

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
<code>abort</code>	Yes	Tests that use <code>--abort-on-uncaught-exception</code> and other situations where we want to test something but avoid generating a core file.
<code>addons</code>	Yes	Tests for addon functionality along with some tests that require an addon.
<code>async_hooks</code>	Yes	Tests for <code>async_hooks</code> functionality.
<code>benchmark</code>	Yes	Test minimal functionality of benchmarks.
<code>cctest</code>	Yes	C++ tests that are run as part of the build process.
<code>code-cache</code>	Yes	Tests for a Node.js binary compiled with V8 code cache.
<code>common</code>	Yes	Common modules shared among many tests. Documentation
<code>doctool</code>	Yes	Tests for the documentation generator.
<code>es-module</code>	Yes	Test ESM module loading.
<code>fixtures</code>	Yes	Test fixtures used in various tests throughout the test suite.
<code>internet</code>	Yes	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.
<code>js-native-api</code>	Yes	Tests for Node.js-agnostic n-api functionality.
<code>known_issues</code>	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 <code>known_issues.status</code> .
<code>messages</code>	Yes	Tests for messages that are output for various conditions (<code>console.log</code> , error messages etc.)
<code>node-api</code>	Yes	Tests for Node.js-specific n-api functionality.
<code>parallel</code>	Yes	Various tests that are able to be run in parallel.
<code>pseudoTTY</code>	Yes	Tests that require stdin/stdout/stderr to be a TTY.
<code>pummel</code>	No	Various tests for various modules / system functionality operating under load.
<code>sequential</code>	Yes	Various tests that must not run in parallel.
<code>testpy</code>	Yes	Test configuration utility used by various test suites.

Directory	Purpose
tick-processor	Runs on Node for the V8 tick processor integration. The tests are for the logic in <code>lib/internal/v8_prof_processor.js</code> and <code>lib/internal/v8_prof_polyfill.js</code> . The tests confirm that the profile processor packages the correct set of scripts from V8 and introduces the correct platform specific logic.
v8-updates	Tests for V8 performance integration.