/\*\*

\* 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.

\*

\* ```bash

\* 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.

\*

\* ```bash

\* 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:

\*

\* ```js

\* 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`](https://www.chromium.org/developers/how-tos/trace-event-profiling-tool) 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}`:

\*

\* ```bash

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

\* ```

\*

\* 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.

\* @experimental

\* @see [source](https://github.com/nodejs/node/blob/v17.0.0/lib/trace\_events.js)

\*/

declare module 'trace\_events' {

/\*\*

\* 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.

\*/

interface Tracing {

/\*\*

\* A comma-separated list of the trace event categories covered by this

\* `Tracing` object.

\*/

readonly categories: string;

/\*\*

\* 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.

\*/

disable(): void;

/\*\*

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

\* the `Tracing` object.

\*/

enable(): void;

/\*\*

\* `true` only if the `Tracing` object has been enabled.

\*/

readonly enabled: boolean;

}

interface CreateTracingOptions {

/\*\*

\* 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.

\*/

categories: string[];

}

/\*\*

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

\*

\* ```js

\* 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();

\* ```

\* @since v10.0.0

\* @return .

\*/

function createTracing(options: CreateTracingOptions): Tracing;

/\*\*

\* 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.

\*

\* ```js

\* 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());

\* ```

\* @since v10.0.0

\*/

function getEnabledCategories(): string | undefined;

}

declare module 'node:trace\_events' {

export \* from 'trace\_events';

}