

Diagnostic tooling support tiers

Diagnostic tooling is important to the consumers of Node.js. It is used both in development and in production in order to investigate problems. The failure of one of these tools may be as big a problem for an end user as a bug within the runtime itself.

The Node.js project has assessed the tools and the APIs which support those tools. Each of the tools and APIs has been put into one of the following tiers.

- Tier 1 - Must always be working (CI tests passing) for all Current and LTS Node.js releases. A release will not be shipped if the test suite for the tool/API is not green. To be considered for inclusion in this tier it must have a good test suite and that test suite and a job must exist in the Node.js CI so that it can be run as part of the release process. Tests on master will be run nightly when possible to provide early warning of potential issues. No commit to the current and LTS release branches should break this tool/API if the next major release is within 1 month. In addition:
 - The maintainers of the tool must remain responsive when there are problems;
 - The tool must be actively used by the ecosystem;
 - The tool must be heavily depended on;
 - The tool must have a guide or other documentation in the Node.js GitHub organization or website;
 - The tool must be working on all supported platforms;
 - The tool must only be using APIs exposed by Node.js as opposed to its dependencies; and
 - The tool must be open source.
- Tier 2 - Must be working (CI tests passing) for all LTS releases. An LTS release will not be shipped if the test suite for the tool/API is not green. To be considered for inclusion in this tier it must have a good test suite and that test suite and a job must exist in the Node.js CI so that it can be run as part of the release process. In addition:
 - The maintainers of the tool must remain responsive when there are problems;
 - The tool must be actively used by the ecosystem;
 - The tool must be heavily depended on;
 - The tool must have a guide or other documentation in the Node.js GitHub organization or website;
 - The tool must be open source.
- Tier 3 - If possible its test suite will be run at least nightly in the Node.js CI and issues opened for failures. Does not block shipping a release.
- Tier 4 - Does not block shipping a release.

- Unclassified - tool/API is new or does not have the required testing in the Node.js CI in order to qualify for a higher tier.

The choice of which tier a particular tool will be assigned to, will be a collaborative decision between Diagnostics WG and Release WG. Some of the criteria considered might be:

- If the tool fits into a key category as listed below.
- Whether the tool is actively used by the ecosystem.
- The availability of alternatives.
- Impact to the overall ecosystem if the tool is not working.
- The availability of reliable test suite that can be integrated into our CI.
- The availability of maintainer or community collaborator who will help resolve issues when there are CI failures.
- If the tool is maintained by the Node.js Foundation GitHub organization.

The current categories of tools/APIs that fall under these Tiers are:

- FFDC (F) - First failure data capture, easy to consume initial diagnostic information.
- Tracing (T) - use of logging to provide information about execution flow.
- Memory (M) - tools that provide additional information about memory used in the Heap or by native code.
- Profiling (P) - tools that provide additional information about where CPU cycles are being spent.
- AsyncFlow (A) - tools that provide additional insight into asynchronous execution flow.

Adding a tool to this list

Any tool that might be used to investigate issues when running Node.js can be added to the list. If there is a new tool that should be added to the list, it should start by being added to the “Not yet classified” or “Tier 4” lists. Once it has been added to the list “promotion” to Tier 3 through Tier 1 requires that the requirements listed above be met AND have agreement from Diagnostics WG and Release WG based on the criteria listed above.

Tiers

The tools are currently assigned to Tiers as follows:

Tier 1

Tool Type	Tool/API Name	Regular Testing in Node.js CI	Integrated with Node.js	Target Tier
FFDC	diagnostic report	Yes	Yes	1

Tool Type	Tool/API Name	Regular Testing in Node.js CI	Integrated with Node.js	Target Tier
-----------	---------------	-------------------------------	-------------------------	-------------

Tier 2

Tool Type	Tool/API Name	Regular Testing in Node.js CI	Integrated with Node.js	Target Tier
-----------	---------------	-------------------------------	-------------------------	-------------

Tier 3

Tool Type	Tool/API Name	Regular Testing in Node.js CI	Integrated with Node.js	Target Tier
Profiling	V8 CPU profiler	Partial (V8 Tests)	Yes	1
Profiling	prof-prof-process flags	Yes	Yes	1
Profiling	V8 CodeEventHandler API	Partial (V8 Tests)	Yes	2
Profiling	V8 -interpreted-frames-native-stack	Yes	Yes	2
Profiling	Linux perf	Yes	Partial	2

Tier 4

Tool Type	Tool/API Name	Regular Testing in Node.js CI	Integrated with Node.js	Target Tier
-----------	---------------	-------------------------------	-------------------------	-------------

Not yet classified

Tool Type	Tool/API Name	Regular Testing in Node.js CI	Integrated with Node.js	Target Tier
FFDC	node-report	No	No	1
Memory	mdb_V8	No	No	4
Memory	node-heapdump	No	No	2
Memory	V8 heap profiler	No	Yes	1
Memory	V8 sampling heap profiler	No	Yes	1

Tool Type	Tool/API Name	Regular Testing in Node.js CI	Integrated with Node.js	Target Tier
AsyncFlow	Async Hooks (API)	?	Yes	1
Debugger	V8 Debug protocol (API)	No	Yes	1
Debugger	Command line Debug Client	?	Yes	1
Debugger	llnode	?	No	2
Debugger	Chrome Dev tools	?	No	3
Tracing	trace_events (API)	No	Yes	1
Tracing	trace_gc	No	Yes	1
Tracing	DTrace	No	Partial	3
Tracing	LTTng	No	Removed?	N/A
Tracing	ETW	No	Partial	3
Tracing	Systemtap	No	Partial	?
Profiling	DTrace	No	Partial	3
Profiling	Windows Xperf	No	?	?
Profiling	0x	No	No	4
Profiling	node-clinic	No	No	too early
F/P/T	appmetrics	No	No	?
M/T	eBPF tracing tool	No	No	?