

You have joined the channel

plach has joined (~plach@host70-188-static.61-79-b.business.telecomitalia.it)

[alia.it](https://host70-188-static.61-79-b.business.telecomitalia.it))

Topic: This channel is for all Drupal performance related matters and especially also D8 performance. | <https://groups.drupal.org/high-performance>

Fabianx set the topic at: 17 Jul 2014 14:14

Mode: +cnt

Created at: 01 Sep 2011 14:42

Disconnected

You have joined the channel

plach has joined (~plach@host70-188-static.61-79-b.business.telecomitalia.it)

[alia.it](https://host70-188-static.61-79-b.business.telecomitalia.it))

Topic: This channel is for all Drupal performance related matters and especially also D8 performance. | <https://groups.drupal.org/high-performance>

Fabianx set the topic at: 17 Jul 2014 14:14

Mode: +cnt

Created at: 01 Sep 2011 14:42

EclipseGc has joined (~kris@worx01.worxco.net)

mherchel_ has left IRC (Remote host closed the connection)

Arrow has left IRC (Quit: Leaving)

Disconnected

You have joined the channel

22:07

plach has joined (~plach@host70-188-static.61-79-b.business.telecomitalia.it)

[alia.it](https://host70-188-static.61-79-b.business.telecomitalia.it))

Topic: This channel is for all Drupal performance related matters and especially also D8 performance. | <https://groups.drupal.org/high-performance>

Fabianx set the topic at: 17 Jul 2014 14:14

Mode: +cnt

Created at: 01 Sep 2011 14:42

Disconnected

You have joined the channel

plach has joined (~plach@host70-188-static.61-79-b.business.telecomitalia.it)

[alia.it](https://host70-188-static.61-79-b.business.telecomitalia.it))

Topic: This channel is for all Drupal performance related matters and especially also D8 performance. | <https://groups.drupal.org/high-performance>

Fabianx set the topic at: 17 Jul 2014 14:14

Mode: +cnt

Created at: 01 Sep 2011 14:42

You have joined the channel

plach has joined (~plach@host70-188-static.61-79-b.business.telecomit

alia.it)

Topic: This channel is for all Drupal performance related matters and especially also D8 performance. | <https://groups.drupal.org/high-performance>

Fabianx set the topic at: 17 Jul 2014 14:14

Mode: +cnt

Created at: 01 Sep 2011 14:42

jhedstrom has left IRC (Ping timeout: 265 seconds)

alexpott berdir1, catch, **plach**: <https://www.drupal.org/node/2535082#comment-10130570>

Druplicon <https://www.drupal.org/node/2535082> => Allow hook_update_N implementations to run before the automated entity updates [#2535082] => 23 comments, 5 IRC mentions

plach ★ alexpott: looking

plach ★ alexpott: btw, I was having a look to the critical queue

plach ★ <https://www.drupal.org/node/2535302> does not look critical to me

Druplicon <https://www.drupal.org/node/2535302> => Selecting too many files with JS off causes WSOD with data loss [#2535302] => 11 comments, 4 IRC mentions

berdir1 yeah, not sure about the data loss argument there... the only data you lose is what you submitted from your browser

plach ★ yep, and you need JS off and a lot of files

berdir1 alexpott: yes, that could be a problem, but I don't see a way to solve that?

vijaycs85 has left IRC ()

alexpott berdir1: me neither

berdir1 alexpott: I think we just have to live with that? what happens if you do that? error, endless loop, ..?

alexpott **plach**: I think <https://www.drupal.org/node/2535302> is critical - it's a lost form submission

Druplicon <https://www.drupal.org/node/2535302> => Selecting too many files with JS off causes WSOD with data loss [#2535302] => 11 comments, 5 IRC mentions
dawehner has joined (quassel@drupal.org/user/99340/view)

YesCT has joined (~yesct@50.240.18.178)

plach ★ alexpott: berdir1: in the example alex makes it's unlikely that the entity updates 8001 depend on would be the same that need to precede 8002

plach ★ likely the module itself (or another known module) would introduce those changes

berdir1 alexpott: I'm not sure if that situation could actually happen.. The thing is that due to the way entity updates work (they don't do sequential

updates.. they just go to the current state, if they can) is likely going to mess up what those two update expect the world to be like when they're called

berdir1 which i think is more or less what **plach** said

plach ★ I'm wondering whether <https://www.drupal.org/node/2346013>. could help with that

Druplicon <https://www.drupal.org/node/2346013> => Improve DX of manually applying entity/field storage definition updates [#2346013] => 10 comments, 4 IRC mentions

plach ★ berdir1: yep, more or less :)

plach ★ my point is that if an update function expects an entity update it probably knows the provider

plach ★ so the issue above would be a way to perform only the updates concerning a specific provider

plach ★ an update function could actually run the entity updates for a specific provider and that way could break the dead lock

plach ★ hopefully :)

plach ★ alexpott: berdir1: ^

berdir1 **plach**: well, I guess my point is that if you have such a situation then your existing update hooks are probably not going to work anymore.. so you might have to update your module in two steps

berdir1 **plach**: I can see how that might help if you have two different updates.. but it's not that uncommon to change the same field multiple times

plach ★ right

alexpott afaics this is ver very tricky

plach ★ the problem is that regular updates are isolated and immutable (in theory)

berdir1 **plach**: remember the fun with twitter.module for example that needed like 5 tries to finally get to something that actually worked with the long tweet ID's

berdir1 updates are and always will be very tricky

berdir1 there's only so much that is possible :)

plach ★ while entity updates are not

berdir1 exactly

plach ★ that's why I was wondering whether there were legitimate use cases aside from head 2 head

alexpott **plach**: views updates

plach ★ once the schema is stable, update function should assume the schema is already correct

alexpott **plach**: if any views schema or view plugin's schema changes we have a problem

berdir1 has left ()

plach ★ alexpott: if we notify subscribers after all update functions have been executed, would the views case be fixed?

plach ★ even if updates always happen before?

alexpott **plach**: that might work for this use case

plach ★ that actually solves a class of problems

plach ★ not only this use case

plach ★ since reacting to entity updates is all affected by this but can be fixed the same way

alexpott **plach**: so what about h2h?

plach ★ alexpott: h2h is a problem because the API never assumed needing to deal with intermediate schema changes

plach ★ it only support switching between known states

plach ★ but I think it's the only case we have

plach ★ (I may be wrong, though)

plach ★ alexpott: the solution we suggested in h2h before <https://www.drupal.org/node/2535082> was opened was to use regular update functions with hardcoded schema definitions

Druplicon <https://www.drupal.org/node/2535082> => Allow hook_update_N implementations to run before the automated entity updates [#2535082] => 23 comments, 6 IRC mentions

plach ★ as we would do in the past

plach ★ I think that's fair because that's what h2h is about

plach ★ transitions between known earlier versions of the system

plach ★ * transitioning

plach ★ alexpott: we could keep the logic implemented in <https://www.drupal.org/node/2535082> if it's reliable

Druplicon <https://www.drupal.org/node/2535082> => Allow hook_update_N implementations to run before the automated entity updates [#2535082] => 23 comments, 7 IRC mentions

plach ★ we could add on top of that the ability to defer entity schema update notifications

plach ★ combined with the ability of executing entity updates from regular update functions for a specific provider we should cover most use cases I think

alexpott **plach**: well it's reliable but it makes it hard to make the automatic entity update first

plach ★ alexpott: do we need that?

plach ★ the important thing would be performing entity update notifications after all updates have run

plach ★ alexpott: oh, well, yes, it would e better to ensure that

plach ★ as most (all?) regular updates would require that at that point

- alexpott** I think <https://www.drupal.org/node/2535082> is basically a no go because of the tangle you can end up in
- Druplicon** <https://www.drupal.org/node/2535082> => Allow hook_update_N implementations to run before the automated entity updates [#2535082] => 23 comments, 8 IRC mentions
- plach** ★ alexpott: if we don't do that, then the only fix is deferring entity update notifications
- plach** ★ and crossing fingers ;)
- alexpott** **plach**: I keep on thinking - but then shaking my head - that we might have to provide another update hook - hook_update_before_entity()
- plach** ★ do you mean hook_update_before_entity_N() ?
- alexpott** **plach**: that is completely separate from hook_update_N but duplicating the whole _N system is very painful
- alexpott** **plach**: yes
- plach** ★ alexpott: before going that way (which is totally scary), can we at least try to come up with a use case, excluding the two we already analyzed?
- plach** ★ a third, legit, use case
- plach** ★ IMHO both of them are special cases
- alexpott** **plach**: that feels like a job for catch - he's excellent at coming up with things like this
- plach** ★ alexpott: ok :)
- alexpott** **plach**: changing views config schema is not special
- plach** ★ alexpott: the fact that happens in an entity update notification while other updates are pending is
dawehner has left IRC (Ping timeout: 244 seconds)
- plach** ★ it's a very specific condition we can prevent from happening
- plach** ★ and preventing it is totally sensible
- alexpott** **plach**: and there will never be a situation where an update expects the notification to have fired?
- plach** ★ alexpott: when you start making too many assumptions on the system state in an update function you are in trouble
- plach** ★ anyway, again, we need use cases :)
- plach** ★ for the Views case I don't think so
- plach** ★ the point of those updates is ensuring the various Views are functional after updates have run
- plach** ★ alexpott: what if every single entity update gets an identifier?
- plach** ★ like an hash or something like that?
- plach** ★ and you can target the hash as a dependency
- alexpott** **plach**: yeah I was pondering that too
- plach** ★ so you can say, my update function depend on the "sticky" field being removed

alexpott plach: or has to run before the "sticky" field being removed

alexpott plach: the problem is what happens if it get added back in again

alexpott plach: toosticky?

plach ★ alexpott: :)

plach ★ alexpott: let's pretend entity updates didn't exist for a moment

alexpott plach: I've got to sleep

plach ★ lol

plach ★ alexpott: I will mull on this a bit more, do you mind if I post the chat?

alexpott plach: please do!

alexpott plach++

plach ★ I will

alexpott plach: thanks

plach ★ np

plach ★ thank YOU :)