

10:57 AM <holoway> Hi guys - I'm going to be mobile for the fort few minutes here

10:57 AM <jtimberman> jonlives: i like the chicklets on the laptop over a stnadard non mechanical keyboard.

10:58 AM <@jonlives> jtimberman me too - that's why I use the fullsize apple keyboard at mah desk

10:58 AM <jtimberman> but now i'm a mechanical convert :)

11:04 AM <stevenmurawski> Hey all! When holoway is ready, he's first up with a review of today's agenda. I'm standing in for Nathen today.

11:05 AM <@jonlives> rgr stevenmurawski

11:06 AM <drag00n> (thumbs up)

11:07 AM <stevenmurawski> Or, to save a few minutes and let holoway spend his time on the interesting bits, I can recap the agenda

11:07 AM <@jonlives> let's do that

11:08 AM <stevenmurawski> We've got a bit of previous business from the last meeting. There is still an open action item on an RFC for nightly builds and the RFC for dialects needs a go/no go into ready to merge.

11:08 AM <stevenmurawski> For this weeks new business

11:08 AM <holoway> ...aand, mobile ment missing

11:08 AM <holoway> recap agenda

11:08 AM <holoway> ok

11:08 AM <holoway> hi guys

11:08 AM <@jonlives> hi holoway!

11:08 AM <kierrr> ohai

11:09 AM <@zts> hola

11:09 AM <stevenmurawski> hi holoway, I started to recap the agenda to save you a few minutes

11:09 AM <holoway> cool

11:09 AM <stevenmurawski> Was just starting to announce the new business if you want to take over

11:09 AM <holoway> yeah, I got it

11:09 AM — **stevenmurawski** turns over the mike to holoway

11:10 AM <holoway> RFC for data bag items, the request to open the opscore-webui2 repo, xml editing, partial tempates during bootstrap, event handler DSL, client\_compat field to metadata

11:10 AM <holoway> so lets jump right in

11:10 AM <holoway> we are going to move to nightly builds and stable releases cut on a cadence

11:10 AM <holoway> we'll put an RFC out with the plan

11:11 AM <@jonlives> nice

11:11 AM <holoway> I've asked the Legion of Builds and the Quality and Tools teams here at Chef to run with that

11:11 AM <jtimberman> :D

11:11 AM <holoway> the side effect should be that LTs for subsystems and maintainers can see the results of their work happen faster

11:11 AM <mmzyk> fyi, we renamed the opscore-webui2 repo to chef-manage.

11:12 AM <holoway> I also predict it will cause us to have lots of conversation about how to preserve back compat while implementing new features (dark launching features on the client)

11:12 AM <holoway> but we'll cross that bridge when we come to it

11:12 AM <holoway> dialects

11:12 AM <@jonlives> that's a good bridge to have.

11:12 AM <jtimberman> I love that idea.

11:12 AM <holoway> right?

11:13 AM <holoway> I'm +1 on merging dialects, with the provisions we added in the conversation. It gets us cleaner internals, rounds out functionality that's obviously a gap, and provides ground for people to experiment

11:13 AM — coderanger ::waves::

11:13 AM <@jonlives> dialects wise I'm mainly +1 cos it's coderanger doing the first impl.

11:13 AM <jtimberman> +1 on dialects

11:14 AM <holoway> what we want to make really clear to the community at large is that we're not adding any dialects :)

11:14 AM <holoway> but, like the POSHChef stuff shows, there is clearly interest

11:14 AM <@jonlives> yeah

11:14 AM <holoway> in connecting to the Chef ecosystem without necessarily being tied up to the existing clients model

11:14 AM <holoway> lets have 30 seconds if anyone thinks this will truly blow up the world

11:15 AM <holoway> otherwise, I'm going to have the RFC editors ship away

11:15 AM <jtimberman> if someone wants to rock out a haskell chef we shouldn't stop em

11:15 AM <@jonlives> someara: you had some questions most recently

11:15 AM <someara> around the scope of it

11:15 AM <Inxchk> jtimberman: we should send them cookies and hugs, because clearly their lives have gone badly wrong

11:15 AM <holoway> coderanger?

11:15 AM <coderanger> someara: scope outlined in the RFC is just adding the hooks internally, no new format support with the exception of maybe chef\_fs

11:16 AM <holoway> if we wanted to be political, this would've been an internals cleanup that happens to make experimenting easy

11:16 AM <someara> carpe diem

11:16 AM <holoway> but coderanger was honest about his intentions :)

11:16 AM — holoway hugs coderanger

11:16 AM <@jonlives> carpe dialect, feliz noahdad.

11:17 AM <holoway> feliz noahdad INDEED

11:17 AM <holoway> ok

11:17 AM <holoway> lets move on

11:17 AM <@jonlives> holoway: is that one decided for merge then?

11:17 AM <holoway> <https://github.com/opscode/chef-rfc/pull/79>

11:17 AM <holoway> jonlives: aye

11:17 AM <@jonlives> gravy

11:17 AM <coderanger> :-)

11:18 AM <@jonlives> #79 I'm really not that keen on

11:18 AM <holoway> because of the dynamic nature of it?

11:19 AM <holoway> I would've given people that API, I was just lazy

11:19 AM <jtimberman> the 'data bag from file' stuff actually would load rb data bags. i've done that in the past. it uses the same file loading mechanism as roles and environments

11:19 AM <@jonlives> largely yes - I like the idea of databags as a simple static lump of json.

11:19 AM <holoway> and it feels like the dialects implementation could easily cover a ruby and json implementation for it

11:19 AM <jtimberman> node from file, client from file, both do it too

11:19 AM <holoway> ya

11:19 AM <holoway> that's what I thought

11:19 AM <holoway> it basically already works

11:19 AM <holoway> it just uploads the json from the created object

11:19 AM <holoway> it just doesn't work everywhere

11:20 AM <holoway> jonlives: they always boil down to a static lump  
11:20 AM <holoway> the idea that you could express it with ruby instead of json isn't so the data bag would be dynamic when its at rest on a server  
11:20 AM <@jonlives> holoway: right - but my point was more you can't tell from your repo what's gonna end up on the srver. I don't like .rb roles much either, but I'm too late for that one :p  
11:20 AM <jtimberman> it makes it harder to reason about what is going on and where the data comes from. if you need to extract data or generate it from somewhere i think writing an intermediary tool is a better plan  
11:21 AM <@jonlives> ^ i agree  
11:21 AM <lamont\_oc> yeah the metadata.rb vs metadata.json war turned me off of .rb quite a bit  
11:21 AM <holoway> (it's cool, the .json people were wrong)  
11:21 AM — holoway dies  
11:21 AM <jtimberman> lol  
11:21 AM — jtimberman hugs holoway  
11:21 AM <kierrr> :D  
11:22 AM <holoway> other maintainers have a view?  
11:22 AM <kierrr> metadata.yaml was better anyway  
11:22 AM <lamont\_oc> but it doesn't make any sense to support rb for everything other than data bags, so +1  
11:22 AM <jtimberman> kierrr: i hate whitespace  
11:22 AM <holoway> this is where I am  
11:22 AM <holoway> we do this for every other object  
11:22 AM <holoway> the support is right there  
11:22 AM <holoway> we just dont' expose it  
11:22 AM <jtimberman> consistency is good for sure  
11:22 AM <holoway> and we're about to send coderanger on a mission to clean up all the duplication  
11:22 AM <coderanger> Make it a dialect gem, if people use it a lot it can be a candidate for core  
11:22 AM <holoway> lets just get it done, even if we think its probably a bad idea in practice  
11:22 AM <jtimberman> we don't have to use it or recommend it but again not get in the way of people that want to do that  
11:23 AM <@jonlives> holoway: so you're saying we could have him \*remove\* .rb support? ;)  
11:23 AM <holoway> it's just inconsistent  
11:23 AM <kierrr> jtimberman: i would never suggest yaml in a non-trolling way  
11:23 AM <holoway> jonlives: no, because I would deciderise that away  
11:23 AM <holoway> :)  
11:23 AM <holoway> but sure, we \*could\*  
11:23 AM <coderanger> kierrr: I would ;-)  
11:23 AM <@jonlives> likewise, not gonna stand in the way of supporting the last thing.

---

11:23 AM <holoway> ok - lets ask the current set of core maintainers to weigh in

11:23 AM <jtimberman> coderanger: you like python and thus significant whitespace though ;)

11:23 AM <jtimberman> anyway

11:23 AM <holoway> I'm provisionally +1 on having this happen as part of coderangers dialect support for core objects

11:24 AM <jtimberman> yed

11:24 AM <jtimberman> yes

11:24 AM <ranjib> -1 from me

11:24 AM <holoway> if there is no other argument than "I think people can footgun themselves"

11:24 AM <holoway> I'm not sure that's compelling, since the people who want to do it seem eager to do so, and saying "write a generator" isn't necessarily friendlier

11:24 AM <holoway> ranjib: reason?

11:25 AM <coderanger> My biggest concern is the proposed format isn't like the other Ruby DSLs

11:25 AM <ranjib> pretty same as others have given, its largely confusing, and for implementing it outside chef, we dont need RFC

11:25 AM <holoway> right - we need to normalize the DSL

11:25 AM <coderanger> It isn't a bunch of key/value method calls, instead it is a single expression value

11:25 AM <holoway> right

11:25 AM <coderanger> technically doable, but kind of confusing

11:26 AM <holoway> so lets move this to a discussion on the rfc

11:26 AM <holoway> and withhold judgement to next meeting

11:26 AM <holoway> and summon the maintainer krakens

11:26 AM <@jonlives> agreed

11:26 AM <@jonlives> I think I like being a kraken.

11:26 AM <holoway> coderanger: please comment that way re the shape of the api

11:26 AM <holoway> jonlives: I'm a giver

11:26 AM <holoway> ranjib: thank you for weighing in. are you a core maintainer yet?

11:27 AM <holoway> ok, next topic

11:27 AM <@jonlives> ranjib isn't a maintainer \*yet\* ;)

11:27 AM <holoway> <https://github.com/opscode/chef-rfc/pull/81/files>

11:27 AM <holoway> ranjib, you want to introduce?

11:28 AM <ranjib> i'll close it,

11:28 AM <ranjib> supporting it will be pain on solaris and aix,

11:28 AM <holoway> why not use the pure ruby xml libs?

11:28 AM <ranjib> due to nokogiri

11:28 AM <ranjib> will be slow?

11:28 AM <holoway> so, in general, I'm a huge +1000 to adding way more resources to core chef

11:28 AM <jtimberman> holoway: rexml?  
11:28 AM <holoway> ranjib: who cares?  
11:28 AM <holoway> :)  
11:29 AM <jtimberman> hahaha  
11:29 AM <holoway> will it really be that slow  
11:29 AM <jtimberman> its already ruby  
11:29 AM <ranjib> yeah,,i was thinking of having xml, yaml etc files in core chef  
11:29 AM <@ssd7> I'd love to see many of these types of file-type resource, xml\_file, ini\_file, json\_file  
11:29 AM <jtimberman> trollface.jpg  
11:29 AM <ranjib> i'll be fine with doing it rexml  
11:29 AM <holoway> if you are editing a ton of xml, for sure that will be true  
11:29 AM <thom> sadface.jpg  
11:29 AM <kallistec> yeah, unless we're talking 10 seconds vs. 0.001 seconds  
11:29 AM <adamleff> +1 - if it's "too slow" for some folks' converges, they don't have to use it, right?  
:)  
11:29 AM <holoway> but you really arent  
11:29 AM <Inxchk> people doing a lot of xml work expect it to be slow anyway  
11:29 AM <@jonlives> i'm not so +1000 on adding stuff like that into core chef  
11:29 AM <jtimberman> ssd7: +697234 for ini\_file, since thats a significant numbere of random config files you want to manage  
11:29 AM <kierrr> ssd7: +1  
11:29 AM <coderanger> This isn't just xml\_file though, it is more like FileEdit  
11:29 AM <ranjib> but this is one of those resources i always use for JVM apps,  
11:29 AM <lamont\_oc> yeah, i'd like to see it implemented as a cookbook first, using rexml, and extended to xml\_file/ini\_file/json\_file/etc -- i don't want a repeat of file/template/cookbook\_file/remote\_file organic growth without consistency  
11:29 AM <holoway> coderanger: we should put FileEdit into core too, finally  
11:29 AM <coderanger> Which is something we pretty actively discourage.  
11:30 AM <jtimberman> holoway: it ... is? or do you mean someara line cookbook  
11:30 AM <holoway> yeah, but I'm pretty sure we're violating the "you know best in the end" tennet  
11:30 AM <someara> that thing needs a rewrite  
11:30 AM <holoway> jtimberman: it's not a real resource, last time I checked  
11:30 AM <@jonlives> vetting it for addition to core when it's already in cookbook form does seem like the way to go I think  
11:30 AM <thom> or just use augeus? ;)  
11:30 AM <thom> trolllololol  
11:30 AM <holoway> yeah, writing it outside core first makes sense  
11:30 AM <jtimberman> thom: dude.  
11:30 AM <someara> actually augeus is a good idea  
11:30 AM <holoway> in general, I think we need to have more batteries included resources  
11:30 AM <jtimberman> let's not get completely crazy  
11:30 AM <coderanger> holoway: Fair, but it needs to be a lot less brittle if it isn't going to be a thing that lives in the attic, and this would be too.  
11:30 AM <jtimberman> ;)  
11:31 AM <holoway> coderanger: agreed  
11:31 AM <holoway> one of the reasons I want to move to dailies and aggressive release automation is that I want more resources in core  
11:31 AM <ranjib> lamont\_oc, ack. will get a cookbook based implementation going , then discuss it further  
11:31 AM <holoway> look at the work someara is doing in awesome cookbook resource factoring

11:31 AM <ranjib> does that make sense?

11:31 AM <holoway> ranjib: yes - update the rfc and lets get a provisional implementation going

11:31 AM <thom> someara: idea yes, implementation sheeesh

11:32 AM <coderanger> holoway: I don't see how those two things are connected at all, but thats a topic for you and I to discuss another day methinks

11:32 AM <holoway> that stuff should be in core

11:32 AM <holoway> yeah, lets put it on the agenda

11:32 AM <lamont\_oc> yeah, i'm not against the idea, i just don't see the time to shave the nokogiri yak

11:32 AM <holoway> ok, next up

11:32 AM <@jonlives> and it's a big yak. and it's angry.

11:32 AM <jtimberman> lamont\_oc: ranjib said he'd be okay with rexml

11:32 AM <holoway> ranjib on a roll - <https://github.com/opscode/chef-rfc/pull/82>

11:32 AM <lamont\_oc> ranjib would also like json\_file as well or one resource with :xml and :json attributes or whatever...

11:33 AM <holoway> [https://github.com/ranjib/chef-rfc/blob/bootstrap\\_partial/new/rfcXXX-bootstrap-partial.md](https://github.com/ranjib/chef-rfc/blob/bootstrap_partial/new/rfcXXX-bootstrap-partial.md)

11:33 AM <coderanger> I'm not sure I understand how this is supposed to work

11:33 AM <holoway> you want to intro, ranjib

11:33 AM <ranjib> lamont\_oc, yeah, currently folks templatize them,, which is pain

11:33 AM <coderanger> What exactly does --partial do?

11:34 AM <@ssd7> ranjib: Is the idea on #82 that we would render the partial content as part of the chef-full template?

11:34 AM <ranjib> yup, thats an RFC that will allow people to maintain their bootstrap customization separately

11:34 AM <ranjib> while using standard bootstrap templates

11:34 AM <ranjib> for example, minimal distros..

11:34 AM <coderanger> +1 to that idea, just unclear how the new system would work.

11:34 AM <miketheman> I don't understand why this needs to be put into the bootstrapper - we can perform imports of another template

11:34 AM <@jonlives> I actually like this one - allowing you to eg use a partial to install curl and then use a std template is pretty neat

11:34 AM <holoway> yeah, I have implementation questions

11:34 AM <ranjib> miketheman, then you would miss all the gears from standard template

11:35 AM <coderanger> refactoring the core bootstrap templates so you can reuse bits of them seems like a cool thing though

11:35 AM <miketheman> @ranjib: no, we don't.

11:35 AM <holoway> can I not import from erb?

11:35 AM <ranjib> and as chef will evolve you'll diverge

11:35 AM <miketheman> sudo <%= import("#{Pathname.new(@config[:template\_file]).dirname}/chef-full.erb") %>

11:35 AM <miketheman> that's what we put in the bottom of each template file

11:35 AM <miketheman> and then we call the "specific" one

11:35 AM <ranjib> miketheman, and you copyover the chef-full.erb ? thats not the gemfile path right?

11:36 AM <@jonlives> miketheman: but then you're modifying the standard templates, which is what this is meant to avoid having to do

11:36 AM <miketheman> jonlives: not modifygin standard templates

11:36 AM <holoway> he's literally slurping

11:36 AM <ranjib> miketheman, this RFC will let you do the same with a CLI flag

11:36 AM <miketheman> holoway: exactly



11:36 AM <miketheman> ranjib: we have a LOT of cli flags already

11:36 AM <@jonlives> miketheman: oh sorry misread what you said you do, I getcha.

11:37 AM <holoway> can I propose that miketheman shows an example on the RFC Comments of how he would implement this with the existing

11:37 AM <coderanger> ranjib: What does that cli flag do exactly?

11:37 AM <holoway> and ranjib reply with how he feels about the shape of that as a solution

11:37 AM <@jonlives> I think that makes sense

11:37 AM <jtimberman> do we have as many command line arguments as rsync yet?

11:37 AM <holoway> and if he doesn't like it, expand the RFC with details of what the flag would do and how it would interact

11:37 AM <ranjib> miketheman, yeah, i was thinking of supporting -- style arbitrary flags, where stuff after -- will be passed on to template

11:37 AM <holoway> command line flags are cheap

11:37 AM <miketheman> jtimberman: that's a bad bar to measure against

11:37 AM <@ssd7> We could maybe make miketheman's life easier by having a standard\_template function that would pull the template from the right location and such

11:37 AM <holoway> right

11:37 AM <jtimberman> what holoway said, flags are cheap

11:37 AM <holoway> we could make it much easier to import

11:38 AM <holoway> if that's the road we prefer

11:38 AM <lamont\_oc> i'm actually more in favor of this road:  
<https://github.com/opscode/bootstrapper>

11:38 AM <holoway> miketheman and ranjib - are you cool with my proposal?

11:38 AM <ranjib> +1

11:38 AM <miketheman> holoway: I'm happy to show some code

11:38 AM <holoway> dope

11:38 AM <coderanger> ranjib: That sounds like an error waiting to happen, but +1 for a --bootstrap-variable option

11:38 AM <holoway> then lets come back and talk next week

11:38 AM <holoway> er next meeting

11:38 AM <kallistec> yes CLI flags are cheap but they lead you to a path of doing run-book-y stuff rather than infra as code stuff

11:38 AM <holoway> and in the github thread

11:38 AM <holoway> lamont\_oc: I think a re-thought bootstrapper is necessary AS WELL

11:39 AM <holoway> but lets not make perfect the enemy of the good :)

11:39 AM <lamont\_oc> k

11:39 AM <jtimberman> kallistec: this is why i've started using chef-provisioning. but there are folks who love some knife bootstrap

11:39 AM <holoway> because ranjib cannot be stopped

11:39 AM <holoway> [https://github.com/ranjib/chef-rfc/blob/event\\_handler\\_hook/new/rfcXXX-event-handler-dsl.md](https://github.com/ranjib/chef-rfc/blob/event_handler_hook/new/rfcXXX-event-handler-dsl.md)

11:39 AM <holoway> ranjib, overview?

11:39 AM <coderanger> +bigint on this one, might use slightly less common word than "on" though

11:40 AM <ranjib> ^ that RFC gives recipe DSL method to add anonymous handlers

11:40 AM <coderanger> possibly on\_event

11:40 AM <coderanger> just to avoid conflicts with existing code

11:40 AM <ranjib> so, i can add resource specific handlers etc

11:40 AM <jtimberman> i love this

11:40 AM <holoway> that looks pretty rad

11:40 AM <@jonlives> yeah, big +1 on this one too

11:40 AM <holoway> kallistec - that's your subsystem in author land  
11:40 AM <holoway> kallistec: how you feeling?  
11:40 AM <lamont\_oc> yeah +1 on that  
11:41 AM <@jonlives> ranjib: did you see my comment about clarifying handlers vs event subscribers?  
11:41 AM <kallistec> I like the intent  
11:41 AM <jtimberman> coderanger: `preposition`  
11:41 AM <@ssd7> I have implementation questions, but the idea looks awesome  
11:41 AM <holoway> I have questions about 'on'  
11:41 AM <holoway> maybe make it on\_event  
11:41 AM <miketheman> definition of 'is'  
11:41 AM <kallistec> implementation-wise, `on :run\_start` is impossible in a recipe though  
11:41 AM <holoway> miketheman: precisely  
11:41 AM <coderanger> holoway: :-)  
11:41 AM <thom> yeah, in terms of idea i am +all\_the\_things  
11:41 AM <jtimberman> on\_run\_start, on\_run\_failed?  
11:41 AM <ranjib> jonlives, yup. i'll update the text  
11:41 AM <@jonlives> ranjib: cool thanks :)  
11:41 AM <jtimberman> make them explicit?  
11:42 AM <kallistec> how about instead we just adopt a something.d/ convention  
11:42 AM <kallistec> we can have the DSL  
11:42 AM <miketheman> I don't think I understand where these would be placed in a recipe vs registering handlers currently?  
11:42 AM <kallistec> and chef can automatically use any handler stuff in there  
11:42 AM <holoway> (honestly, you should probably write this in the config file if you want it early enough)  
11:42 AM <jtimberman> miketheman: have you used the chef handler cookbook?  
11:42 AM <miketheman> jtimberman: every day  
11:43 AM <kallistec> yeah, if we add a config\_files.d convention though, it'll be easier to manage  
11:43 AM <miketheman> and every datadog run uses it too  
11:43 AM <holoway> ok - kallistec and ranjib hash out the syntax and whatever small implementation details we need  
11:43 AM <kallistec> e.g., you can cookbook\_file it  
11:43 AM <holoway> then this group will +1 and merge next meeting, because we're all thumbsup on intent  
11:43 AM <@ssd7> +1 on some .d/ convention. Lot's of our extension points require it  
11:43 AM <kallistec> if this only exists in the recipe DSL  
11:43 AM <coderanger> kallistec: I think the idea is to keep it in the cookbook itself  
11:43 AM <kallistec> half the features can never be used  
11:43 AM <coderanger> kallistec: Simplifies deployment  
11:43 AM <holoway> yeah, we're agreeing  
11:43 AM <jtimberman> .d convention, that is done in the chef-client cookbook config recipe fwiw.  
11:43 AM <lamont\_oc> i think there might be a something.d convention already in the chef-client cookbook?  
11:43 AM <holoway> it needs to be a new cookbook component  
11:43 AM <kallistec> because the events occur before recipe are evaluated  
11:43 AM <lamont\_oc> (ninja'd)  
11:44 AM <coderanger> kallistec: Fair  
11:44 AM <jtimberman> it's super badass and works well (and i'm biased because i wrote it)  
11:44 AM <holoway> ranjib, kallistec, you both cool with your homework?



11:44 AM <ranjib> yup  
11:44 AM <holoway> ranjib: this is a great idea, thank you  
11:44 AM <@jonlives> +1  
11:44 AM <jtimberman> +long\_int  
11:44 AM <kallistec> yeah, I'll write up a coherent thing in the comments  
11:44 AM <holoway> stevenmurawski: you are acting as secretary, yes? :)  
11:44 AM <stevenmurawski> I am  
11:45 AM <holoway> cool  
11:45 AM <holoway> lets make sure we come back to this one  
11:45 AM <stevenmurawski> Sure thing.  
11:45 AM <holoway> from jonlives - <https://github.com/jonlives/chef-rfc/blob/master/new/rfcXXX-chef-ohai-version-metadata.md>  
11:45 AM <@jonlives> I think this is the best idea ever. :p  
11:46 AM <miketheman> biased  
11:46 AM <jtimberman> i concur  
11:46 AM <coderanger> +bigint again  
11:46 AM <@jonlives> this is basically to preserve lamont\_oc's sanity so we can get stuff in minor releases, and indicate when cookbooks will break without that feature.  
11:46 AM <lamont\_oc> i've been itching to fix this one for years  
11:46 AM <holoway> ok  
11:46 AM <holoway> lets do it  
11:46 AM <miketheman> +1  
11:46 AM <@jonlives> side not, if this is approved, I volunteer to implement it.  
11:46 AM <jtimberman> this will require changes to any tool that uploads cookbooks to supermarket  
11:46 AM <lamont\_oc> now that i'm a chef server developer, it was on my list of shit to fix anyway  
11:46 AM <holoway> what happens when you try and run a cookbook on a version that isn't supported? do we die or warn?  
11:46 AM <@jonlives> holoway: I vote we gracefully die.  
11:46 AM <jtimberman> because these will be NoMethodError  
11:47 AM <coderanger> jtimberman: chef\_version if defined?(chef\_version)  
11:47 AM <ranjib> die  
11:47 AM <@ssd7> I'd vote for die by default, but let me yolo it  
11:47 AM <holoway> we can get around that  
11:47 AM <lamont\_oc> we have to die  
11:47 AM <coderanger> jtimberman: Same way we did use\_inline\_resources for years  
11:47 AM <jtimberman> (see also when we added maintainer repo and issue URLs)  
11:47 AM <@jonlives> I think we have to die too  
11:47 AM <jtimberman> coderanger: yes  
11:47 AM <miketheman> valar morghulis  
11:47 AM <holoway> ok  
11:47 AM <@jonlives> this one is just to add the flags btw  
11:47 AM <@jonlives> what those flags actually do is kind of up for discussion.  
11:47 AM <holoway> let expand the scope to include the behavior  
11:47 AM <@jonlives> ok cool  
11:47 AM <holoway> which sounds like its universally "die"  
11:47 AM <@jonlives> yes  
11:47 AM <holoway> and then we accept  
11:48 AM <@jonlives> great  
11:48 AM <@jonlives> holoway: if I add that, since I'm an editor do I have pre-approval to merge it? :p

11:48 AM <holoway> jonlives: the sylosis record is great. I like it heaps more than previous sylosis records

11:48 AM <@jonlives> or want to review?

11:48 AM <@jonlives> holoway: right?

11:48 AM <jtimberman> <https://github.com/jtimberman/mosh-cookbook/blob/master/metadata.rb#L16>

11:48 AM <holoway> jonlives: lets have one more short cycle

11:48 AM <jtimberman> coderanger: ^

11:48 AM <@jonlives> holoway: gotcha

11:48 AM <coderanger> jtimberman: :-)

11:49 AM <holoway> ok, next agenda item

11:49 AM <holoway> stevenmurawski, want to rock a community update

11:49 AM <stevenmurawski> Sure

11:49 AM <holoway> oh, actually

11:49 AM <holoway> I skipped

11:49 AM <stevenmurawski> webui?

11:49 AM <holoway> yeah

11:50 AM <holoway> so, here is the deal - that code is proprietary, and I don't have a license (yet?) that I can put it under that makes sense for having a public repository

11:50 AM <holoway> mmzyk and nathan smith are working on making the places for filing bugs and friends more clear

11:50 AM <holoway> so the answer is no, we wont' open a repository with that code in it

11:50 AM <c\_t> They just made a repository just for filing bugs and such

11:50 AM <c\_t> seems sufficient to me

11:51 AM <nlloyds> <https://github.com/opscode/chef-manage-issues>

11:51 AM <mmzyk> tis done - we have chef-manage-issues as a repo now, pointers added to project readmes and to the docs

11:51 AM <jtimberman> with a superb email to the list about it

11:51 AM <mmzyk> [https://docs.chef.io/community\\_contributions.html#issues-and-bug-reports](https://docs.chef.io/community_contributions.html#issues-and-bug-reports)

11:51 AM <holoway> however, I am talking to Favorite Corporate Lawyer Ever and CEO about having a license that allows copyright assigned contribution and no right to re-use outside having a commercial contact

11:52 AM <holoway> but that's a very long poll conversation

11:52 AM <holoway> and I worry a bit about the fact that most open source contributors are not sophisticated IP folks

11:52 AM <holoway> so, that's that, unless anyone has questions

11:52 AM <coderanger> holoway: Could there maybe be a \$0 contributor license somehow?

11:52 AM <mmzyk> pretty sure nlloyds is nathan smith, by the way, holoway

11:53 AM <holoway> mmzyk: you're blowing my mind right now

11:53 AM <holoway> (boom)

11:53 AM <holoway> ok

11:53 AM <miketheman> [#nohipchat](#)

11:53 AM <c\_t> holoway: I think there might be some kind of license MS uses for that kind of thing

11:53 AM <holoway> c\_t: yes

11:53 AM <coderanger> holoway: You would have to sign up for it, could be more explicit that it is a license that covers reading the code but not using it?

11:53 AM <@jonlives> coderanger: if you set it up on a dev server with less than 25 nodes, you can use it anyway :)

11:53 AM <nlloyds> blast you mmyzk you've revealed my true identity!

11:53 AM <holoway> you would've gotten away with it, too, if it wasn't for those blasted kids

11:53 AM <holoway> ok

11:53 AM <holoway> stevenmurawski:  
11:53 AM <holoway> take it away  
11:54 AM <stevenmurawski> Alright.. the community team is dead, long live the community engineering team!  
11:54 AM <stevenmurawski> myself, lamont\_oc someara and Mark Anderson (don't have his nick) all report to Nathen as the community engineering team now  
11:55 AM — someara waves  
11:55 AM <stevenmurawski> and we are still figuring out our exact roll in our engineering process, but you'll see us involved in all our open source efforts  
11:55 AM <stevenmurawski> as well as out in the community at events and things  
11:55 AM <stevenmurawski> And that's what I got  
11:55 AM <@jonlives> nice!  
11:55 AM <mary\_grace> and those of us who used to be on the community team ( @cwebber, myself) will still be at many, many events as well :)  
11:56 AM <holoway> cool - the tl;dr is that we wanted to have a team who was responsible for community engineering interaction, and then also got to behave like OSS engineers themselves  
11:56 AM <ranjib> neat  
11:56 AM <jtimberman> this gives me warm fuzzies  
11:56 AM <jtimberman> :D  
11:56 AM <@jonlives> yeah this kicks ass  
11:56 AM <holoway> I already talked about most of what is in the chef inc update that's super relevant to this room  
11:56 AM <@ssd7> (for those that aren't familiar with Chef personalities, the team members also represent people who have touched a broad range of our tech, so I expect to see them contributing in lots of ways)  
11:56 AM <jtimberman> #hugops lives on of course too  
11:57 AM <holoway> I'm pushing hard intenrally to get everything we do delivered daily if there are changes, and released on a steady cadence  
11:57 AM <@jonlives> daily builds would be amazing.  
11:57 AM <jtimberman> active work on that is happening now.  
11:57 AM <holoway> yep  
11:57 AM <holoway> is adamedx here?  
11:58 AM <stevenmurawski> I don't see him in the list  
11:58 AM <miketheman> crickets  
11:58 AM <holoway> otherwise, with two minutes left - I call this meeting adjourned  
11:58 AM <holoway> can I just say I really enjoy the way this group is running Chef  
11:58 AM <coderanger> ChefConf early bird ends soon  
11:58 AM <holoway> yes

---

11:58 AM <@jonlives> holoway: I was thinking today, this whole thing has been a pleasure, and feels like it's working pretty damn well :)  
11:58 AM <@zts> Hope to see some of you folks at cfgmgmtcamp ?  
11:58 AM <holoway> if anyone in this room is going to struggle to get to ChefConf, let me and mary\_grace know  
11:58 AM <coderanger> and if you didn't see the email, speaker notifications are delayed, update on Friday from Nathen