Trace events

> *Stability: 1 - Experimental*

The `trace_events` module provides a mechanism to centralize tracing information generated by V8, Node.js core, and userspace code.

Tracing can be enabled with the `--trace-event-categories` command-line flag or by using the `trace_events` module. The `--trace-event-categories` flag accepts a list of comma-separated category names.

The available categories are:

- `node` : An empty placeholder.
- `node.async_hooks` : Enables capture of detailed `async_hooks` trace data. The `async_hooks` events have a unique `asyncId` and a special `triggerId` `triggerAsyncId` property.
- `node.bootstrap` : Enables capture of Node.js bootstrap milestones.
- `node.console` : Enables capture of `console.time()` and `console.count()` output.
- `node.dns.native` : Enables capture of trace data for DNS queries.
- `node.environment` : Enables capture of Node.js Environment milestones.
- `node.fs.sync` : Enables capture of trace data for file system sync methods.
- `node.perf` : Enables capture of Performance API measurements.
  - `node.perf.usertiming` : Enables capture of only Performance API User Timing measures and marks.
  - `node.perf.timerify` : Enables capture of only Performance API timerify measurements.
- `node.promises.rejections` : Enables capture of trace data tracking the number of unhandled Promise rejections and handled-after-rejections.
- `node.vm.script` : Enables capture of trace data for the `vm` module's `runInNewContext()` , `runInContext()` , and `runInThisContext()` methods.
- `v8` : The V8 events are GC, compiling, and execution related.

By default the `node` , `node.async_hooks` , and `v8` categories are enabled.

```
node --trace-event-categories v8,node,node.async_hooks server.js
```

Prior versions of Node.js required the use of the `--trace-events-enabled` flag to enable trace events. This requirement has been removed. However, the `--trace-events-enabled` flag *may* still be used and will enable the `node` , `node.async_hooks` , and `v8` trace event categories by default.

```
node --trace-events-enabled

# is equivalent to

node --trace-event-categories v8,node,node.async_hooks
```

Alternatively, trace events may be enabled using the `trace_events` module:

```
const trace_events = require('trace_events');
const tracing = trace_events.createTracing({ categories: ['node.perf'] });
tracing.enable();  // Enable trace event capture for the 'node.perf' category

// do work

tracing.disable();  // Disable trace event capture for the 'node.perf' category
```

Running Node.js with tracing enabled will produce log files that can be opened in the `chrome://tracing` tab of Chrome.

The logging file is by default called `node_trace.${rotation}.log`, where `${rotation}` is an incrementing log-rotation id. The filepath pattern can be specified with `--trace-event-file-pattern` that accepts a template string that supports `${rotation}` and `${pid}`:

```
node --trace-event-categories v8 --trace-event-file-pattern '${pid}-${rotation}.log'
server.js
```

To guarantee that the log file is properly generated after signal events like `SIGINT`, `SIGTERM`, or `SIGBREAK`, make sure to have the appropriate handlers in your code, such as:

```
process.on('SIGINT', function onSigint() {
  console.info('Received SIGINT.');
  process.exit(130);  // Or applicable exit code depending on OS and signal
});
```

The tracing system uses the same time source as the one used by `process.hrtime()`. However the trace-event timestamps are expressed in microseconds, unlike `process.hrtime()` which returns nanoseconds.

The features from this module are not available in `Worker` threads.

The `trace_events` module

`Tracing` object

The `Tracing` object is used to enable or disable tracing for sets of categories. Instances are created using the `trace_events.createTracing()` method.

When created, the `Tracing` object is disabled. Calling the `tracing.enable()` method adds the categories to the set of enabled trace event categories. Calling `tracing.disable()` will remove the categories from the set of enabled trace event categories.

`tracing.categories`

- {string}

A comma-separated list of the trace event categories covered by this `Tracing` object.

`tracing.disable()`

Disables this `Tracing` object.

Only trace event categories *not* covered by other enabled `Tracing` objects and *not* specified by the `--trace-event-categories` flag will be disabled.

```
const trace_events = require('trace_events');
const t1 = trace_events.createTracing({ categories: ['node', 'v8'] });
const t2 = trace_events.createTracing({ categories: ['node.perf', 'node'] });
t1.enable();
t2.enable();

// Prints 'node,node.perf,v8'
console.log(trace_events.getEnabledCategories());

t2.disable(); // Will only disable emission of the 'node.perf' category

// Prints 'node,v8'
console.log(trace_events.getEnabledCategories());
```

`tracing.enable()`

Enables this `Tracing` object for the set of categories covered by the `Tracing` object.

`tracing.enabled`

- {boolean} `true` only if the `Tracing` object has been enabled.

`trace_events.createTracing(options)`

- `options` {Object}
  - `categories` {string[]} An array of trace category names. Values included in the array are coerced to a string when possible. An error will be thrown if the value cannot be coerced.
- Returns: {Tracing}.

Creates and returns a `Tracing` object for the given set of `categories`.

```
const trace_events = require('trace_events');
const categories = ['node.perf', 'node.async_hooks'];
const tracing = trace_events.createTracing({ categories });
tracing.enable();
// do stuff
tracing.disable();
```

`trace_events.getEnabledCategories()`

- Returns: {string}

Returns a comma-separated list of all currently-enabled trace event categories. The current set of enabled trace event categories is determined by the *union* of all currently-enabled `Tracing` objects and any categories enabled using the `--trace-event-categories` flag.

Given the file `test.js` below, the command `node --trace-event-categories node.perf test.js` will print `'node.async_hooks,node.perf'` to the console.

```
const trace_events = require('trace_events');
const t1 = trace_events.createTracing({ categories: ['node.async_hooks'] });
const t2 = trace_events.createTracing({ categories: ['node.perf'] });
const t3 = trace_events.createTracing({ categories: ['v8'] });

t1.enable();
t2.enable();

console.log(trace_events.getEnabledCategories());
```