydrogen_stats
Print statistics for hydrogen.

--trace_check_elimination
Trace check elimination phase.

--trace_environment_liveness
Trace liveness of local variable slots.

--trace_hydrogen
Trace generated hydrogen to file.

--trace_hydrogen_filter
Type: String
Default: `"*"`

Hydrogen tracing filter.

--trace_hydrogen_stubs
Trace generated hydrogen for stubs.

--trace_hydrogen_file
Type: String
Default: `NULL`

Trace hydrogen to given filename.

--trace_phase
Type: String
Default: `"HLZ"`

Trace generated IR for specified phases.

--trace_inlining
Trace inlining decisions.

--trace_load_elimination
Trace load elimination.

--trace_store_elimination
Trace store elimination.

--trace_alloc
Trace register allocator.

--trace_all_uses
Trace all use positions.

--trace_range
Trace range analysis.

--trace_gvn
Trace global value numbering.

--trace_representation
Trace representation types.

--trace_removable_simulates
Trace removable simulates.

--trace_escape_analysis
Trace hydrogen escape analysis.

--trace_allocation_folding
Trace allocation folding.

--trace_track_allocation_sites
Trace the tracking of allocation sites.

--trace_migration
Trace object migration.

--trace_generalization
Trace map generalization.

--stress_pointer_maps
Stress pointer map for every instruction.

--stress_environments
Stress environment for every instruction.

--deopt_every_n_times
Type: Int
Default: `0`

Deoptimise every `n` times a deopt point is passed.

--deopt_every_n_garbage_collections
Type: Int
Default: `0`

Deoptimise every `n` garbage collections.

--print_deopt_stress
Print number of possible deopt points.

--trap_on_deopt
Put a breakpoint before deoptimising.

--trap_on_stub_deopt
Put a breakpoint before deoptimising a stub.

--no-deoptimize_uncommon_cases
Don't deoptimise uncommon cases.

--no-polymorphic_inlining
Disable polymorphic inlining.

--no-use_osr
Disable on-stack replacement.

--no-array_bounds_checks_elimination
Disable array bounds checks elimination (BCE).

--trace_bce
Trace array bounds check elimination.

--array_index_dehoisting
Perform array index dehoisting.

--no-analyze_environment_liveness
Don't analyse liveness of environment slots and zap dead values.

--no-load_elimination
Disable load elimination.

--no-check_elimination
Disable check elimination.

--store_elimination
Use store elimination.

--no-dead_code_elimination
Disable elimination of dead code.

--no-fold_constants
Disable constant folding.

--trace_dead_code_elimination
Trace elimination of dead code.

--no-unreachable_code_elimination
Disable elimination of unreachable code.

--trace_osr
Trace on-stack replacement.

--stress_runs
Type: Int
Default: `0`

Number of stress runs.

--no-lookup_sample_by_shared
When picking a function to optimise, don't watch for shared function info instead of JSFunction itself.

--flush_optimized_code_cache
Flush the cache of optimised code for closures on every GC.

--no-inline_construct
Disable inline constructor calls.

--no-inline_arguments
Don't inline functions with arguments object.

--no-inline_accessors
Disable inline JavaScript accessors.

--escape_analysis_iterations
Type: Int
Default: `2`

Maximum number of escape analysis fix-point iterations.

--no-concurrent_recompilation
Don't optimise hot functions asynchronously on separate threads.

--trace_concurrent_recompilation
Track concurrent recompilation.

--concurrent_recompilation_queue_length
Type: Int
Default: `8`

Set length of the concurrent compilation queue.

--concurrent_recompilation_delay
Type: Int
Default: `0`

Artificial compilation delay in milliseconds.

--block_concurrent_recompilation
Block queued jobs until released.

--no-omit_map_checks_for_leaf_maps
Do not emit check maps for constant values that have a leaf map.

Disables deoptimisation of optimised code when the layout of the maps changes.

--typed_array_max_size_in_heap
Type: Int
Default: `64`

Threshold for in-heap typed array.

--frame_count
Type: Int
Default: `1`

Number of stack frames inspected by the profiler.

--interrupt_budget
Type: Int
Default: `6144`

Execution budget before interrupt is triggered.

--type_info_threshold
Type: Int
Default: `25`

Percentage of ICs that must have type info to allow optimisation.

--generic_ic_threshold
Type: Int
Default: `30`

Maximum percentage of megamorphic/generic ICs to allow optimisation.

--self_opt_count
Type: Int
Default: `130`

Call count before self-optimisation.

--trace_opt_verbose
Enable extra-verbose compilation tracing.

--debug_code
Generate extra code (assertions) for debugging.

--code_comments
Emit comments in code disassembly.

--no-enable_sse3
Disable use of SSE3 instructions.

--no-enable_sse4_1
Disable use of SSE4.1 instructions.

--no-enable_sahf
Disable use of SAHF instruction. Only relevant to X64.

--no-enable_avx
Disable use of AVX instructions.

--no-enable_fma3
Disable use of FMA3 instructions.

--no-enable_bmi1
Disable use of BMI1 instructions.

--no-enable_bmi2
Disable use of BMI2 instructions.

--no-enable_lzcnt
Disable use of LZCNT instructions.

--no-enable_popcnt
Disable use of POPCNT instructions.

--no-enable_vfp3
Disable use of VFP3 instructions.

--no-enable_armv7
Disable use of ARMv7 instructions. Only relevant to ARM.

--no-enable_armv8
Disable use of ARMv8 instructions. Only relevant to ARM 32-bit.

--no-enable_neon
Disable use of NEON instructions. Only relevant to ARM.

--no-enable_sudiv
Disable use of SDIV and UDIV instructions. Only relevant to ARM.

--no-enable_mls
Disable use of MLS instructions. Only relevant to ARM.

--enable_movw_movt
Enable loading 32-bit constant by means of `movw` / `movt` instruction pairs. ARM only.

--no-enable_unaligned_accesses
Disable unaligned accesses for ARMv7. Only relevant to ARM.

--no-enable_32dregs
Disable use of d16-d31 registers on ARM. Irrelevant without VFP3.

--enable_vldr_imm
Enable use of constant pools for double immediate. ARM only.

--force_long_branches
Force all emitted branches to be in long mode. MIPS/PPC only.

--mcpu
Type: String
Default: `auto`

Enable optimisation for specific CPU.

--expose_natives_as
Type: String
Default: `NULL`

Expose natives in `global` object.

--expose_debug_as
Type: String
Default: `NULL`

Expose debug in `global` object.

--expose_free_buffer
Expose freeBuffer extension.

--expose_gc
Expose gc extension.

--expose_gc_as
Type: String
Default: `NULL`

Expose gc extension under the specified name.

--expose_externalize_string
Expose externalise string extension.

--expose_trigger_failure
Expose trigger-failure extension.

--stack_trace_limit
Type: Int
Default: `10`

Number of stack frames to capture.

--builtins_in_stack_traces
Show built-in functions in stack traces.

--disable_native_files
Disable builtin natives files.

--no-inline_new
Disable fast inline allocation.

--trace_codegen
Print name of functions for which code is generated.

--trace
Trace function calls.

--no-mask_constants_with_cookie
Don't use random JIT cookie to mask large constants.

--no-lazy
Don't use lazy compilation.

--trace_opt
Trace lazy optimisation.

--trace_opt_stats
Trace lazy optimisation statistics.

--no-opt
Don't use adaptive optimisations.

--always_opt
Always try to optimise functions.

--always_osr
Always try to OSR functions.

--prepare_always_opt
Prepare for turning on always opt.

--trace_deopt
Trace optimise function deoptimisation.

--trace_stub_failures
Trace deoptimisation of generated code stubs.

--no-serialize_toplevel
Disable caching of toplevel scripts.

--serialize_eager
Compile eagerly when caching scripts.

--serialize_age_code
Pre age code in the code cache.

--trace_serializer
Print code serialiser trace.

--min_preparse_length
Type: Int
Default: `1024`

Minimum length for automatic enable preparsing.

--max_opt_count
Type: Int
Default: `10`

Maximum number of optimisation attempts before giving up.

--no-compilation_cache
Disable compilation cache.

--no-cache_prototype_transitions
Don't cache prototype transitions.

--cpu_profiler_sampling_interval
Type: Int
Default: `1000`

CPU profiler sampling interval in microseconds.

--trace_js_array_abuse
Trace out-of-bounds accesses to JavaScript arrays.

--trace_external_array_abuse
Trace out-of-bounds-accesses to external arrays.

--trace_array_abuse
Trace out-of-bounds accesses to all arrays.

--trace_debug_json
Trace debugging JSON request/response.

--no-enable_liveedit
Disable liveedit experimental feature.

--no-hard_abort
Disable aborting by crashing.

--stack_size
Type: Int
Default: `984`

Default size of stack region V8 is allowed to use (in kilobytes).

--max_stack_trace_source_length
Type: Int
Default: `300`

Maximum length of function source code printed in a stack trace.

--always_inline_smi_code
Always inline SMI code in non-opt code.

--verify_operand_stack_depth
Emit debug code that verifies the static tracking of the operand stack depth.

--min_semi_space_size
Type: Int
Default: `0`

Minimum size of a semi-space in megabytes. The new space consists of twosemi-spaces.

--max_semi_space_size
Type: Int
Default: `0`

Maximum size of a semi-space in megabytes. The new space consists of twosemi-spaces.

--semi_space_growth_factor
Type: Int
Default: `2`

Factor by which to grow the new space.

--experimental_new_space_growth_heuristic
Grow the new space based on the percentage of survivors instead of their absolute value.

--max_old_space_size
Type: Int
Default: `0`

Maximum size of the old space in megabytes.

--initial_old_space_size
Type: Int
Default: `0`

Initial old space size in megabytes.

--max_executable_size
Type: Int
Default: `0`

Maximum size of executable memory in megabytes.

--gc_global
Always perform global GCs.

--gc_interval
Type: Int
Default: `-1`

Garbage collect after `n` allocations.

--retain_maps_for_n_gc
Type: Int
Default: `2`

Keep maps alive for `n` old space garbage collections.

--trace_gc
Print one trace line following each garbage collection.

--trace_gc_nvp
Print one detailed trace line in `name=value` format after each garbage collection.

--trace_gc_ignore_scavenger
Don't print trace line after scavenger collection.

--trace_idle_notification
Print one trace line following each idle notification.

--trace_idle_notification_verbose
Print the heap state used by the idle notification.

--print_cumulative_gc_stat
Print cumulative GC statistics in `name=value` format on exit.

--print_max_heap_committed
Print statistics of the maximum memory committed for the heap in `name=value` format on exit.

--trace_gc_verbose
Print more details following each garbage collection.

--trace_allocation_stack_interval
Type: Int
Default: `-1`

Print stack trace after `n` free-list allocations.

--trace_fragmentation
Report fragmentation for old space.

--trace_fragmentation_verbose
Report detailed fragmentation for old space.

--trace_evacuation
Report evacuation statistics.

--trace_mutator_utilization
Print mutator utilisation, allocation speed, gc speed.

--no-weak_embedded_maps_in_optimized_code
Don't make maps embedded in optimised code weak.

--no-weak_embedded_objects_in_optimized_code
Don't make objects embedded in optimized code weak.

--no-flush_code
Don't flush code that's not expected to be reused.

--trace_code_flushing
Trace code flushing progress.

--no-age_code
Don't track un-executed functions to age code.

--no-incremental_marking
Disable incremental marking.

--min_progress_during_incremental_marking_finalization
Type: Int
Default: `32`

Keep finalising incremental marking as long as `n` unmarked objects (or more) are discovered.

--max_incremental_marking_finalization_rounds
Type: Int
Default: `3`

Maximum number of attempts to finalise incremental marking.

--no-black_allocation
Disable black allocation.

--no-concurrent_sweeping
Disable concurrent sweeping.

--no-parallel_compaction
Disable parallel compaction.

--no-parallel_pointer_update
Disable parallel pointer update during compaction.

--trace_incremental_marking
Trace progress of the incremental marking.

--track_gc_object_stats
Track object counts and memory usage.

--trace_gc_object_stats
Trace object counts and memory usage.

--no-track_detached_contexts
Don't track native contexts that are expected to be garbage collected.

--trace_detached_contexts
Trace native contexts that are expected to be garbage collected.

--no-move_object_start
Disable moving of object starts.

--no-memory_reducer
Disable memory reducer.

--no-scavenge_reclaim_unmodified_objects
Don't remove unmodified and unreferenced objects.

--heap_growing_percent
Type: Int
Default: `0`

Specify heap growing factor as (1 + *n*/100).

--histogram_interval
Type: Int
Default: `600000`

Time interval for aggregating memory histograms (in milliseconds).

--trace_object_groups
Print object groups detected during each garbage collection.

--heap_profiler_trace_objects
Dump heap object allocations/movements/size_updates.

--sampling_heap_profiler_suppress_randomness
Use constant sample intervals to eliminate test flakiness.

--no-use_idle_notification
Don't use idle notification to reduce memory footprint.

--no-use_ic
Disable inline caching.

--trace_ic
Trace inline cache state transitions.

--native_code_counters
Generate extra code for manipulating stats counters.

--always_compact
Perform compaction on every full GC.

--never_compact
Testing only.

Never perform compaction on full GC.

--no-compact_code_space
Don't compact code space on full collections.

--no-cleanup_code_caches_at_gc
Don't flush inline caches prior to mark compact collection, or flush code caches in maps during mark compact cycle.

--no-use_marking_progress_bar
Don't use a progress bar to scan large objects in increments when incremental marking is active.

--zap_code_space
Zap free memory in code space with `0xCC` while sweeping.

--random_seed
Type: Int
Default: `0`

Default seed for initialising random generator. `0` means to use system random.

--trace_weak_arrays
Trace WeakFixedArray usage.

--no-track_prototype_users
Don't keep track of which maps refer to a given prototype object.

--trace_prototype_users
Trace updates to prototype user tracking.

--no-eliminate_prototype_chain_checks
Don't collapse prototype chain checks into single-cell checks.

--no-use_verbose_printer
Disable verbose printing.

--trace_for_in_enumerate
Trace `for-in` enumerate slow-paths.

--allow_natives_syntax
Allow natives syntax.

--trace_parse
Trace parsing and preparsing.

--trace_sim
Trace simulator execution.

--debug_sim
Enable debugging of the simulator.

--check_icache
Check icache flushes in ARM and MIPS simulator.

--stop_sim_at
Type: Int
Default: `0`

Stop simulator after `x` number of instructions.

--sim_stack_alignment
Type: Int
Default: `8`

Stack alignment (in bytes) in simulator. Either `4` or `8` (default).

--sim_stack_size
Type: Int
Default: `2048`

Stack size of the ARM64, MIPS64 and PPC64 simulator in kilobytes. Default is 2 MB.

--no-log_regs_modified
Don't print modified registers when logging register values.

--no-log_colour
Disable coloured output when logging.

--ignore_asm_unimplemented_break
Don't break for ASM_UNIMPLEMENTED_BREAK macros.

--trace_sim_messages
Trace simulator debug messages. Implied by **--trace-sim**.

--stack_trace_on_illegal
Print stack trace when an illegal exception is thrown.

--abort_on_uncaught_exception
Abort program (dump core) when an uncaught exception is thrown.

--no-randomize_hashes
Don't randomise hashes to avoid predictable hash collisions.

--hash_seed
Type: Int
Default: `0`

Fixed seed to use to hash property keys. `0` means random.

With snapshots, this option cannot override the baked-in seed.

--runtime_call_stats
Report runtime call counts and times.

--profile_deserialization
Print the time it takes to deserialise the snapshot.

--serialization_statistics
Collect statistics on serialised objects.

--no-regexp_optimization
Disable generation of optimised regexp code.

--testing_bool_flag
Type: Bool
Default: `true`

(Unknown)

--testing_maybe_bool_flag
Type: maybe_bool
Default: `unset`

(Unknown)

--testing_int_flag
Type: Int
Default: `13`

(Unknown)

--testing_float_flag
Type: Float
Default: `2.5`

(Unknown)

--testing_string_flag
Type: String
Default: `"Hello, world!"`

(Unknown)

--testing_prng_seed
Type: Int
Default: `42`

Seed used for threading test randomness.

--testing_serialization_file
Type: String
Default: `"/tmp/serdes"`

File in which to serialise heap.

--startup_src
Type: String
Default: `NULL`

Write V8 startup as C++ source. mksnapshot only.

--startup_blob
Type: String
Default: `NULL`

Write V8 startup blob file. mksnapshot only.

--profile_hydrogen_code_stub_compilation
Print the time it takes to lazily compile hydrogen code stubs.

--predictable
Enable predictable mode.

--force_marking_deque_overflows
Force overflows of marking deque by reducing its size to 64 words.

--stress_compaction
Stress the GC compactor to flush out bugs. Implies **--force_marking_deque_overflows**.

--manual_evacuation_candidates_selection
Test mode only flag. It allows a unit test to select evacuation candidates pages. Requires **--stress_compaction**.

--external_allocation_limit_incremental_time
Type: Int
Default: `1`

Time spent in incremental marking steps (in milliseconds) once the external allocation limit is reached.

--disable_old_api_accessors
Disable old-style API accessors whose setters trigger through the prototype chain.

--dump_counters
Dump counters on exit.

--map_counters
Type: String
Default: `" "`

Map counters to a file.

--js_arguments
Type: Arguments
Default: `" "`

Pass all remaining arguments to the script. Alias for `--`.

--gdbjit
Enable GDBJIT interface.

--gdbjit_full
Enable GDBJIT interface for all code objects.

--gdbjit_dump
Dump elf objects with debug info to disk.

--gdbjit_dump_filter
Type: String
Default: " "

Dump only objects containing this substring.

--log Minimal logging without API, code, GC, suspect, or sample-handling.

--log_all
Log all events to the log file.

--log_api
Log API events to the log file.

--log_code
Log code events to the log file without profiling.

--log_gc
Log heap samples on garbage collection for the hp2ps tool.

--log_handles
Log global handle events.

--log_suspect
Log suspect operations.

--prof
Log statistical profiling information. Implies **--log-code**.

--prof_cpp
Like **--prof**, but ignore generated code.

--no-prof_browser_mode
Disable browser-compatible mode for profiling when using **--prof**.

--no-log_regexp
Log regular expression execution.

--logfile
Type: String
Default: `"v8.log"`

Specify name of the log file.

--no-logfile_per_isolate
Don't separate log files for each isolate.

--ll_prof
Enable low-level Linux profiler.

--perf_basic_prof
Enable perf Linux profiler (basic support).

--perf_basic_prof_only_functions
Only report function code ranges to perf (i.e. no stubs).

--perf_prof
Enable perf Linux profiler (experimental annotate support).

--perf_prof_debug_info
Enable debug info for perf Linux profiler (experimental).

--gc_fake_mmap
Type: String
Default: `"/tmp/__v8_gc__"`

Specify the name of the file for fake gc mmap used in `ll_prof`.

--log_internal_timer_events
Time internal events.

--log_timer_events
Time events, including external callbacks.

--log_instruction_stats
Log AArch64 instruction statistics.

--log_instruction_file
Type: String
Default: `"arm64_inst.csv"`

AArch64 instruction statistics log file.

--log_instruction_period
Type: Int
Default: `4194304`

AArch64 instruction statistics logging period.

--redirect_code_traces
Output deopt information and disassembly into file `code-<pid>-<isolate id>.asm`.

--redirect_code_traces_to
Type: String
Default: `NULL`

Output deopt information and disassembly into the given file.

--hydrogen_track_positions
Track source code positions when building IR.

SEE ALSO

node(1)