

NAME

node — server-side JavaScript runtime

SYNOPSIS

node [*options*] [*v8-options*] [-e *string* | *script.js* | -][--][*arguments* ...]

node inspect [-e *string* | *script.js* | - | <*host*>:<*port*>] ...

node [--v8-options]

DESCRIPTION

Node.js is a set of libraries for JavaScript which allows it to be used outside of the browser. It is primarily focused on creating simple, easy-to-build network clients and servers.

Execute **node** without arguments to start a REPL.

OPTIONS

- Alias for stdin, analogous to the use of - in other command-line utilities. The executed script is read from stdin, and remaining arguments are passed to the script.
- Indicate the end of command-line options. Pass the rest of the arguments to the script.
If no script filename or eval/print script is supplied prior to this, then the next argument will be used as a script filename.
- abort-on-uncaught-exception
Aborting instead of exiting causes a core file to be generated for analysis.
- allow-fs-read
Allow file system read access when using the permission model.
- allow-fs-write
Allow file system write access when using the permission model.
- allow-addons
Allow using native addons when using the permission model.
- allow-child-process
Allow spawning process when using the permission model.
- allow-wasi
Allow execution of WASI when using the permission model.
- allow-worker
Allow creating worker threads when using the permission model.
- completion-bash
Print source-able bash completion script for Node.js.
- C, --conditions *string*
Use custom conditional exports conditions. *string*
- cpu-prof
Start the V8 CPU profiler on start up, and write the CPU profile to disk before exit. If --cpu-prof-dir is not specified, the profile will be written to the current working directory with a generated file name.
- cpu-prof-dir
The directory where the CPU profiles generated by --cpu-prof will be placed. The default value is controlled by the --diagnostic-dir. command-line option.
- cpu-prof-interval
The sampling interval in microseconds for the CPU profiles generated by --cpu-prof. The default is **1000**.
- cpu-prof-name
File name of the V8 CPU profile generated with --cpu-prof.
- diagnostic-dir
Set the directory for all diagnostic output files. Default is current working directory. Set the directory to which all diagnostic output files will be written to. Defaults to current working directory. Affects the default output directory of: --cpu-prof-dir. --heap-prof-dir. --redirect-warnings.

- `--disable-proto=mode`
Disable the `Object.prototype.__proto__` property. If *mode* is `'delete'`, the property will be removed entirely. If *mode* is `'throw'`, accesses to the property will throw an exception with the code `'ERR_PROTO_ACCESS'`.
- `--disable-wasm-trap-handler=mode`
Disable trap-handler-based WebAssembly bound checks and fall back to inline bound checks so that WebAssembly can be run with limited virtual memory.
- `--disallow-code-generation-from-strings`
Make built-in language features like `'eval'` and `'new Function'` that generate code from strings throw an exception instead. This does not affect the Node.js `'vm'` module.
- `--enable-fips`
Enable FIPS-compliant crypto at startup. Requires Node.js to be built with `./configure --openssl-fips`.
- `--enable-source-maps`
Enable Source Map V3 support for stack traces.
- `--entry-url`
Interpret the entry point as a URL.
- `--experimental-addon-modules`
Enable experimental addon module support.
- `--experimental-default-type=type`
Interpret as either ES modules or CommonJS modules input via `--eval` or `STDIN`, when `--input-type` is unspecified;
- `--experimental-config-file`
Specifies the configuration file to load.
- `--experimental-default-config-file`
Enable support for automatically loading `node.config.json`.
- `--experimental-import-meta-resolve`
Enable experimental ES modules support for `import.meta.resolve()`.
- `--experimental-loader=module`
Specify the *module* to use as a custom module loader.
- `--permission`
Enable the permission model.
- `--experimental-shadow-realm`
Use this flag to enable ShadowRealm support.
- `--experimental-test-coverage`
Enable code coverage in the test runner.
- `--experimental-test-isolation=mode`
Configures the type of test isolation used in the test runner.
- `--experimental-test-module-mocks`
Enable module mocking in the test runner.
- `--experimental-transform-types`
Enable transformation of TypeScript-only syntax into JavaScript code.
- `--experimental-eventsource`
Enable experimental support for the EventSource Web API.

- `--no-experimental-fetch`
Disable experimental support for the Fetch API.
- `--no-experimental-websocket`
Disable experimental support for the WebSocket API.
- `--experimental-webstorage`
Enable experimental support for the Web Storage API.
- `--no-experimental-global-customevent`
Disable exposition of the CustomEvent on the global scope.
- `--no-experimental-global-webcrypto`
Disable exposition of the Web Crypto API on the global scope.
- `--no-experimental-repl-await`
Disable top-level await keyword support in REPL.
- `--no-experimental-sqlite`
Disable the experimental node:sqlite module.
- `--no-experimental-strip-types`
Disable experimental type-stripping for TypeScript files.
- `--experimental-vm-modules`
Enable experimental ES module support in VM module.
- `--experimental-wasi-unstable-preview1`
Enable experimental WebAssembly System Interface support. This flag is no longer required as WASI is enabled by default.
- `--experimental-inspector-network-resource`
Enable experimental support for inspector network resources.
- `--force-context-aware`
Disable loading native addons that are not context-aware.
- `--force-fips`
Force FIPS-compliant crypto on startup (Cannot be disabled from script code). Same requirements as `--enable-fips`.
- `--frozen-intrinsics`
Enable experimental frozen intrinsics support.
- `--heapsnapshot-near-heap-limit=max_count`
Generate heap snapshot when the V8 heap usage is approaching the heap limit. No more than the specified number of snapshots will be generated.
- `--heapsnapshot-signal=signal`
Generate heap snapshot on specified signal.
- `--heap-prof`
Start the V8 heap profiler on start up, and write the heap profile to disk before exit. If `--heap-prof-dir` is not specified, the profile will be written to the current working directory with a generated file name.
- `--heap-prof-dir`
The directory where the heap profiles generated by `--heap-prof` will be placed. The default value is controlled by the `--diagnostic-dir` command-line option.
- `--heap-prof-interval`
The average sampling interval in bytes for the heap profiles generated by `--heap-prof`. The default is **512 * 1024**.

- `--heap-prof-name`
File name of the V8 heap profile generated with `--heap-prof`.
- `--icu-data-dir=file`
Specify ICU data load path. Overrides `NODE_ICU_DATA`.
- `--input-type=type`
Set the module resolution type for input via `--eval`, `--print` or `STDIN`.
- `--inspect-brk=[host:]port`
Activate inspector on `host:port` and break at start of user script.
- `--inspect-port=[host:]port`
Set the `host:port` to be used when the inspector is activated.
- `--inspect-publish-uid=stderr,http`
Specify how the inspector WebSocket URL is exposed. Valid values are **stderr** and **http**. Default is **stderr,http**.
- `--inspect-wait=[host:]port`
Activate inspector on `host:port` and wait for debugger to be attached.
- `--inspect=[host:]port`
Activate inspector on `host:port`. Default is **127.0.0.1:9229**.
- V8 Inspector integration allows attaching Chrome DevTools and IDEs to Node.js instances for debugging and profiling. It uses the Chrome DevTools Protocol.
- `--insecure-http-parser`
Use an insecure HTTP parser that accepts invalid HTTP headers. This may allow interoperability with non-conformant HTTP implementations. It may also allow request smuggling and other HTTP attacks that rely on invalid headers being accepted. Avoid using this option.
- `--jitless`
Disable runtime allocation of executable memory. This may be required on some platforms for security reasons. It can also reduce attack surface on other platforms, but the performance impact may be severe.
- This flag is inherited from V8 and is subject to change upstream. It may disappear in a non-semver-major release.
- `--localstorage-file=file`
The file used to store localStorage data.
- `--max-http-header-size=size`
Specify the maximum size of HTTP headers in bytes. Defaults to 16 KiB.
- `--max-old-space-size-percentage=percentage`
Sets the maximum memory size of V8's old memory section as a percentage of available system memory. This flag takes precedence over `--max-old-space-size` when both are specified. The *percentage* parameter must be a number greater than 0 and up to 100, representing the percentage of available system memory to allocate to the V8 heap.
- Note: This flag utilizes `--max-old-space-size`, which may be unreliable on 32-bit platforms due to integer overflow issues.
- `--napi-modules`
This option is a no-op. It is kept for compatibility.
- `--no-deprecation`
Silence deprecation warnings.

- `--no-extra-info-on-fatal-exception`
Hide extra information on fatal exception that causes exit.
- `--no-force-async-hooks-checks`
Disable runtime checks for ``async_hooks``. These will still be enabled dynamically when ``async_hooks`` is enabled.
- `--no-addons`
Disable the ``node-addons`` exports condition as well as disable loading native addons. When ``--no-addons`` is specified, calling ``process.dlopen`` or requiring a native C++ addon will fail and throw an exception.
- `--no-global-search-paths`
Do not search modules from global paths.
- `--no-warnings`
Silence all process warnings (including deprecations).
- `--node-memory-debug`
Enable extra debug checks for memory leaks in Node.js internals. This is usually only useful for developers debugging Node.js itself.
- `--openssl-config=file`
Load an OpenSSL configuration file on startup. Among other uses, this can be used to enable FIPS-compliant crypto if Node.js is built with `./configure --openssl-fips`.
- `--pending-deprecation`
Emit pending deprecation warnings.
- `--preserve-symlinks`
Instructs the module loader to preserve symbolic links when resolving and caching modules other than the main module.
- `--preserve-symlinks-main`
Instructs the module loader to preserve symbolic links when resolving and caching the main module.
- `--prof`
Generate V8 profiler output.
- `--prof-process`
Process V8 profiler output generated using the V8 option `--prof`.
- `--redirect-warnings=file`
Write process warnings to the given *file* instead of printing to stderr.
- `--report-compact`
Write **diagnostic reports** in a compact format, single-line JSON.
- `--report-dir --report-directory`
Location at which the **diagnostic report** will be generated. The ``file`` name may be an absolute path. If it is not, the default directory it will be written to is controlled by the `--diagnostic-dir` command-line option.
- `--report-filename`
Name of the file to which the **diagnostic report** will be written.
- `--report-on-fatalerror`
Enables the **diagnostic report** to be triggered on fatal errors (internal errors within the Node.js runtime such as out of memory) that leads to termination of the application. Useful to inspect various diagnostic data elements such as heap, stack, event loop state, resource consumption etc. to reason about the fatal error.

- `--report-on-signal`
Enables **diagnostic report** to be generated upon receiving the specified (or predefined) signal to the running Node.js process. Default signal is SIGUSR2.
- `--report-signal`
Sets or resets the signal for **diagnostic report** generation (not supported on Windows). Default signal is SIGUSR2.
- `--report-uncaught-exception`
Enables **diagnostic report** to be generated on un-caught exceptions. Useful when inspecting JavaScript stack in conjunction with native stack and other runtime environment data.
- `--secure-heap=n`
Specify the size of the OpenSSL secure heap. Any value less than 2 disables the secure heap. The default is 0. The value must be a power of two.
- `--secure-heap-min=n`
Specify the minimum allocation from the OpenSSL secure heap. The default is 2. The value must be a power of two.
- `--test`
Starts the Node.js command line test runner.
- `--test-concurrency`
The maximum number of test files that the test runner CLI will execute concurrently.
- `--test-coverage-branches=threshold`
Require a minimum threshold for branch coverage (0 – 100).
- `--test-coverage-exclude`
A glob pattern that excludes matching files from the coverage report
- `--test-coverage-functions=threshold`
Require a minimum threshold for function coverage (0 – 100).
- `--test-coverage-include`
A glob pattern that only includes matching files in the coverage report
- `--test-coverage-lines=threshold`
Require a minimum threshold for line coverage (0 – 100).
- `--test-force-exit`
Configures the test runner to exit the process once all known tests have finished executing even if the event loop would otherwise remain active.
- `--test-name-pattern`
A regular expression that configures the test runner to only execute tests whose name matches the provided pattern.
- `--test-reporter`
A test reporter to use when running tests.
- `--test-reporter-destination`
The destination for the corresponding test reporter.
- `--test-only`
Configures the test runner to only execute top level tests that have the `only` option set.
- `--test-shard`
Test suite shard to execute in a format of `<index>/<total>`.

- `--test-skip-pattern`
A regular expression that configures the test runner to skip tests whose name matches the provided pattern.
- `--test-timeout`
A number of milliseconds the test execution will fail after.
- `--test-update-snapshots`
Regenerates the snapshot file used by the test runner for snapshot testing.
- `--throw-deprecation`
Throw errors for deprecations.
- `--title=title`
Specify `process.title` on startup.
- `--tls-cipher-list=list`
Specify an alternative default TLS cipher list. Requires Node.js to be built with crypto support. (Default)
- `--tls-keylog=file`
Log TLS key material to a file. The key material is in NSS SSLKEYLOGFILE format and can be used by software (such as Wireshark) to decrypt the TLS traffic.
- `--tls-max-v1.2`
Set default `maxVersion` to `'TLSv1.2'`. Use to disable support for TLSv1.3.
- `--tls-max-v1.3`
Set default `maxVersion` to `'TLSv1.3'`. Use to enable support for TLSv1.3.
- `--tls-min-v1.0`
Set default `minVersion` to `'TLSv1'`. Use for compatibility with old TLS clients or servers.
- `--tls-min-v1.1`
Set default `minVersion` to `'TLSv1.1'`. Use for compatibility with old TLS clients or servers.
- `--tls-min-v1.2`
Set default `minVersion` to `'TLSv1.2'`. This is the default for 12.x and later, but the option is supported for compatibility with older Node.js versions.
- `--tls-min-v1.3`
Set default `minVersion` to `'TLSv1.3'`. Use to disable support for TLSv1.2 in favour of TLSv1.3, which is more secure.
- `--trace-atomics-wait`
Print short summaries of calls to `Atomics.wait()`. This flag is deprecated.
- `--trace-deprecation`
Print stack traces for deprecations.
- `--trace-event-categories categories`
A comma-separated list of categories that should be traced when trace event tracing is enabled using `--trace-events-enabled`.
- `--trace-event-file-pattern pattern`
Template string specifying the filepath for the trace event data, it supports `${rotation}` and `${pid}`.
- `--trace-events-enabled`
Enable the collection of trace event tracing information.
- `--trace-exit`
Prints a stack trace whenever an environment is exited proactively, i.e. invoking `process.exit()`.

- `--trace-sigint`
Prints a stack trace on SIGINT.
- `--trace-sync-io`
Print a stack trace whenever synchronous I/O is detected after the first turn of the event loop.
- `--trace-tls`
Prints TLS packet trace information to stderr.
- `--trace-uncaught`
Print stack traces for uncaught exceptions; usually, the stack trace associated with the creation of an **Error** is printed, whereas this makes Node.js also print the stack trace associated with throwing the value (which does not need to be an **Error** instance).

Enabling this option may affect garbage collection behavior negatively.
- `--trace-warnings`
Print stack traces for process warnings (including deprecations).
- `--track-heap-objects`
Track heap object allocations for heap snapshots.
- `--unhandled-rejections=mode`
Define the behavior for unhandled rejections. Can be one of `'strict'` (raise an error), `'warn'` (enforce warnings) or `'none'` (silence warnings).
- `--use-bundled-ca`, `--use-openssl-ca`
Use bundled Mozilla CA store as supplied by current Node.js version or use OpenSSL's default CA store. The default store is selectable at build-time.

The bundled CA store, as supplied by Node.js, is a snapshot of Mozilla CA store that is fixed at release time. It is identical on all supported platforms.

Using OpenSSL store allows for external modifications of the store. For most Linux and BSD distributions, this store is maintained by the distribution maintainers and system administrators. OpenSSL CA store location is dependent on configuration of the OpenSSL library but this can be altered at runtime using environment variables.

See `SSL_CERT_DIR` and `SSL_CERT_FILE`.
- `--use-env-proxy`
Parse proxy settings from `HTTP_PROXY/HTTPS_PROXY/NO_PROXY` environment variables and apply the setting in global HTTP/HTTPS clients.
- `--use-system-ca`
Use the trusted CA certificates present in the system store, in addition to the certificates in the bundled Mozilla CA store and certificates from `'NODE_EXTRA_CA_CERTS'`. On Windows and macOS, it uses system APIs to integrate additional trusted certificates. On other platforms it is similar to `'--use-openssl-ca'` with the difference being that it caches the certificates on first load, and the system certificates are added in a complementary manner without replacing certificates from other sources. This flag supersedes `'--use-openssl-ca'`.
- `--use-largepages=mode`
Re-map the Node.js static code to large memory pages at startup. If supported on the target system, this will cause the Node.js static code to be moved onto 2 MiB pages instead of 4 KiB pages.

`mode` must have one of the following values: `'off'` (the default value, meaning do not map), `'on'` (map and ignore failure, reporting it to stderr), or `'silent'` (map and silently ignore failure).
- `--v8-options`
Print V8 command-line options.

- `--v8-pool-size=num`
Set V8's thread pool size which will be used to allocate background jobs. If set to 0 then V8 will choose an appropriate size of the thread pool based on the number of online processors. If the value provided is larger than V8's maximum, then the largest value will be chosen.
- `--watch`
Starts Node.js in watch mode. When in watch mode, changes in the watched files cause the Node.js process to restart.

By default, watch mode will watch the entry point and any required or imported module. Use `--watch-path` to specify what paths to watch.
- `--watch-path`
Starts Node.js in watch mode and specifies what paths to watch. When in watch mode, changes in the watched paths cause the Node.js process to restart.

This will turn off watching of required or imported modules, even when used in combination with `--watch`.
- `--watch-kill-signal`
Customizes the signal sent to the process on watch mode restarts.
- `--zero-fill-buffers`
Automatically zero-fills all newly allocated Buffer and SlowBuffer instances.
- `-c, --check`
Check the script's syntax without executing it. Exits with an error code if script is invalid.
- `-e, --eval string`
Evaluate *string* as JavaScript.
- `-h, --help`
Print command-line options. The output of this option is less detailed than this document.
- `-i, --interactive`
Open the REPL even if stdin does not appear to be a terminal.
- `-p, --print string`
Identical to `-e`, but prints the result.
- `-r, --require module`
Preload the specified *module* at startup. Follows `require()`'s module resolution rules. *module* may be either a path to a file, or a Node.js module name.
- `-v, --version`
Print node's version.

ENVIRONMENT

FORCE_COLOR

Used to enable ANSI colorized output. The value may be one of: `1`, `true`, or *an empty string* to indicate 16-color support, `2` to indicate 256-color support, or `3` to indicate 16 million-color support. When used and set to a supported value, both the `NO_COLOR` and `NODE_DISABLE_COLORS` environment variables are ignored. Any other value will result in colorized output being disabled.

NO_COLOR

Alias for `NODE_DISABLE_COLORS`

NODE_COMPILE_CACHE *dir*

Enable the **module compile cache** for the Node.js instance.

NODE_DEBUG *modules...*

Comma-separated list of core modules that should print debug information.

NODE_DEBUG_NATIVE *modules...*

Comma-separated list of C++ core modules that should print debug information.

NODE_DISABLE_COMPILE_CACHE

Disable the **module compile cache** for the Node.js instance.

NODE_DISABLE_COLORS

When set to *1*, colors will not be used in the REPL.

NODE_EXTRA_CA_CERTS *file*

When set, the well-known “root” CAs (like VeriSign) will be extended with the extra certificates in *file*. The file should consist of one or more trusted certificates in PEM format.

If *file* is missing or misformatted, a message will be emitted once using **process.emitWarning()**, but any errors are otherwise ignored.

This environment variable is ignored when ``node`` runs as `setuid root` or has Linux file capabilities set.

The `NODE_EXTRA_CA_CERTS` environment variable is only read when the Node.js process is first launched. Changing the value at runtime using `process.env.NODE_EXTRA_CA_CERTS` has no effect on the current process.

NODE_ICU_DATA *file*

Data path for ICU (Intl object) data. Will extend linked-in data when compiled with small-icu support.

NODE_NO_WARNINGS

When set to *1*, process warnings are silenced.

NODE_OPTIONS *options...*

A space-separated list of command-line *options*, which are interpreted as if they had been specified on the command line before the actual command (so they can be overridden). Node.js will exit with an error if an option that is not allowed in the environment is used, such as `--print` or a script file.

NODE_PATH *directories...*

A colon-separated list of *directories* prefixed to the module search path.

NODE_PENDING_DEPRECATION

When set to *1*, emit pending deprecation warnings.

NODE_PENDING_PIPE_INSTANCES

Set the number of pending pipe instance handles when the pipe server is waiting for connections. This setting applies to Windows only.

NODE_PRESERVE_SYMLINKS

When set to *1*, the module loader preserves symbolic links when resolving and caching modules.

NODE_REDIRECT_WARNINGS *file*

Write process warnings to the given *file* instead of printing to stderr. Equivalent to passing `--redirect-warnings file` on the command line.

NODE_REPL_HISTORY *file*

Path to the *file* used to store persistent REPL history. The default path is `~/.node_repl_history`, which is overridden by this variable. Setting the value to an empty string (`""` or `" "`) will disable persistent REPL history.

NODE_REPL_EXTERNAL_MODULE *file*

Path to a Node.js module which will be loaded in place of the built-in REPL. Overriding this value to an empty string (``) will use the built-in REPL.

NODE_SKIP_PLATFORM_CHECK

When set to *1*, the check for a supported platform is skipped during Node.js startup. Node.js might not execute correctly. Any issues encountered on unsupported platforms will not be fixed.

NODE_TEST_CONTEXT

When set to *'child'*, test reporter options will be overridden and test output will be sent to std-out in the TAP format. If any other value is provided, Node.js makes no guarantees about the reporter format used or its stability.

NODE_TLS_REJECT_UNAUTHORIZED

When set to *0*, TLS certificate validation is disabled.

NODE_USE_ENV_PROXY

When enabled, Node.js parses the *HTTP_PROXY*, *HTTPS_PROXY* and *NO_PROXY* environment variables during startup, and tunnels requests over the specified proxy.

This currently only affects requests sent over *fetch()*. Support for other built-in http and https methods is under way.

NODE_USE_SYSTEM_CA

Similar to *--use-system-ca*. Use the trusted CA certificates present in the system store, in addition to the certificates in the bundled Mozilla CA store and certificates from *'NODE_EXTRA_CA_CERTS'*.

NODE_V8_COVERAGE *dir*

When set, Node.js writes JavaScript code coverage information to *dir*.

OPENSSL_CONF *file*

Load an OpenSSL configuration file on startup. Among other uses, this can be used to enable FIPS-compliant crypto if Node.js is built with *./configure --openssl-fips*.

If the *--openssl-config* command-line option is used, this environment variable is ignored.

SSL_CERT_DIR *dir*

If *--use-openssl-ca* is enabled, this overrides and sets OpenSSL's directory containing trusted certificates.

SSL_CERT_FILE *file*

If *--use-openssl-ca* is enabled, this overrides and sets OpenSSL's file containing trusted certificates.

TZ Specify the timezone configuration.**UV_THREADPOOL_SIZE** *size*

Sets the number of threads used in libuv's threadpool to *size*.

BUGS

Bugs are tracked in GitHub Issues: <https://github.com/nodejs/node/issues>

COPYRIGHT

Copyright Node.js contributors. Node.js is available under the MIT license.

Node.js also includes external libraries that are available under a variety of licenses. See <https://github.com/nodejs/node/blob/HEAD/LICENSE> for the full license text.

SEE ALSO

Website: <https://nodejs.org/>

Documentation: <https://nodejs.org/api/>

GitHub repository and issue tracker: <https://github.com/nodejs/node>