**# 2018-03-30 Public User Feedback Meeting - Node.js Tooling**

## Links

* **Recording**: https://www.youtube.com/watch?v=g5efRv1Wat0
* **GitHub Issue:** https://github.com/nodejs/user-feedback/issues/45
* **Previous Minutes:** https://docs.google.com/document/d/1NXH7qzaLGJcgNaswAcHVfwqluwaZPjp7iXtyRa4JZZA/edit
* https://github.com/orgs/nodejs/teams/user-feedback/members
* Issue #45 User feedback repo
* 2 files captured in Gist-by Tierney

# Agenda

Extracted from neuf-agenda labelled issues and pull requests from the user-feedback repo prior to the meeting.

- * Introduction to the Node.js Tooling with @boneskull
- * Topics (ref: #38):
- - Describe how your Tooling leverages Node.js.
- - Why do you use Node.js for Tooling?
- - What's working in the Node.js tooling ecosystem?
- - What isn't working in the Node.js tooling ecosystem?
- - What's new that could impact Node.js?
- * Open feedback forum

# Present

- Dan Shaw (@dshaw - User Feedback Champion, Mentorship, CommComm)
- Mihai Ene-Pietrosanu (@mihaiep, User Feedback)
- Michael Dawson (@mhdawson - User Feedback, CommComm, TSC Chair)
- Tierney Cyren (@bnb - User Feedback, CommComm Chair)
- Chris Hiller(@boneskull.com)
- Rebecca Turner (@iarna)
- Gleb Bahmutov (@bahmutov - Cypress.io)
- Uttam Pawar (@uttampawar - Intel)
- Ben Newman (@benjamn - Meteor)
- Kent C. Dodds (@kentcdodds - PayPal)
- Sean Larkin
- John David-Dalton (@jdalton)
-

- Gus Caplan (@devsnek)
- Kelly Selden (@kellyselden - @ember-cli)
-

****TOPIC****
## Notes

* Introduction to the Node.js Tooling with @boneskull

  * Developers of tooling have different needs from those active in core, for example user

    experience may be more important than performance.

  * Good to have more discussion/connection so that we can keep the needs of tooling

     developers in mind.

 * Topics (ref: #38):

  - Describe how your Tooling leverages Node.js.

  * Rebecca

      * npm perspective are that most users are tooling users as opposed to server users.

       * most command line tooling writing in Node.js (previously perl) as it is most familiar.

    * Kent

      * Almost all tooling is in node and used to build websites

      * Also tooling to build/deploy up web servers (Java used to package manifests)

      *  As Rebecca mentioned, even though may not be ideal scripting language, we like node because everyone can contribute easily.

      * Also use for servers and deploying servers.

      * Dan, used to build legacy Java?, Kent not as far as I'm aware. But Java is used to package build archives of node services.

* Gleb

  * Everything they do now runs on node, all shell js, don't want to install anything else

  * Biggest challenge is slight differences in how child process/streams across platforms (ex linux, windows etc)

* Uttam (Intel)

  * don't have tooling leveraging Node.js.  Focus on performance for node.js.  Have Vtune tool for Performance measurement

  * Contributed node-dc-eis which uses a number of the tooling

  * What is the hook from VTune to Node.js.  There is support in V8 and that is the link. Run Vtune from command line and it captures hardware data and gives micr

*  Ben (Meteor)

  * Fully embrace npm even though they have their own package system

  * They ship a specific version of Node.js in each release

  * Very nice as they don't have to test a matrix of versions.  Also means that they can ship new Node.js versions and count on them being there with the new features.

 * Michael what version do you use?  Ben on 0.10.X way to long, then jumped to version 4.X, latest Version jumped to Node 8.x, plan to keep up to date.

 * Tierney, what is transition path, what have the blockers beein.  Ben, despite moving in large jumps has been relatively smooth.  Have not had to hold too many peoples hands for upgrades either.

 * Dan, do Meteor users consider themselves Node.js users? Ben -> yes. Expectation is that they can drop large portion of Node app into Meteor.

- Why do you use Node.js for Tooling?

* Christopher (Moca)

* Testing framework for JavaScript. Node.js launcher for testing Node.js, can also be used in the Browser.

* John David Dalton (lodash)

* used node and node more and more over time to generate various builds of Node.js

* also works on module loader which is all node, adds es syntax to Node.js

* Kelly Selden (work on ember CLI)

* build tool for ember projects

* They have an LTS system which is tied to the Node.js LTS versions. Try to avoid transpiler so end up writing code for older versions for which a while.

* Community is interested in whole monorepo

* Ember.js runtime (6 or 8 week lifetime), build current release supports LTS releases. They support the LTS versions as opposed to having LTS for their modules.

* Gus Caplan (from node modules group, participating to see what the module owners are doing)

- What's working in the Node.js tooling ecosystem?

- What isn't working in the Node.js tooling ecosystem?

- What's new that could impact Node.js?

* Open feedback forum

```
  * What is not working
    * John David Dalton
      * Try to work across different versions(6 and 8). Need to
juggle when
        new APIs are added to newer versions
      * Some handy features which are not public, but if you ask
```

tendency is to try to remove/lock down so hesitant to ask.
* Attitude in node core that userland is less than. Sometimes will see userland option as not viable.
* ex Node fetch, Node should be more open to pulling in external packages (bundle more as opposed to integrate)

* Rebecca (npm)
* libuv does a relatively good job of hiding platforms, except for error codes where it just forwards on OS error. You don't even have doc on what error codes you might get.
* Historically have had some stdio juggling, but that is solid now
* Tierney Question for everybody
* please respond in the issue -> What would you like out of this Meeting
* Kelly - no additional issues beyond what was mentioned earlier
* Chris(Moca) - frustrations from technical standpoint are cross platform issues (for example fs.watch). Why are there things like cross-env, graceful-fs. Cross platform gotcha's. Much more Of a problem for tooling/tooling modules as they tend to run across more platforms. Deployment is more likely just linux
* Kent - concur that biggest problem are cross-platform issues. If we could make it so fs-extra,rimraf etc does not need to exist. Good thing that they do exist though as it makes things much easier than it used to be and
* debugging could still use more work. Inspector is good, what is needed is education story as well as people understanding That node-modules just contains a bunch of JavaScript.
* Gleb - linux - terminal support and progress bars are a problem for them. Would like to see better support in node core. We already used wrappers around progress bars, but there is great amount of how it works and how it looks like from Windows to Linux (docker).

* Ben - v8 has intimate relationship with Node.js. Source of bugs is backporting process for bringing changes into Node.js Segmentation faults due to partial gc backport. Wonders if Node.js being more up to date with v8 versions would help. They

use fibers which.  The one thing that keeps him up at night is what they have to do in order to support threading.  Dan suggestion to look at N-API and share what you still need that is not there.

****Closing Thoughts****

* potential topics for later session.
  * How to handle LTS for modules and relationship to Node.js LTS cycles.
  * Benchmarking to provide coverage for tooling concerns, does this Matter?

## Upcoming Meetings

* The next meeting is scheduled for Friday, 2018-04-13 at 17:00 UTC / noon ET / 11am CT / 9am PT.
* **Node.js Foundation Calendar**: https://nodejs.org/calendar - Click `+GoogleCalendar` at the bottom right to add to your own calendar.