Deadlock's Compact Loaders

DCL provides players with single tile loaders, unlocked by Logistics research. It also has options to re-style belts and underneathies with new sprites, and can create any colour of belt, underground or loader: these options are off by default but players can turn them on. If you are a Factorio modder and your mod provides new belts or changes vanilla ones, or if you're a player and DCL doesn't already support your favourite mod, you can get DCL to automatically generate matching loaders. It's quite easy to do and you don't need to provide any new graphics or icons for the loaders.

• Step 1

Add "DeadlockLoaders" as a dependency in your mod's **info.json**, so that it sets everything up before your mod, with a question mark if you want it to be optional. For example:

```
{
       "name": "DeadlockSuperBelts",
       "version": "0.1.0",
       "title": "Deadlock's Incredible Super-Grindy Belts",
       "author": "Deadlock989",
       "contact": "",
       "homepage": "",
       "dependencies": ["base >= 0.16.0", "?DeadlockLoaders”],
       "description": "Belts and loaders that are 1/4 the speed of a yellow belt.",
       "factorio_version": "0.16"
}
```

• Step 2

DCL exposes a function which does everything needed. You should call this function from either your mod's **data-updates.lua** or **data-final-fixes.lua**. Test that the function is available, then send it a table of info about your belt tier. That's all there is to it. Here is a made-up example – see also the /prototypes/mods/ subfolder in the mod for working examples.

```
-- [data-updates.lua]
-- this mod makes super-belts. don’t blink.
-- we already created our belts earlier on ...

-- get DCL to generate a matching loader
if deadlock_loaders then
    deadlock_loaders.create( {
        tier = 4,
        transport_belt = "super-transport-belt",
        underground_belt = "super-underground-belt",
        splitter = "super-splitter",
        colour = {r=255,g=100,b=255},
        ingredients = { {"super-plates", 1000}, {"super-oil", 2} },
        crafting_category = "crafting-with-fluid”,
        technology = "super-logistics-4",
        localisation_prefix = "superbelts",
    } )
end
```

• The create() function

deadlock_loaders.create({...}) expects you pass it a table, with some or all of the following key/value pairs. At least three need to be specified, the more the better. Order doesn't matter.

Key	Type	Optional / Mandatory?	Explanation
tier	integer	**Mandatory**	DCL assumes that belts are organised into tiers. It assumes there are at least 3, by default corresponding to the yellow (1), red (2) and blue (3) belts in vanilla Factorio. You can add higher tiers (4, 5 etc.) or lower ones (0), or overwrite existing ones. **But you must do this in a sensible order:** for example DCL will throw an error if you specify 4 then 6. This prevents gaps.
transport_belt	string	**Mandatory**	The name of the transport belt entity in the same tier. Must already exist as an entity prototype in data.raw**.**
underground_belt	string	Optional if ingredients are specified	The name of the underground belt entity in the same tier. Can be nil, but the underground won't be styled, and you must then specify ingredients (see below).
splitter	string	Optional	The name of the splitter entity in the same tier. Can be nil, but the splitter's belts won't be styled.
colour	table	Optional	This is the tier's main colour, used by the loader icons and sprites and the belt styling. It must be the standard RGB colour table that Factorio uses, e.g. {r=255, g=0, b=0}. If nil or not a valid colour, it defaults to the vanilla colour (tiers 1-3 only) or a lovely delicate Barbie pink (every other tier).
ingredients	table	Optional if underground is specified	A table of ingredients, exactly the same as the ingredients table in a recipe prototype, which will define the crafting recipe for one loader of this tier. If nil, DCL will base the loader's recipe on the ingredients used to make the specified underground belt. This means that if you don't specify ingredients, you must specify an underground belt.
crafting_category	string	Optional	The crafting category that the loader's recipe will have. Usually either "crafting" or "crafting_with_fluid", but mods may provide others. If nil, defaults to "crafting". Not used if no ingredients are specified.
technology	string	Optional	The name of the technology that DCL will insert the unlocks for this loader into. For vanilla items, this is logistics, logistics-2 etc. If nil, DCL will auto-enable the recipe so that it is available from game start.
localisation_prefix	string	Optional	The string put in front of entity names in the localisation config. For example, "superbelts" generates localised names of "superbelts-loader-4", "superbelts-loader-5" etc. If nil, defaults to "deadlock".

• More details

Loader speeds. Compact loaders take their speed directly from the corresponding belt prototype, so you don't usually need to worry about it. If some other mod changes existing belt speeds and you want to be certain to catch that change, you need to add that mod as a dependency and call deadlock_loaders.create() after they make their changes.

Auto-ingredients. As described above, if you don't specify ingredients, then DCL will base the loader's ingredients on those of the corresponding underground belt of the same tier. It works like this: (1) Take the underground recipe (2) Replace any underground belts in it with one loader (of the previous tier) (3) Replace any belts in it with one belt (4) Halve the amount of everything else, rounded up. If you don't like this scheme, you have to specify ingredients.

Migrations. DCL can't handle your migrations for you. If you create new loaders and specify an existing technology as the unlocker, and if players are mid-game and have already researched that technology, then they won't be able to unlock the loader. See DCL's migrations folder for an example of how to handle that. If you're unwilling to do migrations then you'll have to create completely new technologies instead.

Locale. You need to provide your own locale strings for the loader names if you're creating any brand new ones that DCL doesn't know about. See DCL's /locale/en/ config for an example. You only need to specify [entity-name], this is re-used for the item and the recipe as well.

Overwriting. If you want to change a vanilla tier (1-3) or change something about another mod's loaders (e.g. ingredients), just overwrite it – see the Xander Mod file in /prototypes/mods/ for an example of overwriting tier 3 and moving express belts to tier 4.

Errors. DCL does some sanity checking on what you asked it to do. If your parameters would cause real problems, it stops the loading process with an error and suggests disabling the mod. Fix your error and restart.

"Helper" or "bridge" mods. If your favourite modder doesn't want to or have time to provide support for compact loaders, you could make your own mini "helper" mod which simply bridges the gap. In this case you would require both their mod and DCL as compulsory dependencies, and then all your mini-mod does is loop through their belts in its data-final-fixes.lua. If your code is simple, like, on the order of the things in the /prototypes/mods/ subfolder kind of simple, and it makes sensible decisions, I'm willing to include it in the main DCL mod.

If you run into problems, use the Factorio forums thread to contact me.

- Deadlock989