# Node.js Core Tests

This directory contains code and data used to test the Node.js implementation.

For a detailed guide on how to write tests in this directory, see [the guide on writing tests](#).

On how to run tests in this directory, see [the contributing guide](#).

For the tests to run on Windows, be sure to clone Node.js source code with the `autocrlf` git config flag set to true.

## Test Directories

| Directory | Runs on CI | Purpose |
|-----------|------------|---------|
| `abort` | Yes | Tests that use `--abort-on-uncaught-exception` and other situations where we want to test something but avoid generating a core file. |
| `addons` | Yes | Tests for [addon](#) functionality along with some tests that require an addon. |
| `async-hooks` | Yes | Tests for [async_hooks](#) functionality. |
| `benchmark` | Yes | Test minimal functionality of benchmarks. |
| `cctest` | Yes | C++ tests that are run as part of the build process. |
| `code-cache` | No | Tests for a Node.js binary compiled with V8 code cache. |
| `common` | | Common modules shared among many tests. [Documentation](#) |
| `doctool` | Yes | Tests for the documentation generator. |
| `es-module` | Yes | Test ESM module loading. |
| `fixtures` | | Test fixtures used in various tests throughout the test suite. |
| `internet` | No | Tests that make real outbound network connections. Tests for networking related modules may also be present in other directories, but those tests do not make outbound connections. |
| `js-native-api` | Yes | Tests for Node.js-agnostic [n-api](#) functionality. |
| `known_issues` | Yes | Tests reproducing known issues within the system. All tests inside of this directory are expected to fail. If a test doesn't fail on certain platforms, those should be skipped via `known_issues.status`. |
| `message` | Yes | Tests for messages that are output for various conditions (`console.log`, error messages etc.) |
| `node-api` | Yes | Tests for Node.js-specific [n-api](#) functionality. |
| `parallel` | Yes | Various tests that are able to be run in parallel. |
| `pseudo-tty` | Yes | Tests that require stdin/stdout/stderr to be a TTY. |

| | | |
|---|---|---|
| `pummel` | No | Various tests for various modules / system functionality operating under load. |
| `sequential` | Yes | Various tests that must not run in parallel. |
| `testpy` | | Test configuration utility used by various test suites. |
| `tick-processor` | No | Tests for the V8 tick processor integration. The tests are for the logic in `lib/internal/v8_prof_processor.js` and `lib/internal/v8_prof_polyfill.js`. The tests confirm that the profile processor packages the correct set of scripts from V8 and introduces the correct platform specific logic. |
| `v8-updates` | No | Tests for V8 performance integration. |