

Shulker box tech

- [Box basics](#)
- [Box loaders](#)
- [Box unloaders](#)
- [Box splitters](#)
- [Box mergers](#)
- [Box sorting](#)
- [Temporary Storage \(for bulk\)](#)

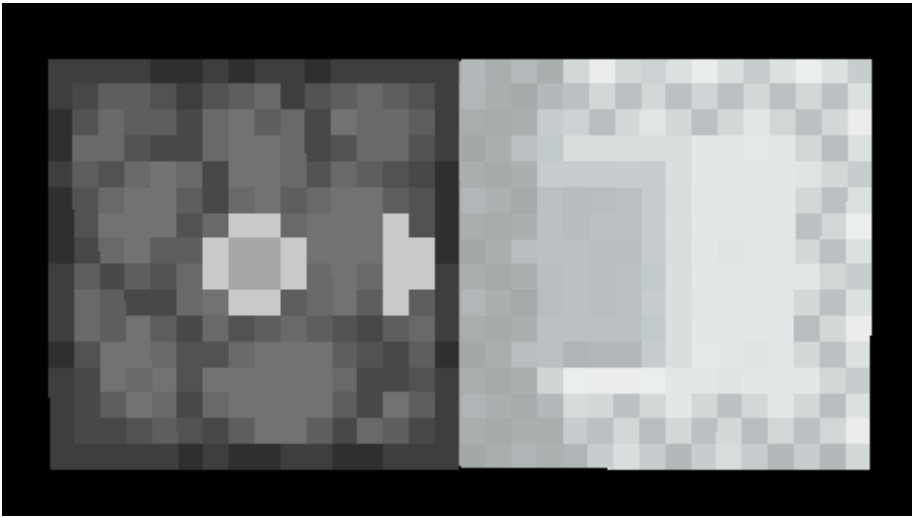
Box basics

All you need to know to start learning about shulker box tech

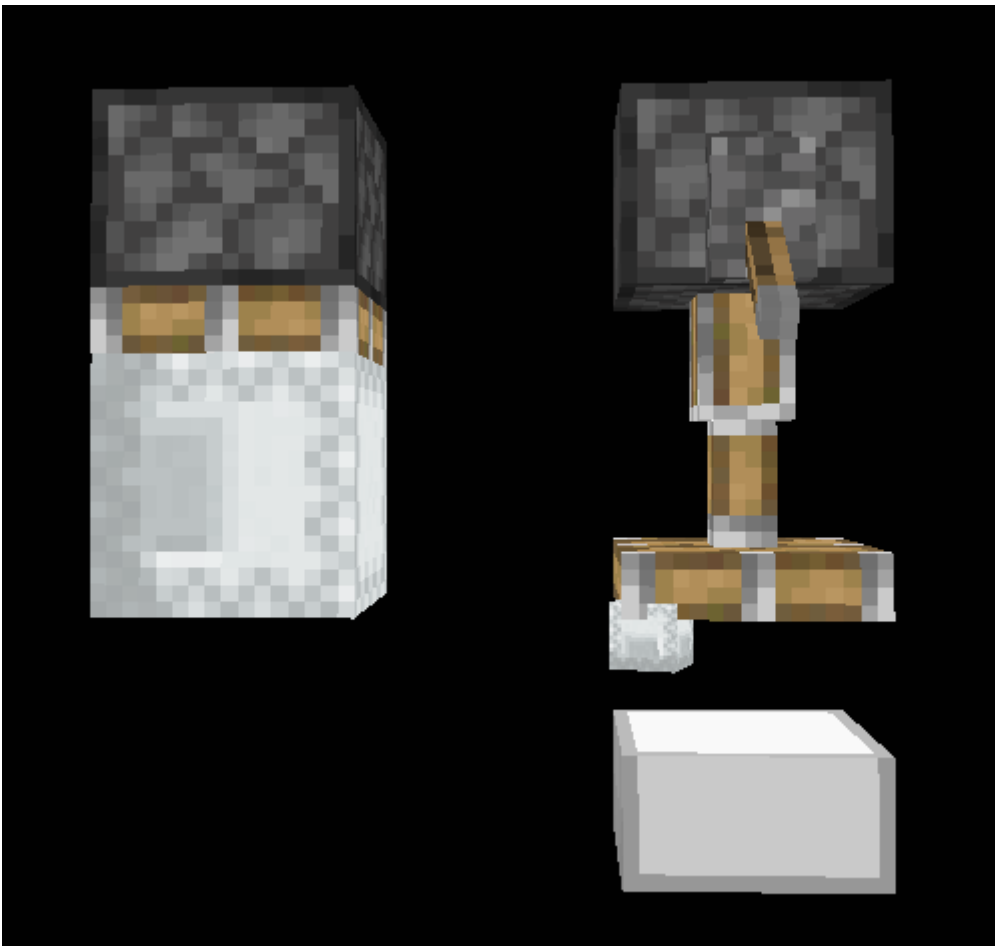


Shulker boxes are the basis of many parts of storage tech. They are crafted from 2 shulker shells (acquired from killing a shulker mob) and a chest. Essentially, they are a movable chest. When breaking a shulker box, its inventory is maintained while in item form, making it an excellent way to transport and store items efficiently.

Sharing the inventory size of a single chest, a shulker box has 27 slots, holding up to 1728 64-stackable items. Shulker boxes cannot be placed inside other shulker boxes, and box items are nonstackable. The most important mechanic with shulker boxes in storage tech is that they may be placed and broken automatically. This is done using dispensers and pistons respectively. When powered while holding a shulker box in its inventory, and given the block on its face is air, a dispenser will place it (or choose one randomly from its inventory if there are multiple) in the space in front of its face.



When a piston pushes or pulls, it will break any boxes that would be in the way of blocks it is moving. The box's item is dropped, and moves along with the piston's pushing direction (assuming the block that broke it's hitbox is touching it), ending with a random momentum as an item.



Box loaders

Box loaders are a common storage tech contraption that takes in items and "loads" them into a shulker box using hoppers or droppers. They can be used to compact items into smaller spaces and make handling items easier for both the player and storage systems.

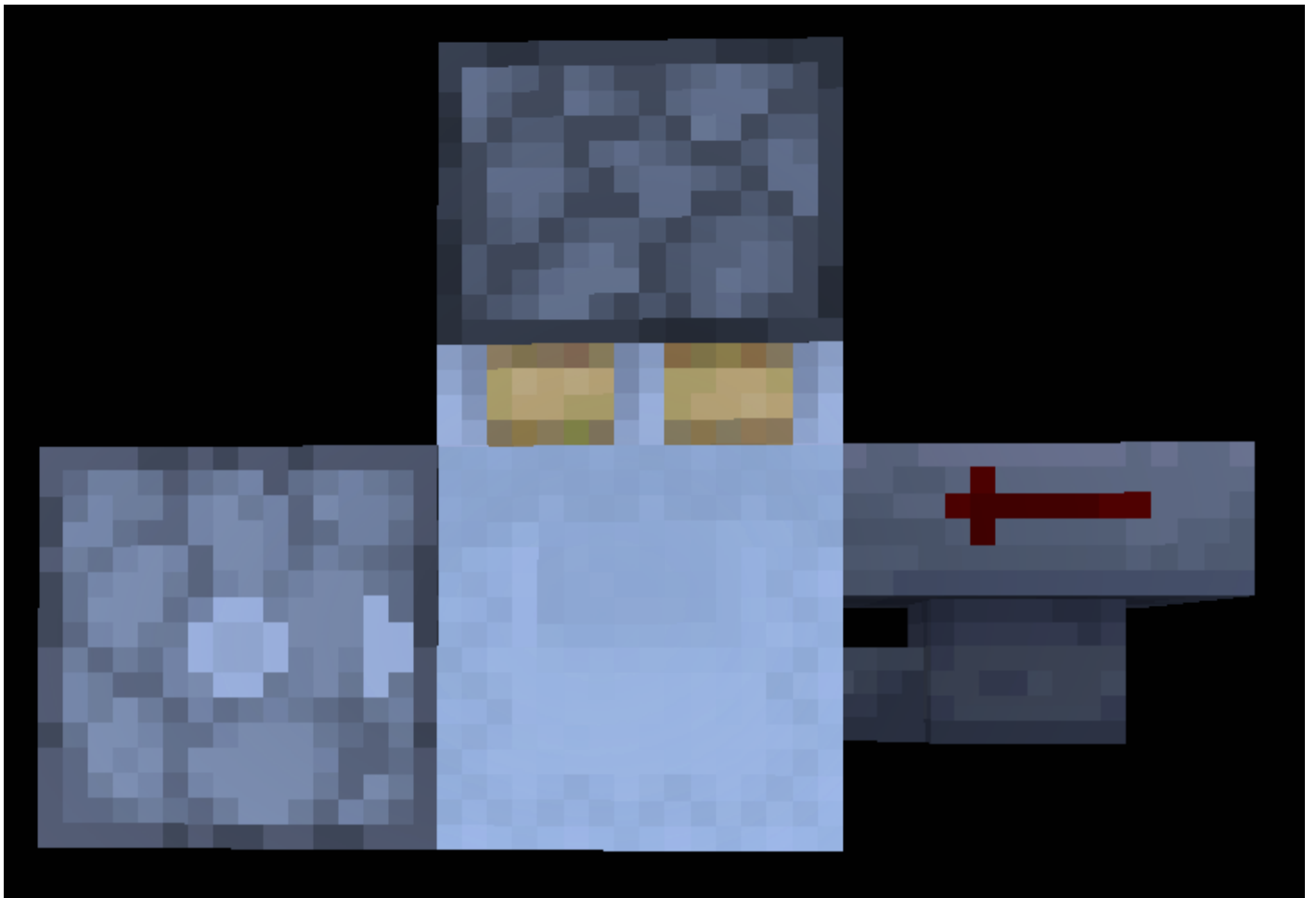
Fundamental Mechanics:

It is useful (and often required) when designing loaders to know how filters work. Please see [this page](#) for an in-depth explanation of item filters. Second, you need to know how shulker boxes are automatically placed and broken in the world. [This page](#) explains that.

For a box loader to function, there are three components that need to interact with the box to be loaded.

1. A dispenser to place new empty boxes when one fills up,
2. A hopper/dropper to fill the box with items from the loader's input,
3. A piston to break the box once it fills up.

The image below shows a common layout for a single-hopper speed loader that includes all three of these components



Box unloaders

A Box unloader is a contraption used to take items out of a box. They are used to move items somewhere else like a chest. They are also used in storage inputs to send items around a storage. There's lots of different types of unloaders such as:

FITs (First Item Type) a unloader that only unloads the first Item in the box.

shulker yeeters a system that unloads a box instantly but burns the actual box they are not used very often for that reason.

arrays these are multiple unloaders that run in parallel to increase item output commonly found in bigger storages. They have some extra logic to make sure each unloader is unloading a different item type.

SSUs (sequential shulker unloader) this type of unloader has a filter for the items so each unloader only unloads one item. If the box doesn't contain that item it will be moved into the next unloader and so on until it finds the right unloader. This type of unloader is very useful because they can unload in parallel to increase speed.

Box splitters

A box splitter splits the items in the box into non-stackables and the other items are loaded into boxes. The loaders replace the box whenever the item changes, so you end up with lots of boxes with a single item type. And the non-stackables are sent down a separate water stream

Box mergers

A box merger takes in 2 boxes of the same item type and combines them into one or more boxes. It will place the most full box out of the two and unload the least full. What it's doing is unloading a box into a loader. The loader box will either get loaded as normal and you send up with a partly full box and a empty box, however if it completely fills it will break both boxes and you end up with 1 full box and a partly full box.

Box sorting

Box sorting is very powerful especially when used in the correct way but first what is a box sorter?

box sorting works by seeing what is in the first slot of a box and taking one item out. It will then send that item and the box into two separate item transportation lines. The item will be sent over a ton of filters and if the item is picked up by the filter the box will be inducted into the slice. This is timed perfectly so that the correct box is inducted. The item is then returned to the box.

note: there's some types of box sorters that can change their filter item these are known as SVARs

you can also get a box sorter that can sort boxes based on if they have a specific item or not these are known as box searchers. The difference is the box is placed on top of a filter instead of a normal hopper. If that filter is triggered it will redirect the box into a item to box rerun **note:** because it's placed on a filter it can check all 27 slots not just the first

this is incredibly powerful because you can sort 1728 items in one go with a 8gt sorter that's **15.5 million items per hour**. These are commonly used in bulk storages. Or to filter out unwanted boxes like ones that include filter items that you use in a storage

Temporary Storage (for bulk)

The following page does not apply to loader-based bulks nor any bulk that accepts loose items (as opposed to full shulker boxes) as input.

Temporary "temp" storage is a front-logic component for a merging array (see the Mergers page). Temporary storages will hold up to one partially-filled shulker box of each item type assigned to the bulk section of a storage. It is also an option for non-bulk to be stored in temp in the case of "smart" encoded halls that place shulkers in chests that are half full. Temp storages must hold this box because mergers cannot do anything with only a single box of an item type, so the temp will hold a box until it receives an box of duplicate item type from the output of the bulk whitelister, which sorts boxes of chest hall/MIS/NS from boxes of bulk items.

There are 2 common ways that temporary storages are set up, depending on the sorting method:

Encoded Temporary Storage (AKA Disc Drive)

In encoded systems, each item type is assigned to a binary or hexadecimal "code" which informs the system of where to store the item type. In encoded temps, this code is used to locate the box into the silo, which has a slot for each item type assigned to merging (note again that encoded systems sometimes allow chest hall items to be routed through mergers). For speed and convenience reasons, the slots are split across different inventories, typically barrels containing 16 slots each.

EX: an 8 bit/2 hexit temp allows up to 256 item types. 4 bits/1 hexit can be utilized to select a single barrel out of 16 in a row, and the remaining 4 bits/1 hexit can be used to select one of the 16 slots inside the barrel. 16 barrels, 16 slots each, $16 \times 16 = 256$ total slots.

When not holding a box, these slots are filled with an unstackable filler item (usually water bottles, although some use water buckets to store unused filler items at 16x efficiency in the form of empty buckets).

The machine first receives a partially full, single-item-type box along with its code. A decoder selects the matching inventory, and a variable clock removes as many items as needed from the inventory to access the selected slot. The last item removed is put through an isBox sorter (see nonstackable sorting). If the slot contained a box, then the inputted box and previously stored box will be ejected to the input of the shulker box merging array. The now empty slot is then replaced with a filler item. If the slot contained a filler item, the filler item is replaced with the inputted box. All removed items are returned to the matching barrel in the same order they are removed. The temp is now ready to receive another shulker box/code.

Non-Encoded Temporary Storage

Non-encoded systems have no way of identifying the item type inside of a box without a filter, so the mechanism is more complex than an encoded system's temp. However, the explanation is shorter because there are far fewer steps. Boxes without matches are stored inside an unorganized silo.

The machine receives a partially full, single-item-type shulker box. It uses an item in that box to set a variable filter, and then allows all boxes in the silo to be checked against the item type in the filter. If any box in the silo's item type matches the item type of the inputted box, then both are ejected to the shulker box merging array. If there are no matches, then the inputted box will be placed into the silo to be checked against for later box input. The temp is now ready to receive another box.

Non-encoded temps are generally slower than encoded temps, especially if there are a lot of boxes in the silo.