

# Jesse McConnell and John Casey (transcript from IRC)

Jan 19 18:59:06 jmccconnell this is a pretty big change i think, opens up a lot of potential for great things...and abuse, rather like profiles in that respect

Jan 19 18:59:31 jmccconnell people have been asking for a global context for plugin communication for a long time

Jan 19 19:00:20 jcasey bporter: I'm not sure I know what you mean about the project as a way to centralize the info between plugins, other than using properties and static config that's manually edited

Jan 19 19:01:47 jmccconnell jcasey: basically you just have a mechanism for one plugin to say X=FOO and another plugin to get that variable state, right?

Jan 19 19:02:02 jcasey right

Jan 19 19:02:17 jmccconnell ya, thats hugely powerful

Jan 19 19:03:01 jmccconnell the crux is that up til now the only way to do that was through external methods like files

Jan 19 19:03:29 jmccconnell and it makes plugins able to be dependent on other plugins

Jan 19 19:03:42 jmccconnell \_wholely\_ dependent

Jan 19 19:04:14 jmccconnell personally I love it, its just dangerous :)

Jan 19 19:04:35 jcasey right, dependency is possible...but it's possible using files too

Jan 19 19:04:53 jcasey it's just info passing, and you could even do it using sockets...

Jan 19 19:05:07 jcasey you can't keep people from screwing themselves up if they're determined to do so

Jan 19 19:05:16 jmccconnell right, like profiles :)

Jan 19 19:05:26 jcasey well...yeah

Jan 19 19:08:17 jmccconnell the only thing I worry about is how it could poliferate into plugins in the wild

Jan 19 19:08:34 jmccconnell where plugin X needs plugin Y to determine some state

Jan 19 19:09:10 jmccconnell imo there ought to be some kinda registration of that dependency is this is the case, so error reporting can be useful

Jan 19 19:09:38 jmccconnell know what I mean?

Jan 19 19:10:40 jcasey yeah, I think the registration would be in the form of a <dependency/>...

Jan 19 19:11:06 jcasey the way I've been using it is as a DTO of sorts that can serialize/deserialize primitive info to/from the container context

Jan 19 19:11:36 jcasey you have to have the DTO in your classpath, so if you use it as an extension, or have one plugin extend the other where the DTO exists, it works

Jan 19 19:12:19 jmccconnell DTO? sorry, acronym challenged atm

Jan 19 19:12:46 jcasey data-transfer object

Jan 19 19:13:03 jcasey this has always been possible, using \${session} and the core container context, BTW

Jan 19 19:13:12 jcasey I just made it a little more formal

Jan 19 19:13:22 jmccconnell ah

Jan 19 19:13:26 jcasey I've been doing this for quite some time now

Jan 19 19:14:08 jcasey and I've just been careful to make plugins resilient to nulls for this stuff

Jan 19 19:14:09 jmccconnell do you think <dependency> is enough

Jan 19 19:14:41 jcasey for the object, I think so...but if you don't formalize it into a DTO, then you have to be resilient to possible nulls coming out of the context

Jan 19 19:15:03 jcasey you don't have to have inter-plugin deps with this, it's not a foregone conclusion

Jan 19 19:15:17 jmccconnell I would almost be in favor of something like `${globalContext.maven-foo-plugin.value}` so it could complain about the plugin not having the said value

Jan 19 19:16:01 jmccconnell `default-value=${that thing}`

Jan 19 19:16:14 jcasey we could do that too, but remember that the value in context doesn't have to be tied to a plugin

Jan 19 19:16:25 jcasey in fact, many plugins could be used to inject the same value, or read it

Jan 19 19:16:56 jcasey and, the context throws an exception in case the value isn't there, so it forces you to think about it

Jan 19 19:17:11 jmccconnell sure they could, but to make a relationship clear and concise you could argue against allowing that

Jan 19 19:17:40 jcasey sure, but why force the plugin-plugin relationship if it's not a 1:1 mapping?

Jan 19 19:17:55 jcasey if two plugins could put it on the wire, how would that expression look?

Jan 19 19:18:05 jcasey I suppose `${globalContext.value}`

Jan 19 19:18:06 \* jmccconnell is just thinking of ways to limit your new found power and enforcing accountability to a value going missing where one is required

Jan 19 19:18:40 jcasey IMO, it's a good practice to inject it via plugin parameter

Jan 19 19:18:45 jcasey then, you can use `@required`

Jan 19 19:18:49 jmccconnell accountability in such a way that maven itself can complain and let the user know what plugin failed on the chain that getting built up now

Jan 19 19:18:54 jcasey right

Jan 19 19:19:05 jcasey well, sort of...

Jan 19 19:19:16 jcasey that's part of what makes this hard

Jan 19 19:19:20 jmccconnell cause thats really what this is...you are allowing a chain of plugins to execute now

Jan 19 19:19:31 jmccconnell thta carry a context with them

Jan 19 19:19:31 jcasey sure, kind of

Jan 19 19:19:54 jcasey I can also read System properties out of the context without using `System.getProperties` (which is not embeddable, really), and without passing them around explicitly

Jan 19 19:20:08 jcasey that's more powerful than just inter-plugin comm, IMO

Jan 19 19:20:13 jmccconnell and if god forbid you have a chain of 10 plugins in a state map...it needs to be able to pin the one that failed and how

Jan 19 19:20:32 jcasey hopefully, if it really does fail, it should kill the build...that's what the exceptions are for

Jan 19 19:20:35 jmccconnell sure, but no one allows that kinda usage of system

Jan 19 19:20:52 jcasey not sure what you mean

Jan 19 19:20:55 jmccconnell you would be beaten upside the head if you tried it...or I would if I did at least :)

Jan 19 19:21:02 jmccconnell using System for your context  
Jan 19 19:21:02 jcasey :)  
Jan 19 19:21:07 jcasey no, not what I mean  
Jan 19 19:21:10 jcasey the opposite  
Jan 19 19:21:24 jcasey pulling the sysprops, and storing them in a build-level context for reference  
Jan 19 19:21:38 jcasey without reading from System.getProperties() like we are in core in some places now  
Jan 19 19:21:43 jcasey for example, in profile activation  
Jan 19 19:22:03 jcasey Milos has worked around that, but it's passing the properties from hand to hand that don't need them...  
Jan 19 19:22:17 jcasey the project builder, and the profile manager both have to know about them, needlessly  
Jan 19 19:22:57 jmccconnell ok, implementation can be countless ways...but the part that concerns me is the declaration of the dependency  
Jan 19 19:23:49 jmccconnell its not so much a <dependency> of the second plugin on the first, which would imply class file requirements, but its really a <stateDependency>  
Jan 19 19:24:13 jmccconnell which is all I am concerned about having clear  
Jan 19 19:24:15 jcasey I'm starting to think this debate would be better on dev@...but I agree that the globalContext thing works for accounting, though I'm not sure it would help tell you where the value should come from...  
Jan 19 19:24:31 jcasey what if it's not a plugin at all, but an extension?  
Jan 19 19:24:36 jmccconnell so when it really fails it can be reported in a way that you have a hope of figuring it out  
Jan 19 19:24:59 jcasey yeah, a state dep of sorts makes some sense  
Jan 19 19:25:28 jmccconnell and ya, I think this is probably what brett wanted talked about on dev@  
Jan 19 19:25:29 jcasey like "something was supposed to set the global state: systemproperties" ?  
Jan 19 19:25:36 jmccconnell yep  
Jan 19 19:25:49 jcasey I'm not sure how that would help, unless the plugins doc what they set (which they should do, I suppose)  
Jan 19 19:26:03 jmccconnell so if I fail I can blame maven-sparky-plugin for putting me in this unrecoverable state  
Jan 19 19:26:12 jcasey would be cool if we could provide a mechanism for setting state that would track it somehow  
Jan 19 19:26:17 jcasey hehe  
Jan 19 19:26:33 jmccconnell \${globalContext.maven-sparky-plugin.value} would at least tell you the plugin and value  
Jan 19 19:26:49 jcasey right, this is the discussion we should be having. I just wanted to make sure it was feasible before firing off an abstract email  
Jan 19 19:26:50 jmccconnell hm..  
Jan 19 19:27:02 jmccconnell sure thing, I love the idea frankly  
Jan 19 19:27:21 jmccconnell its been the topic of many mails I have read in the last year+  
Jan 19 19:28:00 jcasey yup  
Jan 19 19:28:06 jmccconnell a plugin listener could do it maybe  
Jan 19 19:28:09 jcasey I'll draft something and send it off tonight  
Jan 19 19:28:14 jmccconnell cool  
Jan 19 19:28:27 jmccconnell c&p from here if you want, you have my

permission :P

Jan 19 19:28:52 jcasey I was thinking about a new method, like

```
export(Object key, Object value) on AbstractMojo, but that would still  
leave open the possibility of rogue values like I used to do
```