Deadlock's Crating Machine

DCM is a Factorio mod which packs items into crates in order to boost belt throughput. If you are a Factorio modder, you can get DCM to automatically generate crated versions of items from your own mods, or any vanilla item which you think needs stacking.

• Step 1

Add "?DeadlockCrating" as an optional dependency in your mod's **info.json**. For example:

```
{
	"name": "DeadlockTweaks",
	"version": "0.1.0",
	"title": "Deadlock's Tweaks",
	"author": "Deadlock989",
	"contact": "",
	"homepage": "",
	"dependencies": ["base >= 0.17.0", "?DeadlockCrating],
	"description": "Some small quality of life adjustments.",
	"factorio_version": "0.17"
}
```

• Step 2

DCM exposes a set of functions which handle crate creation (and optionally machine tier creation) for you. You should call these functions from your mod's **data-final-fixes.lua**. Check that **deadlock_crating** exists before you try and call any of the functions, so that your mod will still run without errors if DCM isn't installed. For example:

```
-- [data-final-fixes.lua]
-- this mod makes diamonds. don’t dig straight down
-- we already created our items earlier on

-- get DCM to crate my itamz
if deadlock_crating then
	-- repeat this for every item you want crated
	deadlock_crating.create("deadlock-uber-diamond", "deadlock-crating-1", 64)
end
```

Alternatively, if you have a long list of items which you want to insert into different technologies, you could put them in a table and then loop through them.

This is the simplest use case. Several more functions are provided: these are described on the following pages.

• deadlock_crating.create()

This is the function used to crate things that DCM doesn't handle by default. It takes three parameters, with only the first mandatory: **deadlock_crating.create(item_name, target_tech, icon_size)**

Parameter	Optional / Mandatory?	Explanation
item_name	Mandatory	The name of the item you want stacked, e.g. "iron-plate" or "mymod-uber-reactor".
target_tech	Optional	The technology name (e.g. "deadlock-stacking-1") that the recipe unlock effects will be inserted into. If specified, the technology must already exist as a prototype in data.raw. If omitted or nil, the mod won't update any technologies, and you'll have to handle gaining access to the recipes yourself.
icon_size	Optional	The icon size of your base item. Must be 32, 64 or 128. If omitted or nil, this currently defaults to 32, but this default will be increased to 64 when Factorio 0.17 releases the new character GUI and high-res icons.

Note that **if crating recipes already exist for the item name specified**, they are replaced by the new one – but if you specified a different technology to the original's, you will need to clean that up. Consider using destroy_crate() described below to remove the item before you create a new version.

• deadlock_crating.destroy_crate()

This takes one parameter, the item name: **deadlock_crating.destroy_crate(item_name)**

Calling this function will destroy the crated version of the item, **all** recipes which include the crated version of the item as an ingredient or result (i.e. the crating and uncrating recipes at minimum, but could in theory include any other modded recipe that has used a crate as an ingredient), and removes **all** references to the relevant crating and uncrating recipes from **all** technologies. For example, DCM stacks copper cables by default. If you want to stop copper cables from being crated, you would call the function like this:

```
deadlock_crating.destroy_crate("copper-cable")
```

Bear in mind that there could be other mods which expect any given crate to exist. If you destroy them, you're responsible for sorting out any issues that causes. You can also use this function to clear a crated item and then rebuild it with create(), for example to resolve conflicts between two overhaul mods.

• deadlock_crating.destroy_vanilla_crates()

This takes no parameters and removes all of the vanilla crate items and recipes that DCM creates by default. Effectively it calls destroy_crate() on every item the mod sets up in the early data stage. Use this if you want to wipe the slate clean and rebuild all the crating technologies from scratch.

• deadlock_crating.create_tier()

By default, DCM creates three tiers of machine, numbered 1-3, which correspond to yellow, red and blue belts in vanilla; it also creates three tiers of technology which each unlock the machines and a collection of crate recipes. You can use this function to create new tiers of machine unlocked by new technologies which correspond to belts that are provided by other mods (or have some other reason for existing). Or you can use it to replace the default tiers with different settings.

The function takes a table as its single parameter. The table can have the following key/value pairs:

Parameter	Optional / Mandatory?	Explanation
tier	Mandatory	An integer or a string. This is the suffix on machine and technology names. DCM uses tiers 1-3 to create the yellow, red and blue tiers for vanilla. Using an integer (e.g. 4) is easier because DCM usually calculates various properties of the crating process using the tier number, and it's the Factorio convention for chains of technologies anyway. You are allowed to specify a string as the tier if you really need to, but you will then have to specify almost everything else.
ingredients	Mandatory	A table of IngredientPrototypes. The ingredients needed to craft the machine. Exactly the same as the ingredients property in a recipe prototype (see https://wiki.factorio.com/Prototype/Recipe#ingredients).
colour	Optional	A colour table. The tint colour for the machine and its lights. Must use the Factorio {r=…,g=…,b=…} format. Alpha (a=) can be specified but is optional. If this is not specified or nil, DCM will use the default tier colour scheme for tiers 1-3, or pink for any other value. It's spelled colour because that's the way I've spelled it all my life.
pollution	Optional	A number. The pollution units per minute emitted by the machine when active. Defaults to (5-tier) if the tier is a number between 1-3, and to 1 for any other tier value.
prerequisites	Optional	A table containing a list of technology names. The same as the prerequisites property in a technology prototype (see https://wiki.factorio.com/Prototype/Technology#prerequisites). Used to chain the crating tier within the technology tree. Defaults to nil (the technology will be immediately researchable in a brand new game).
upgrade	Optional	A string. Defines the entity/item that the Upgrade Planner will upgrade this machine to by default. Defaults to nil.
energy	Optional for numbered tiers	A string. How much energy the machine uses when active. Uses the same energy value format as other prototypes, i.e. a string containing a number followed by a valid watts unit, e.g. "500kW" or "2.6MW". Defaults to the default DCM scheme for numbered tiers, mandatory for non-numbered tiers.
drain	Optional for numbered tiers	A string. The constant energy drain of the machine, whether active or not. Same kind of energy unit string as above. Defaults to the default DCM scheme for numbered tiers, mandatory for non-numbered tiers.
health	Optional for numbered tiers	A number. The maximum health of the machine. If not specified or nil and the tier is a number, the default DCM scheme will be used. Mandatory if tier is not a number.

speed	Optional for numbered tiers	A number. The crafting speed of the machine. A value of 1 corresponds to a vanilla yellow belt, 2 to a red belt, etc. If you want to match the machine speed with a belt tier whose speed is not an integer multiple of 15 item/s, you need to specify a fractional value (i.e. the belt speed divided by 15). Defaults to the tier number for numbered tiers, is mandatory for non-numbered tiers.
unit	Optional if tier is 1-3	The cost and speed of the technology research. Must be a unit table identical to those specified in technologies with a count or count_formula, time and ingredients (see https://wiki.factorio.com/Prototype/Technology#unit). If not specified or nil, and the tier is a number between 1-3, the default DCM cost will be used. If the tier is outside that range or a non-number, this property must be specified.

Once called, there will be a new machine entity/item/recipe named **deadlock-crating-machine-[tier]** and a new technology which unlocks it called **deadlock-crating-[tier]**. You are then responsible for adding unlocks to the technology, either by crating new items with **deadlock_crating.create()** or otherwise.

Here is an example of creating a new numbered tier which creates a purple machine in tier 4:

```
deadlock_crating.create_tier({
	tier = 4,
	colour = { r=0.75, g=0.0, b=0.75 },
	ingredients = { {"wet-socks",2}, {"old-bread",17} },
	prerequisites = {"deadlock-crating-3", "nuclear-laser-bomb-99"},
	unit = {
		count = 750,
		time = 60,
		ingredients = {{"automation-science-pack",1}, {"logistic-science-pack",1}}
	},
})
```

And an example of a non-numbered tier:

```
deadlock_crating.create_tier({
	tier = "alpha",
	colour = { r=0.75, g=0.75, b=0.75},
	ingredients = { {"iron-plate", 1000} },
	prerequisites = {"deadlock-crating-1"},
	speed = 10,
	energy = "20MW",
	drain = "5MW",
	health = 1000,
	unit = {
		count = 1,
		time = 60,
		ingredients = {{"automation-science-pack",1}, {"logistic-science-pack",1}},
	},
})
```

More details

Tech/migration. DCM use tiers of technology 1-3 to unlock stacking/crating recipes and also the tiered machines themselves. You can specify these as the required tech to unlock your own stacked things if you like (see table below). If you add your items to DSB or DCM technologies and then change them later, or remove anything that they provide, **you are responsible for your migrations**. See DCM’s migrations folders for an example.

Errors. In some cases, you can call the API functions to create an item or a tier with parameters which would not allow the game to load at all. In those cases, the loading process will halt with an error message. **Please read it.** In other cases, you can specify things that would allow the game to load but just don’t make much sense. In those cases, the game will continue to load but the function call may be skipped and an error or warning will be printed in the default game log. So if your crates or machines are missing, check the log.

“Helper” or “bridge” mods. If your favourite modder doesn’t want to / have time to provide support for stacking/crating, you could make your own mini “helper” mod which simply bridges the gap. In this case you would require both their mod and DSB/DCM as compulsory dependencies, not optional, and then all your mini-mod does is loop through some items and maybe create a new tier if the target mod provides different belts.

Item/recipe/entity/technology names. If you want to use crate items in recipes, they have conventionalised prefixes (see table below).

Crate items	deadlock-crate-[item-name]
Crate recipes	deadlock-crates-pack-[item-name] deadlock-crates-unpack-[item-name]
Machine items/recipes/entities	deadlock-crating-machine-[tier]
Technologies	deadlock-crating-[tier]

If you run into problems and have tried to solve them yourself but are stuck, use the Factorio forums thread to contact me. Provide any error messages and show your code – we can’t guess from a vague description.

This mod was originally authored by Deadlock989. It was ported to 0.17 and is maintained by shanemadden with contributions from Deadlock989.

- Deadlock989