Jesse McConnell and John Casey on IRC, 01-Feb-2007

```
Feb 01 10:43:01 jdcasey two things came up
in my mind last night
Feb 01 10:43:27 jdcasey first, I was
thinking about something brett mentioned
about this approach for replacement of mojo
bindings violating DRY
[SNIP]
Feb 01 10:44:36 jdcasey I think he's got a
point. I think the declaration for the
replacement target should have to disable
that plugin/mojo/execution before it can be
replaced, otherwise throw an error
Feb 01 10:44:51 jdcasey that way, you can
always look at that plugin decl to see
what's up with it
Feb 01 10:45:03 jdcasey if it's disabled,
you'll see it there...
Feb 01 10:45:18 jdcasey small price for a
lot of clarity, IMO
Feb 01 10:45:41 jesse not following, sorry
Feb 01 10:45:59 jdcasey the other thing I
was thinking is that we may not have a lot
of use for a disable-mojo command...I'd tend
to think only <disablePlugin>true/false</>
and <disableExecution>true/false</>
Feb 01 10:46:02 jdcasey oh, ok
Feb 01 10:46:04 jesse whats getting
repeated?
Feb 01 10:46:16 jdcasey it's not just about
```

repeating...hang on
Feb 01 10:46:29 jdcasey http://en.wikipedia.org/wiki/Don't_repeat_yourself
Feb 01 10:46:56 jesse I like the disablePlugin boolean, +1
Feb 01 10:46:59 jdcasey last sentence in first para
Feb 01 10:47:02 jdcasey cool
Feb 01 10:47:11 jdcasey sorry, second-last
Feb 01 10:47:24 jdcasey "a modification of any single element of a system does not change other logically-unrelated elements"
Feb 01 10:47:49 jdcasey if we have a replace command in one plugin that effectively disables part of another, that could be very difficult to track down
Feb 01 10:47:54 jdcasey you see what I mean?
Feb 01 10:47:58 jesse the whole idea of replacement was because you don't have access tot he upstream definition though
Feb 01 10:48:04 jdcasey hmm, yeah
Feb 01 10:48:10 jdcasey though you'd have to know the executionId
Feb 01 10:48:16 jdcasey right?
Feb 01 10:48:19 jesse yep
Feb 01 10:48:39 jdcasey so, you can re-declare that executionId in your POM, and disable it first
Feb 01 10:48:42 jesse its for undoing something someone did upstream, you said

etrade wanted this
Feb 01 10:48:58 jdcasey the whole point of having executionIds in the first place was to allow merging of execution info during inheritance
Feb 01 10:48:59 jesse that sees like repeating yourself though
Feb 01 10:49:23 jdcasey well, but it's contained in the same plugin decl, and when you run help:effective-pom, it'll show up as one thing
Feb 01 10:49:30 jdcasey hmm
Feb 01 10:49:53 jdcasey it's a little more verbose than I'd like
Feb 01 10:49:55 jesse replace is mergeing the concept of disable
Feb 01 10:50:09 jdcasey disable + addition-with-order
Feb 01 10:50:11 jdcasey yup
Feb 01 10:50:18 jesse actually
Feb 01 10:50:27 jesse its more like overriding a method
Feb 01 10:50:43 jesse and you don't declare the overriding of a method in java or c, you just do it
Feb 01 10:50:59 jesse which would imply not needing replace at all
Feb 01 10:51:05 jdcasey well...it's not a 1:1 analogy, because you could redeclare the execution with config and all in a child POM...that'd be more like overriding, I'd

think
Feb 01 10:51:09 jdcasey but sure, it's similar
Feb 01 10:51:15 jesse just a WARNING that you are overwriting a previous definition
Feb 01 10:51:39 jdcasey you're assuming the executionId has meaning outside of the plugin declaration, which it currently does not
Feb 01 10:51:50 jesse its global to a phase
Feb 01 10:51:50 jdcasey b/c otherwise, there's no way to override from outside that plugin
Feb 01 10:51:58 jdcasey I don't think it is, currently
Feb 01 10:52:15 jdcasey I'm pretty sure it's just a merge target within a plugin declaration
Feb 01 10:52:27 jesse well, that was part of the discussion yesterday, it has to be global to allow smoking with a noop plugin or something
Feb 01 10:52:30 jdcasey (unless something has changed since I wrote it)
Feb 01 10:52:34 jdcasey I see
Feb 01 10:52:49 jdcasey ...but if we have a disable element for executions, we don't need noop
Feb 01 10:52:50 jdcasey right?
Feb 01 10:52:58 jdcasey it's cleaner than noop

Feb 01 10:53:02 jdcasey it's built-in
Feb 01 10:53:18 jesse so long as there is a disable and the default execution of the plugin doesn't happen
Feb 01 10:53:58 jesse which if there is only one exec and you disable it then when its all merged up it will execute the default plugin exec since no exec's would exist...maybe
Feb 01 10:54:08 jdcasey yup...well, that's another thing...the mojos in a lifecycle mapping in components.xml only have an _implied_ execId of 'default' fwiw
Feb 01 10:54:43 jdcasey if it's in the mapping, you should be able to manipulate it using the 'default' execId
Feb 01 10:55:07 jdcasey if it's not in the mapping, and you disable all executions, nothing should run from that plugin...
Feb 01 10:55:16 jdcasey it'll be dormant
Feb 01 10:55:27 jesse my fear here is that this is going to lead to 'libraries' of poms that you would inherit from and then just chose the bits and pieces you want
Feb 01 10:56:08 jdcasey that can't really happen on a large scale unless we allow mixins or multiple inheritance...since you'll have your own parent POM hierarchy...right?
Feb 01 10:56:23 jesse maven is nice because your build is your build and there is one

superpom of conventions..
Feb 01 10:56:27 jdcasey the only thing you could do is inherit from one of those library poms at the head of your hierarchy
Feb 01 10:56:50 jesse company pom -> 3 libraries -> choose one as your parent -> project
Feb 01 10:56:54 jdcasey well, sort of. you still have lifecycle mappings that can change according to which extension you use
Feb 01 10:57:05 jdcasey that exists today, though
Feb 01 10:57:08 jdcasey right?
Feb 01 10:57:31 jdcasey the "library pom" is named for the type of project it represents, or configures
Feb 01 10:57:34 jdcasey like webapp-parent
Feb 01 10:57:38 jesse yes, but the concept of forcing your children to undo what you declare isn't so much
Feb 01 10:58:14 jdcasey no, it's impossible to undo right now, except where individual plugins have a skip param
Feb 01 10:59:18 jesse it feels like we are replacing a unified build philosopy where decisions have cause and effect with one where anything goes, you just grab what you want and disable the rest
Feb 01 10:59:18 jesse ie ant
Feb 01 10:59:43 jdcasey but the reality is that for a large development setup (like, I

imagine, jakarta), you'll want to cover as much of the 80% scenario as you can, and allow enough flexibility for the 20% to make the mods they need to get by...right?
Feb 01 10:59:52 jdcasey it takes the most load off the system that way
Feb 01 11:00:10 jdcasey otherwise everyone always has to add something, because you're using least-common denominator
Feb 01 11:01:00 jdcasey I'm not sure I understand why, though...
Feb 01 11:01:27 jdcasey it's not really allowing just anything...it's just the same as adding a new plugin to the build, except in reverse.
Feb 01 11:01:29 jesse right now the 20% is additive
Feb 01 11:01:46 jdcasey actually, 80/20 isn't approachable
Feb 01 11:02:01 jdcasey because you can't get to the 80%-with-no-extra-config point
Feb 01 11:02:27 jdcasey not without screwing up the 20%, or introducing a multitude of parent POMs
[SNIP]
Feb 01 11:05:44 jesse ok, brett;s issue with replace is what exactly? that you need to redeclare it and then <skip> it and then declare another with the same execiD?
Feb 01 11:05:53 jdcasey at any rate, I was thinking that forcing the user to explicitly

disable the execution before replacing it would make it easier for someone else to come along and understand why that (disabled) execution wasn't running
Feb 01 11:06:12 jdcasey I'm not clear on his exact issue with it, other than it not being a total solution by itself
Feb 01 11:06:21 jesse but doens't replace>execid<> already communicate that?
Feb 01 11:06:33 jdcasey it does, but it's hidden in some other plugin
Feb 01 11:06:36 jdcasey that's the key
Feb 01 11:06:42 jesse it seems totally more consice
Feb 01 11:06:46 jesse concise
Feb 01 11:07:00 jdcasey you're wondering what's wrong with plugin A...and why it's not running...you're not looking at plugin B, which declares a replacement
Feb 01 11:07:04 jdcasey yeah, it is
Feb 01 11:07:13 jdcasey but I'm worried it's not clear enough
Feb 01 11:07:31 jdcasey I'm worried it'll feel like being ambushed when you're trying to debug the build
Feb 01 11:07:51 jesse exec is global to a plugin in a phase then
Feb 01 11:07:58 jesse not global to a phase
Feb 01 11:08:11 jdcasey it's global to a plugin, period, right now
Feb 01 11:08:24 jdcasey b/c an exec can

actually span multiple phases (ick)
Feb 01 11:08:31 jesse if thats the default then having the exec replaced somewhere is irrelvent
Feb 01 11:08:33 jdcasey you just leave off the <phase/>
Feb 01 11:08:56 jdcasey hmm
Feb 01 11:09:05 jesse you don't need replace then
Feb 01 11:09:14 jdcasey why is that?
Feb 01 11:09:20 jesse just disable in the other plugin and declare your new one
Feb 01 11:09:26 jesse funcationally equivalent
Feb 01 11:09:43 jdcasey yeah, but it's important to preserve the ordering of the new mojo with where the old one was bound
Feb 01 11:09:47 jesse execid is a non-issue then
Feb 01 11:09:48 jdcasey that's key
Feb 01 11:09:57 jdcasey it could be the wrong issue
Feb 01 11:10:10 jdcasey it's too damned ambiguous
Feb 01 11:10:12 jesse you would then duplicate the before/after in the new decl
Feb 01 11:10:28 jesse I am cool with this
Feb 01 11:10:59 jdcasey hmm
Feb 01 11:11:00 jesse replace is more concise, but verbose can be a good thing
Feb 01 11:11:08 jdcasey so, how would you

declare before/after, though?
Feb 01 11:11:35 jdcasey damn, why did I allow this crap with executions spanning phases?
Feb 01 11:11:37 jesse hm, execid still needs to be global to a phase
Feb 01 11:11:49 jdcasey and scoped within that phase, IMO
Feb 01 11:11:54 jesse for ordering purposes
Feb 01 11:12:04 jesse yep
Feb 01 11:12:14 jdcasey well, you pair plugin id with exec id, and you don't need global-execId
Feb 01 11:12:36 jesse yI stil think that if your dealing with this complexity its trivial to grep for the execid
Feb 01 11:12:47 jdcasey but when you say "put it just before execA" and execA spans compile, install, and deploy, where do you put it? just before compile?
Feb 01 11:12:47 jesse ah right
Feb 01 11:13:04 jdcasey that's ugly, but I suppose it'd work
Feb 01 11:13:04 jesse _._
Feb 01 11:13:12 jdcasey what's that?
Feb 01 11:13:24 jesse execA spans compile, install, and deploy, where do you put it?
Feb 01 11:13:42 jesse yuck
Feb 01 11:13:58 jdcasey yup, yuck indeed
Feb 01 11:14:00 jesse I don't think ghtat should be possible

Feb 01 11:14:03 jdcasey I hate that we allowed that
Feb 01 11:14:11 jesse that violates the lifecycle
Feb 01 11:14:18 jdcasey you mean order-before something like that?
Feb 01 11:14:22 jdcasey no, what I mean is this:
Feb 01 11:14:34 jdcasey you have three mojos that each declare @phase
Feb 01 11:15:00 jdcasey you can put them in a single execution, leave off <phase/>, and provide configuration, and it'll configure and bind each to the appropriate phase given by @phase
Feb 01 11:15:12 jdcasey <gag>
Feb 01 11:15:13 jesse ah
Feb 01 11:15:39 jdcasey but we have too much ambiguity in this system, and that was a way of handling two locations for phase declaration
Feb 01 11:16:22 jesse k, regardless...all of this needs to boil down into a DAG
Feb 01 11:16:37 jesse with no cycles
Feb 01 11:17:01 jesse and if we decide on a strict naming convention of the nodes of that DAG
Feb 01 11:17:09 jdcasey agreed...and to do any ordering/replacement type thing, you'll need some primary key to go after for the index

Feb 01 11:17:16 jesse we can easily allow for people to shove stuff into it
Feb 01 11:17:18 jdcasey sure
Feb 01 11:17:33 jdcasey the strictest naming convention for that DAG is: g:a:v:e:m
Feb 01 11:17:48 jdcasey that tells beyond a doubt where you are
Feb 01 11:18:05 jesse gavme? or gavem
Feb 01 11:18:10 jdcasey em
Feb 01 11:18:16 jesse k
Feb 01 11:18:21 jdcasey because an execution can have multiple mojos
Feb 01 11:18:37 jdcasey a mojo can be in multiple executions too, but this should work for both, I'd say
Feb 01 11:18:47 jesse so if its replace>gavem</
Feb 01 11:18:49 jdcasey you find that in a lifecycle mapping, and you'll always have a single place
Feb 01 11:18:54 jesse that ought to be clear enough
Feb 01 11:19:24 jesse that replaces the whole node in the graph
Feb 01 11:19:32 jdcasey well, though I still think we need to require that the replacement target should be disabled before it can be replaced
Feb 01 11:19:41 jdcasey yes
Feb 01 11:19:44 jesse adding in the notation to skip that removes the node all together

Feb 01 11:19:53 jdcasey right
Feb 01 11:20:18 jesse and we need a plugin help tool that dumps the DAG out
Feb 01 11:20:24 jdcasey yes
Feb 01 11:20:40 jesse I say skip the skip thing and provide tooling to debug
Feb 01 11:20:49 jdcasey call it help:plan or something...that may be too database-ish, though
Feb 01 11:21:03 jesse its needlessly verbose for the edge case of someone trying to figure it out
Feb 01 11:21:14 jesse help:build-order
Feb 01 11:21:19 jdcasey hmm
Feb 01 11:21:40 jesse IE to do ONE thing I need to change TWO places
Feb 01 11:21:47 jdcasey I understand what you're saying, I'm just not too wild about depending on tooling as the sole means of figuring this stuff out
Feb 01 11:21:54 jdcasey yup, understood
Feb 01 11:21:58 jdcasey *shrug*
Feb 01 11:22:08 jesse doing it that way seems to violate DRY :)
[SNIP]
Feb 01 11:46:41 jdcasey so, we're settled on replace, disablePlugin/disableExecution, and before/after for ordering on straight addition?
Feb 01 11:46:47 jdcasey or, what do you think?

Feb 01 11:46:54 jdcasey I think you're probably right about the replacement stuff
Feb 01 11:47:14 jdcasey two moves is a pita, and we can handle debugging with tools
Feb 01 11:47:16 jesse and the help tooling for dumping the exec DAG
Feb 01 11:47:21 jdcasey just like we do with help:effective-pom
Feb 01 11:47:23 jdcasey right
Feb 01 11:47:27 jesse q:
Feb 01 11:47:32 jdcasey ya?
Feb 01 11:47:44 jesse do we enforce the gavem naming convention?
Feb 01 11:47:49 jesse i say yes
Feb 01 11:48:13 jdcasey that, or we adopt a modified artifact stanza for it, IMO
Feb 01 11:48:27 jdcasey the artifact stanza is more conventional, and might be more intuitive

Feb 01 11:48:28 jesse right right, like everything else