NodeJS vendoring plan

Please see the meta bug for more information about the status or specifications of this feature.

Slack channel: #nodejs
Engineering contact: Dan Mosedale
EPM: Ron Manning

Category: Improvements to infrastructure

Audiences:

- Interested parties (without tons of context)
    - Much of this document is for you! The intent is to provide useful context about what exactly is going on.
    - The most interesting information is near the top, and things get increasingly detail-focussed further down.
    - Comments/suggestions are welcome and encouraged.
- NodeJS peers and other folks involved in executing this plan (people with more detailed context)
    - Any and all comments and suggestions are welcome and encouraged!
    - Much of our work is expected to be guided and tracked in the Project Details section at the bottom.
    - I'll be keeping the Project Status section in sync with the Project Details work as that evolves so that any interested parties can refer back here to see progress.

Project Scope

This feature will be landed in mozilla-central (i.e. Nightly) once complete. It is a Firefox-developer-facing feature, not an end-user facing one.

Background and Overview of the Goal of the Project?

This is a continuation of work done in 2018. The goal is to vendor in the packages already in the top-level mozilla-central package.json and have documented policy and processes for any Firefox engineer to vendor in other NodeJS-based build-tooling.

This unlocks the rich set of tooling in the NodeJS/npm ecosystem to both UJET (User Journey Engineering Team) and the larger Firefox team, so that we can start depending on

things available there in a coherent, managed, secure way. For UJET/New Tab, this will mean the ability to make uplifts easier, and make the standard build/test/source-code tooling work as expected with our code bases.

The work to be done here is largely coordination, documentation and sign-off work, likely along with some automation code to ensure coherency in security, license, and policy work.

What are the major Requirements for these Projects?

- A simple, documented, node-module-owners group responsible for reviewing vendored-in packages and managing node-related issues.
- Drive general policy documents to sign-off by key parties (node module owners, CTO organization, build team, security stakeholders).
- Implement any necessary automation for vendoring, license-checking, and security
- Vendor in existing packages currently in top-level mozilla-central package.json

Will there be any visible (UI) changes (please specify)?

No

Are there any backend changes?

No

Project Progress

Milestones	% complete	Status	Notes
Refactor, update, and socialize plan	100%	DONE	11/12/19
Draft/update policy docs for public posting	100%	DONE	11/21/19
Draft/update “How to Vendor” policy section.	100%	DONE	11/11/19

Set up nodejs-peers group	100%	DONE	12/2/19
Drive Node8/NPM/Yarn security updates into tree	100%	DONE	12/20/19
Upgrade to Node 10	100%	DONE	02/18/20
Solicit and incorporate public feedback on policies	20%	IN PROGRESS	11/21/19
Implement MVP landing mechanisms (`mach vendor node`, etc).	0%	WAITING on some policy feedback	
Get Required Signoffs	0%	WAITING on previous steps	
Vendor in eslint & friends	55%	IN PROGRESS	10/29/19 - draft patch for eslint itself created, next: incorporate remaining top-level packages.

Key Project Stakeholders/RACI chart

A, R dmose
A UJET mgmt (tspurway, ckarlof)
C Licensing (mhoye?, dnazar?)
C Static Analysis(Linting) (sylvestre)
C Sheriffs (ryanvm)
C Build (chman, mshal replacement?, nalexander?, glob)
C Developer Productivity (kmoir, glob)
A?, C Security (tritter, dveditz, jason, ulfr, stpeter, ekr, shuffman; others? Informed: jewilde, gguthe)
C NodeJS Module Owners/Peers (dmose on behalf of nodejs-peers@mozilla.org mailing list)
C CTO's office (stpeter, ekr)
C DevTools team (jlast, Informed: loganfsmyth, honza)
C UJET proj mgmt/eng (RoMan, k88hudson, Mardak)
I Principal Engineers/Directors (fx-eng-directors mailing list)
I public newsgroups (governance for NodeJS module setup, dev-builds, dev-platform, firefox-dev for rest)
I #go-node/#nodejs in Slack

QA Scope

What do you think is in-scope from the testing point of view?

Since this is a build-facing change, my expectation is that changes will happen a piece at a time, and will generally be handled by automated testing (i.e. try/treeherder). Some security and license issues may be exceptions to that; that will shake out and be handled in discussions and policy work with those teams.

What do you think is out of scope from the testing point of view?

I don't think PI manual QA is useful here. Automated testing should cover most things, and if there are non-trivial problems not caught by that, we'll either fix quickly or back out.

What platforms need to be covered?

All key desktop platforms are already covered by automated testing.

What Operating Systems should be covered?

All key desktop build OSes are already covered by automated testing. It's possible that an edge case or two could cause a build error on some obscure platform that's not caught that way, but not terribly likely. If such a thing does happen, we'll see bugs filed.

Are there any specific locales need to be covered apart from en-US?

No.

Any specific areas that would require Regression Testing (depending on the changes)?

No.

Risks and mitigations

Developers/reviewers unaware that our current policy is to only use vendored in NodeJS packages for builds import JavaScript code into the Firefox product itself, which has different license and security constraints. pull some package code directly into product.	TODO(dmosedale)
Could be framed as adding new security attack surface.	We're really just acknowledging the existing attack surface and making it more manageable and auditable. The security policy doc is all about the various mitigations that we're putting in place.

Known issues

(+Are there any known issues? Any tentative timeline to fix them?)

- Need to draft process for Node engine upgrades. I
    - It's not really clear how much process is going to be required here, if any.
    - We'll be doing such an upgrade (Node 8 -> Node 10) as part of this project, so that will help us get a sense of what's needed
- Consider grandfathering packages already-in-use with less stringent review requirements, since we're effectively exposed to any issues they create, and getting them in the tree's top-level package.json makes them auditable.
    - Not part of this specific project, but will probably be something we'll want to consider as a next step.
- Need to figure out how to manage local/top-level package.json sync for activity-stream.
    - Once we finish this project, User Journey / New Tab / DevTools teams will immediately be affected by this, and we'll discuss what's needed here then.

Additional notes

-

Project Details

Refactor, update, and socialize plan

- Pull details from now obsolete "Strawman policies" doc (DONE)
- Vet against "Strawman" doc to make sure we've covered all key pieces (DONE)
- Talk w/pauljt re sec next steps (DONE)
- Talk to w/chman re build tie-ins, node-module-owners group (DONE)
- Research sec options (DONE)
    - Talk to Julien (DONE)
    - Talk to tritter (DONE)
- Iterate on/refactor technical plan so that it's much clearer and more correct (DONE)
- Request basic review from key stakeholders (DONE 10/24/19))
    - Initial nodejs-peers, tspurway, RoMan
- Check in with kmoir (DONE 10/23/19)
- Request basic review for this plan from larger group of stakeholders v (DONE - 10/29/19)
    - Key project stakeholders + gguthe, vporov
- Figure out signoff process for overall policy docs and incorporate into this plan (DONE - 11/6/19)
    - decided w/Mossop on RACI charts for individual markdown docs
- Decide on ordering of public posting w.r.t. other docs (DONE - 11/12/19)
- Solicit and incorporate plan review from fx-eng-directors (DONE - 11/12/19)

Draft/update policy docs for public posting

- Extracted into npm policy draft, v2 (DONE)
- Review current licensing verbiage and notes with mhoye (DONE)
- Incorporate mbanner suggestion re phab group (DONE 10/22/19)
- Incorporate chman suggested build details (DONE 10/24/19)
- TODO(Mossop): new policy doc draft (IN PROGRESS 11/5/19)
- Convert to markdown for better formatting (DONE 11/6/19)
- Get markdown working in `mach doc` for later landing in m-c (IN PROGRESS 11/7/19, dmose)
- Draft/update how-to-vendor doc (DONE 11/11/19)
- Extract security policy into npm security doc draft v2 (DONE)
    - Draft threat model doc (DONE 11/8/19 - Mossop)
    - Add reference to https://snyk.io/blog/why-npm-lockfiles-can-be-a-security-blindspot-for-injecting-malici ous-modules/ and lockfile-lint (DONE 11/11/19)

- Link in how-to-vendor doc from main doc (DONE 11/13/19)
- Check with mhoye (IN PROGRESS)
- Add in security section:
    - link in threat-model doc (DONE 11/14/19)
    - Pull in other threats from previous iterations (DONE 11/14/19)
        - Transitive dependency sizes (resolve now) (npm-page
        - Sandboxing (ongoing/later)
        - Time-to-wait before vendoring (resolve now)
        - …
- Decide & document use of github issues (DONE 11/18/19)

Draft/update “How to Vendor” document for public posting (DONE)

- Edit draft to reviewable state
- Update docs with draft or agreed-upon content from other policy sections

Set up nodejs-peers group (DONE)

- Chat re possible build sub-module with build owner (chman) (DONE)
- Clarify peer roles (DONE)
- Work with Mossop to draft top-level NodeJS module description (DONE)
- Create mailing list (nodejs-peers) (DONE 10/22/19)
- Merge slack #go-node into #nodejs channel for general use (DONE 10/24/19)
- Key members have verbally agreed to be part of this group in the past; need to reach out to them and see if they’re still available
    - Mossop (DONE - YES)
    - Mardak (DONE - YES)
    - k88hudson (DONE - YES)
    - jlast (DONE - YES)
    - Standard8 (DONE - YES)
    - nalexander (DONE - YES, but helper with build-perspective, not a peer to do reviews)
    - mshal (DONE 10/24/19 - YES, but helper with build-perspective, not a peer to do reviews)
    - dcoates (DONE - YES)
- Set up meeting of nodejs-peers (both to drive project forward, and find folks interested in picking up individual pieces of work) (DONE - 11/21/19)
- Post proposal to governance mailing list (DONE)
- Add NodeJS module to module list after proposal period expires (DONE)

Upgrade Node 8 to Node 10 because of Node 8 EOL (DONE)

- update patch to latest NodeJS and push to try (DONE - 12/10/19)
- Meet with key stakeholders and Berlin and come to a decision (DONE - week of 1/27/20)
- Update Node 10 bug with status from Berlin meeting (DONE - 02/11/20)
- File issues to document policy changes from Berlin meeting (DONE - 02/11/20)
- Initial post announcing change (DONE by Mardak - week of 02/06/20)
- Post announcing incoming change and specific consequences (DONE - 02/11/20)

- make TC aliasing plan (see #NodeJS convo) (DONE - 2/12)
- Document how to test toolchain archives of `mach bootstrap`
    - MOZ_SCM_LEVEL=0 ./mach try fuzzy
        - Select node stuff, “build-{sorta-*}/opt”
    - MOZ_SCM_LEVEL=0 ./mach bootstrap
    - Run `./node -v` in ~/.mozbuild/node to check the version
- Patch stuff
    - Update various versions in touched files (DONE - 2/12)
    - update repack-node.sh (DONE - 2/12)
    - Remove # XXX (DONE - 2/12)
    - Get rid of npx (DONE)
    - Post patch (DONE)
        - Ask about why last commit makes `mach try fuzzy` require --full to show all 4 toolchain-*node10 builds? Only macosx64 and win32 are shown by default.
    - Get review & land update

Solicit and incorporate necessary feedback on policies

- Request nodejs-peers start watching repo and responding to issues as available (DONE 11/21/19)
- Post plan & policy docs publicly (DONE 11/21/19)
- Triage existing feedback into github issues (IN PROGRESS 11/21/19)
- Pull context from v2 into v3(github version) policy draft (DONE 12/2019, dmose)
- Put together signoff plan (IN PROGRESS, 2020-Mar-12)
- Reach out to folks re sec ownership & signoff
    - TRitter (DONE), Joe Walker, Sylvestre to understand how their orgs may be involved
- Figure out RACI plan details & github issue details
- Add blurb to upcoming fx-developer tues meeting
- Address any incorporate incoming feedback
- Iterate on open issues

Implement MVP landing mechanisms (eg `mach vendor node`, etc.)

- Scope out which pieces of automation (if any) are required to start landing vendored packages. Get signoff from license & security folks
- Mach vendor node
    - Scope out bugs (review How to Vendor github doc…)
    - Triage bugs w/nodejs-peers & prioritize what’s truly required for MVP
    - Code
    - Land!
- Set up nodejs-peers group in Phabricator to auto-block reviews that touch the top-level vendored node_modules / package.json / package-lock.json.
- Set up notification mechanisms (email to nodejs-peers?) for package sec updates, nodejs version sec updates, other updates

Get Required Signoffs

- Use RACI chart to mail probably signatories with deadline

- Consider meeting a la NewTab / Pocket handoff; schedule if appropriate
-

Vendor in eslint & friends (WAITING on required signoffs)

- Work with +mbanner@mozilla.com on eslint-related packages (bug)
- Work with +vporof@mozilla.com on prettier related packages
- Work with +jlast@mozilla.com on devtools-related pckages

Post vendoring Tidbits

- Consider how to prioritize switch to “fetch” artifacts (see “Node Engine concerns” in strawman doc
- Discuss with Node peers how to schedule/plan switch to Node 12 (filing a bug seems less than ideal)...