

# **ITEM STORAGE HELP MANUAL (v4)**

## **Starting Out / Intro (Pg.2)**

## **Important Info (Pg.2)**

- Updating Inventory Saves (to work with v4)
- Dedicated Server Support
- Item Stacking / Auto-Stacking
- Encumbrance System
- Text-Backpack and Backpack Slash Commands
- Player Add-On Detection

## **Brick Inventory GUI (Pg.6)**

- Inventory Size
- Access
- Remote Access
- Resaving Deleted Saves

## **Player Inventory GUI (Pg.8)**

## **Backpack Item and Inventory GUI (Pg.8)**

- Dropping / Picking Up Item Stacks

## **Saves Manager GUI (Player) (Pg.9)**

## **Host Manger GUIs (Pg.10)**

- Saves Manager GUI (Host)
- Server Settings Manager GUI
- Items Manager GUI

## **Events (Pg.12)**

- Input
- Output

## **VCE Variables (Pg.18)**

## Starting Out / Intro

The Item Storage add-on allows you to store and manipulate items on your server in various ways, which opens the door to a variety of game modes and experiences that weren't previously possible.

When starting out, first plant the brick that you want to store items in. Next, give the brick a name using the Wrench tool (must be one word, start with a letter and be less than 30 characters). Then, go to your Wrench Events menu and setup the `IS_StoreItems` output event under the Self target (for instance: **onActivate** -> **Self** -> **IS\_StoreItems** -> **Medium**).

Next, apply the events and activate/click the brick; a brick and player inventory window should both pop up on your screen. Simply click an item in your player inventory window, and then click an empty cell in the brick's inventory window; the item should switch to the brick's inventory. That's all there is too it! Your item has been instantly saved to the brick, and you can retrieve it at any time.

Later on, you can get into other features this add-on has to offer, such as events, VCE variable support, dedicated server support, managing saves, setting custom item stack/weight values, item stacking, encumbrance, etc., all of which are explained in this manual.

## Important Info

### Updating Inventory Saves

If you updated from a previous version of Item Storage to version 4, your past brick and backpack inventory saves are no longer supported. However, a script has been included that will automatically convert your saves to work with v4 (it works by giving all saved items a stack value of "1"). Once you start up Blockland, execute **ISConvertSaves()**; in your console (which can be opened with "~") – this script can be executed at any time, as long as you can open the console. **Note:** this will not work if you updated from v1 to v4 (it only works if you previously had v2 or v3). It's strongly recommended to convert your saves as soon as possible, especially before using the add-on.

### Dedicated Server Support

Version 4 introduces support for using the Item Storage add-on on dedicated servers, including RTB servers. When hosting the add-on on a dedicated server, it automatically loads previous item / server settings for the add-on that you may have saved (or uses default values if not present). Settings are loaded and applied the first time a player joins your dedicated server, and can be overwritten if you join the server and change them via the Server Settings Manager GUI or Items Manger GUI. **Note:** as of v4, the add-on now takes disconnecting from servers into account (i.e. if you're host on one server, but then join another in which you're not host, it adapts the GUIs to prevent issues).

## Item Stacking / Auto-Stacking

A new feature included with this add-on is the ability to stack items into a single cell; both the brick and backpack inventories support this. To stack items, simply select the item you want to transfer, and then select another stack of the same item. The first selected stack will merge with the second into one stack, depending on the amount of items in both. Once you have a stack of items selected, you can move the entire stack, swap it with another stack, or transfer a specified amount from it. The general rule of thumb is that you left-click your mouse onto a cell to move the entire stack, or right-click (onto an empty cell or onto a stack of the same item) to bring up a GUI (Transfer GUI) that will allow you to select the amount you want to send. However, the add-on takes many different situations into account, and will adapt accordingly. When transferring stacks to your Player Inventory window, remember that only one item can be stacked per cell (for that window only).



You can only store a certain amount of items per stack, depending on the limit for that particular item set by the server host. When an item is present within an inventory, you will notice a small, black bar with two numbers. The number in white is the amount of items within that stack, and the number in gray is the stack limit set by the host for that item. **Note:** The server host can set custom stack limits for each item using the Server Settings Manager GUI, accessible via keybind; the default value is 16.

You can also auto-stack items within the brick and backpack inventories. Auto-stacking is a way of easily/quickly grouping as many like-items as possible into a single stack. Quickly double-clicking an item (within about  $\frac{1}{4}$  of a second) will auto-stack all like-items into that cell, depending on the item's stack limit.

## Encumbrance

Encumbrance, in short, is a limitation added to players that allows them to only carry a certain amount of items at a time, based on item-weight. If enabled, items in both the Player and Backpack inventories will count towards a player's overall encumbrance, relative to the amount of items in each stack and weight values for each individual item. All players are assigned the same max weight amount they can carry (default is "500"); if they exceed this amount, they won't be able to move until they reduce the burden (they go into orbit-cam mode).

Picking up/ dropping a single item or a stack of items alters your weight-burden in real-time, as does using events, moving items around in the inventory windows, etc. Also, if you pickup or drop a backpack item, your burden will be affected not only by the backpack item's weight, but also by the weight of all items within in, relative to the amount of items stacked (if the item requirement is disabled by the server host, backpack encumbrance is always applied to the player, whether they have the backpack item or not). Encumbrance is instantly applied or removed when spawning/respawning, teleporting using F7, transferring items, and when enabled/disabled by the host using the Server Settings Manager GUI. The host can also set custom weight values for each item, as well as the max weight amount for players, using this GUI (the default weight value for items is "10").

The weight value for a single item will be displayed when a stack of items is selected, if encumbrance is enabled. The weight symbol will be also displayed<sup>1</sup>; a custom symbol/unit can be set by the host, although it doesn't have any real purpose except for giving players a standard of reference. **Note:** your encumbrance will be displayed in the Backpack Inventory GUI. Also, be advised that when using the admin orb while encumbrance is enabled (and depending on backpack settings), the backpack GUI will open every time you right-click.

## Text-Backpack and Backpack Slash Commands

If players would prefer not to download the add-on, an auxiliary method for using the backpack has been included. A player's backpack inventory can be displayed in text-format, using the default message box; items can be transferred between the backpack and player/tool inventories using the **/ISTrans InvA CellA InvB CellB Amount** command in chat. With this command, you can also divide item stacks, combine stacks, auto-stack items and drop stacks, just like if you were using the GUI.

When using the `/ISTrans` command, replace `InvA` and `InvB` with the inventory names you are transferring stacks between, which will either be BackPack or Player; Amount is the amount within the first stack you want to transfer (a value of `-1` will transfer the entire stack). Replace `CellA` and `CellB` with the cells you are transferring stacks between, starting with `0` (if one inventory is set to Backpack, the cell would be between `0` and `14`, if set to Player, the cell would be between `0` and the number of max player tool slots subtracted by one, which is usually `4`).

If you want to drop an item, `InvA` should be Backpack, `InvB` should be Drop, `CellA` should be the cell you want to drop items from, `CellB` should be `0`, and Amount should be `0`. If you want to auto-stack items within a cell, both `InvA` and `InvB` should be Backpack, `CellA` and `CellB` should be the same (the specified cell with the item to stack items to), and the Amount should be `-2`. **Note:** after using the `/ISTrans` command, the text-backpack will automatically refresh and reopen to show the changes (it takes half a second to reopen in order to avoid failing the spam filters). Besides using `/ISTrans`, the text-backpack must be manually reopened to refresh.

#### Examples:

```
/ISTrans Backpack 0 Player 4 1
```

```
/ISTrans Backpack 0 Backpack 14 16 (transfers 16 items from cell 0 to 14 in backpack)
```

```
/ISTrans Backpack 0 Backpack 14 -1 (transfers entire stack from cell 0 to 14 in backpack)
```

```
/ISTrans Player 0 Backpack 8 1
```

```
/ISTrans Player 0 Player 2 1
```

```
/ISTrans Backpack 12 Drop 0 0 (drops an item from cell 12 in your backpack)
```

```
/ISTrans Backpack 1 Backpack 1 -2 (auto-stacks items to cell 1 in your backpack)
```

Your text-backpack can be displayed by using `/ISShowBp`, and your encumbrance can be displayed by using `/ISShowEnc` (although, your encumbrance is also displayed in the text-backpack window). When your text-backpack is displayed, it'll layout the cell numbers (starting with `0`), and list present item names, stack amounts and stack limits. For an overview of the commands, players can type `/ISHelp` in chat. It might help to notify players of these commands on your server if desired; disabling event restrictions is also useful with these commands (i.e., so players without the add-on can still transfer items from their backpack or tools menu to the brick), explained later on under the Server Settings Manger GUI section.

**Note:** these commands can be used whether players have the add-on or not. Also, these commands depend on server settings set by the host (i.e., if backpack storage is disabled, you won't be able to view the text-backpack).

## Player Add-On Detection Script

Certain functions will automatically check if a player has downloaded the Item Storage add-on once they spawn, as well as check if they have an outdated version. If a player tries to use server commands or events, but they don't have the add-on downloaded (or have an outdated version), they will be notified to download the add-on from RTB and will be restricted using them.

However, another new feature with this add-on is the ability to disable event restrictions. If disabled, players will still be able to use certain features of the add-on, explained in detail under the Server Settings Manger GUI section of this manual.

## Brick Inventory GUI



The Brick GUI allows you to store, move or retrieve items from a brick's inventory (items can be stacked/moved/swapped by selecting cells within the brick, backpack or player inventory windows, or even with events). Any changes made to the brick's inventory window will also be applied instantly to the GUIs of other players viewing that inventory.

On the Brick GUI itself, you will notice 4 different push-buttons, as well as a number below the inventory grid. That number displays the current number of players currently viewing the same inventory (it's updated every time someone opens or closes their GUI). The **Close** button closes the inventory GUIs, the **Help** button gives you a quick description of the GUI, and the **Saves** button opens your saves manager (used for deleting your past saves, covered below).

The colored button on the bottom left of the GUI, which usually is red and reads "Private" is the **Access** button. The access button shows all players (including yourself) what the current permission is set to for that inventory; clicking it will open up another window within the GUI with six different buttons. The four buttons at the top alter the brick's access value, the blue button under those four sets the remote access value, and the last button hides the window. These values are described in detail below. **Note:** Only the server host or the owner of the inventory save can open this window and change the access settings; to everyone else the button will be faded out and disabled. Also, the first person to save an inventory becomes that inventory's owner (make sure you save before someone else does).

## Inventory Size

When setting up a brick's inventory, you are given the option of selecting 4 different inventory sizes: **Small**, **Medium**, **Large** and Extra Large (or **X-Large**). Small sets the brick's inventory to 4 items, Medium to 9, Large to 16 and X-Large to 32. This feature can be used to add a sense of realism to your storage bricks; for instance, the inventory size could be relative to the size of the brick.

## Access

The Access value for a brick is the restriction that blocks certain players from being able to view or alter the brick's inventory (whether through the GUIs or events). Setting this value to **Private** will prevent anyone except the save owner and the server host from accessing it; this is also the default value when a new save is created. Setting it to **Public** will allow everyone to access it. Setting it to **Admin Only** will prevent anyone except administrators from accessing it, and likewise for **Super Admin Only**. The server host will be able to access any inventory, despite the access value set for it.

If you accidentally set the access value to Admin Only or Super Admin Only, and you are not admin/super admin, you can still change this value back to Private or Public. If you don't, you won't be able to alter the brick's inventory nor reopen the inventory GUI once closed.

**Note:** Changes to Access settings are applied immediately to the brick's inventory and to all players viewing it. Also, a brick's inventory must be saved before you can change access settings for it.

## Remote Access

The Remote Access value allows or prevents accessing a brick's inventory from other bricks. When remote access is disabled, it will not only prevent players from viewing/accessing that inventory from other bricks through the Brick Inventory GUI (even if it's set to Public), but will also prevent them from using events on it.

If the host or save owner, however, tries to access a brick's inventory from another brick while Remote Access is Disabled, it will give them the option of setting the new brick as the **Source Brick**. The Source Brick is main brick in which the inventory was first saved to; if Remote Access is disabled for a brick's inventory, you can still open/alter it as long as that

brick is the Source. If the Source Brick is changed however, the new brick will be set as the Source, and the inventory can then be accessed from that location. However, you will no longer be able to access it from the old location, unless you change it back.

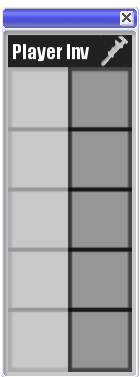
**Note:** When creating a new save, it will automatically enable Remote Access for that save by default. Also, a brick's inventory must be saved before you can change the remote access setting for it. As of version 4, remote access has been improved by referencing a brick's position, rather than its ID. This feature has many uses, but was included mainly for NPC inventories (using the brick as the actual inventory).

## Resaving Deleted Saves

If a brick's inventory save is deleted while you are viewing it through the GUI, and you try to save items to that inventory, it'll give you the option of resaving. If you choose to resave the inventory, it'll carry over the items and access values to the new save, but will now label you as the owner. If you decline, however, and close the GUI, the items will be cleared and the option to resave will be gone.

If you choose to resave, make sure you do so before someone else does. Otherwise, they will become the owner. **Note:** You will only be able to resave if you have access and remote access permission. Also, you will need to close and reopen the GUI (after you resave) in order to unhide the access button at the bottom left (now that you are the owner and can change the access settings). Changes in ownership when resaving are immediately sent to all players viewing the inventory.

## Player Inventory GUI



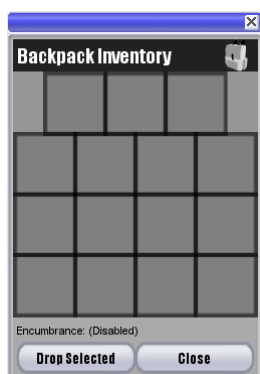
The Player Inv GUI shows all the items in your personal inventory. It also auto-adapts to your max tools slots. Changes made to this GUI will instantly apply to your actual personal inventory. **Note:** support was recently added for items without (or with errant) preview icons; letter-print icons are used when necessary. However, due to how the add-on works, the letter icons may be different between your Player Inventory GUI and actual tools menu, although the item is the same. It's not really an issue, just something to be aware of. Also, picking up or dropping items now auto-updates this window, if open (to prevent cheating/issues).

## Backpack Item and Inventory GUI





The **ISBackpack** item allows you to use an additional personal inventory system, the Backpack GUI. Once a backpack is equipped, you can store, move and retrieve items from the GUI at any time (accessible via keybind – which can be setup in your Options menu - or by activating it in your tools menu). It also gives you the option of dropping item stacks from its inventory to make room for other items.



Adding the backpack item to your inventory (either manually or through events) - while accessing a brick's inventory - will automatically open the backpack window, if server settings allow (likewise for removing the item). **Note:** if the backpack item is added to player inventories through the Minigame Menu, it will not actually unlock the inventory menu for them; each player will need to drop and pick it up in order to refresh it. However, the backpack automatically checks if the item present in a player's inventory when they spawn, and updates accordingly; if you add or

remove the backpack to each player's inventory whilst in a minigame, all you need to do is reset the minigame and it should update/unlock the backpack inventory for players with it. Limit of one backpack save per player.

## Dropping / Picking Up Item Stacks

With the inclusion of item stacking, you can now drop entire stacks of items from your backpack. Once a stack is dropped, it will be handled as one physical item, with the amount of items within that stack shown above. If you don't have a similar item in your tools menu, you can pick up that stack; one item will be added to your player inventory window / tools inventory, and the rest will intelligently stack themselves in your backpack (if present). When added to your backpack, the stack will try to add as many items as possible to existing stacks of the same item, and then stack the rest in empty cells. If some items can't be added, the remaining amount will be redropped in a new stack.

## Saves Manager GUI (Player)



The Saves Manager GUI allows players to list and remove their past saves, which can be useful if you are close to the player-save quota limit (covered below). Simply click the Load/Refresh button to list your saves (as well as to display the current player quota limit set by the host), select the save you want to remove,

then click the delete button to remove it. **Note:** Up to 50 saves can be listed at a time. This GUI can be toggled by setting up a keybind in your options menu; the server will automatically decide whether to open the player or host GUI for a player.

## Host Manager GUIs

### Saves Manger GUI (Host)



options menu.

The host's Saves Manager has the option of not only listing/removing their own saves, but also for saves of other players. Inventory saves can be listed by specific save criteria, as well as in specific files/all files (both of which can be set via the drop down boxes). **Note:** Up to 50 saves can be listed at a time. This GUI (which opens along with the Server Settings Manager, documented below) can be toggled by setting up a keybind in your

### Server Settings Manager GUI



When settings for this GUI are saved/applied to the server, a client-sided save file is created in your "config/client/" folder. For each game instance, previous settings are automatically loaded into the GUI from this file, if present (otherwise default values are used).

**Backpack:** There are 5 options included that affect backpack usage. The first is whether or not to allow backpack usage. If disabled, no one will be able to open or use the backpack, even with events. The second option is whether or not to require the backpack item for accessing the backpack inventory. The third is a toggle option for enabling/disabling encumbrance to all players. The fourth is for whether or not to clear a player's backpack inventory if they die, and the fifth is basically the same, but only applies if a player spawns/respawns.

**Quotas:** Two quota options are included, a player quota and a file quota. The player quota is the max amount of inventory saves allowed per player. The file quota is the max amount of inventory saves per file. Saves are divided into individual files depending on the

save's name; less saves per file puts less stress on the server, depending on certain functions.

**Note:** The file quota also pertains to backpack saves. Also, use caution when maxing out the file quota to 1000 - it hasn't been beta tested.

**Action Against Event-Spam:** Two options allow for preventing event-routine spam. The first option for ("Cancel Event-Spam Routines") will cancel a brick's events if they exceed an output of 25 in 20 seconds. The second option ("Clear Brick's Events If It Continues") (which can be enabled independently of the first option) will remove the brick's events if the event-routines continue and exceed 30 in 20 seconds (i.e. if the events are re-executed by another brick's events).

**Action Against ServerCmd-Spam:** These four options deal with server command spam, specifying the appropriate response to spammers who cross relative redlines set for each function. "None" will only warn the spammer of potential spam, while "Notify Spammer/Server" will warn the spammer (through client message) and the host/other players (through server chat) that the spammer crossed a function's redline. "Kick Spammer" will kick them from the server while "Ban Spammer for (minutes):" will ban them for the amount of time you designate with a preset ban reason. **Note:** It's important to take into account that players may accidentally cross spam redlines.

**Miscellaneous:** The last three options deal with misc. features. The first of these options is for automatically backing up inventory files once saved. Normally, if the server crashes, your inventory saves will remain unaffected. However, if the server crashes while writing a save file, all saves within that file will be erased; since files are constantly rewritten when items are moved around in the inventories, the risk of losing saves is high. When this option is enabled and you save an inventory, the server will first write the original save file, close that file, then use the same information to write a backup file (located in a backup sub-folder within the same directory). In theory, if the server crashes while writing one file, the other file should remain, which you can use to copy data back to the original file (or vice versa) if it fails.

**Note:** This feature also applies to backing up the client-sided server/item GUI settings file, and will work even if you check the checkbox without clicking the apply button. To apply this feature to brick and backpack saves, however, you will need to click the Save/Apply button.

The second option deals with the max encumbrance level per player. In other words, this is the max amount of item weight players can carry before having their movement

restricted. The third option is for toggling event-restrictions; if enabled, players without the Item Storage add-on (or with an outdated version) will still be able to use events (events will even adapt by using default message boxes), except for the `Is_StoreItems` event. If disabled, they will receive an error notification when attempting to use any events for this add-on. The fourth is your preference for the item weight symbol/unit.

## Items Manager GUI



When settings for this GUI are saved/applied to the server, a client-sided save file is created in your “config/client/” folder (the Items Manager and the Server Settings Manager share the same save file). For each game instance, previous item settings are automatically loaded into the GUI from this file, if present (otherwise the server will default to 16 for the stack limits for all items and 10 for weight values).

The Items Manager allows the server host to set custom stack limits and weight values for items. As mentioned above, saved stack and weight values are automatically reloaded for each game instance, relative to the items that are enabled. Also, an **Apply To All** feature has been included at the top of the GUI for quickly setting values for all items. When using this feature, leaving a textbox blank with no space will ignore that column (for instance, if you just want to set all stack values to 999, but don’t want to change weight values, set the first textbox to 999 and the second to “” (without the quotes)).

## Events

### Input Events

**onISGUIOpened** – executed when a player opens the Inventory GUIs using the `IS_StoreItems` output event.

**onISGUIClosed** – executed if anyone closes the Inventory GUIs when the brick’s inv is accessed (even remotely) (use the Client target to output events for the specific client who closed it).

**onISGUIClosedAll** – executed when all players accessing the brick’s inventory (even remotely) close their GUIs (meant for closing the single chest brick).

**onISGUIAddItem** – executed when someone adds an item to the brick’s inventory through the inventory GUIs (useful for GUI-based stores).

**onISGUIMoveItem** – executed when someone moves an item or swaps items within the brick’s inventory through the inventory GUIs (very useful for crafting systems).

**onISGUIDropItem** – executed when someone drops a stack of items from their backpack GUI (only works if the brick inv is being accessed).

**onISGUIRemoveItem** – executed when someone removes an item from the brick’s inventory through the inventory GUIs (useful for GUI-based stores).

**onISGUISwapItems** – executed if anyone manually alters any of the inventory GUIs (such as by swapping items, transferring a stack, etc.).

**onISEventTransfer** – executed if an item is transferred via the IS\_TransferItem output event without error (can be used for transaction messages for stores).

**onISConditionTrue** – executes if IS\_ifItemPresent, IS\_ifInvSameAs, IS\_ifTransferred or IS\_ifItemInCells returns true.

**onISConditionFalse** – executes if IS\_ifItemPresent, IS\_ifInvSameAs, IS\_ifTransferred or IS\_ifItemInCells returns false.

## Output Events

Access and remote access values still apply to the brick’s inventory when using events. Also, make sure the brick’s inventory has an existing save before using events on it (however, using events on a player’s backpack doesn’t require the backpack to have an existing save). Changes to a brick’s inventory through events will update the GUIs of all players viewing that inventory in real time. **Brick Inventory cell-reference below (useful for certain events):**

SMALL (0-3), MEDIUM (0-8), LARGE (0-15), X-LARGE (0-31)

0	1	4	9
2	3	5	10
6	7	8	11
12	13	14	15
16	17	18	19
20	21	22	23
24	25	26	27
28	29	30	31

**IS\_StoreItems** – Opens the Brick Inventory GUI, using the brick's name for the inventory save name.

**Parameters:**

Brick inventory size (MENU)

<input checked="" type="checkbox"/> 12	0	onActivate	Self	IS_StoreItems	X-Large
--	---	------------	------	---------------	---------

**IS\_SetInv** – Overwrites certain cells within the brick's inventory with specified item stacks.

**Parameters:**

Start and End cells (TEXTBOX: Two Words) – specified cells in inventory to add stacks to; overwrites all items within range, if already present.

Specified item to add (MENU)

Toggle to add random items instead (CHECKBOX) – chooses a random item for each stack instead of the item specified; the wrench is automatically prevented from being randomly selected.

Stack amount per item (TEXTBOX) – amount of items per stack (setting it to “0” will choose a random amount for each stack)

<input checked="" type="checkbox"/> 3	0	onActivate	Self	IS_SetInv	0 32	Bazooka	<input checked="" type="checkbox"/>	1
---------------------------------------	---	------------	------	-----------	------	---------	-------------------------------------	---

**IS AddItem** – Adds items to empty cells within the brick’s or backpack’s inventory.

**Parameters:**

Specified inventory (MENU)

Specified item to add (MENU)

Toggle to use random items instead (CHECKBOX) - chooses a random item for each stack instead of the item specified; the wrench is automatically prevented from being randomly selected.

Stack and Item Amounts (TEXTBOX: Two Words) – the amount of stacks to add and the amount of items per stack (“0” for either value will choose a random amount) (If not all the items can be added, it’ll give the player a message notifying them of how many stacks and individual items from last added stack were successfully added.)

<input checked="" type="checkbox"/> 1	0	onActivate	Self	IS_AddItem	BackpackInv	Bazooka	<input checked="" type="checkbox"/>	0 999
---------------------------------------	---	------------	------	------------	-------------	---------	-------------------------------------	-------

**IS ClearItem** – Searches the brick’s or backpack’s inventory for certain items and removes entire item stacks if found.

**Parameters:**

Specified inventory (MENU)

Specified item to clear (MENU)

Amount of stacks to clear (TEXTBOX) - (the player will be notified if no item was found in the specified inventory, but not if the amount cleared is less than the amount requested).

<input checked="" type="checkbox"/> 2	0	onActivate	Self	IS_ClearItem	BrickInv	NONE	<input type="checkbox"/>	1
---------------------------------------	---	------------	------	--------------	----------	------	--------------------------	---

**IS TransferItem** – Transfers an item from one inventory to another (if item is found in former inventory and if there is an empty slot in the latter inventory).

**Parameters:**

Inventories to transfer item between (MENU)

Item to transfer (MENU)

Toggle to transfer any item (CHECKBOX) – if selected, the first item found will be transferred, rather than the item specified (Backpack items will be ignored when transferring items from Brick to Backpack) (This event is very useful for event-based stores and pick-pocketing)

Amount from stack to transfer (TEXTBOX) – (an amount of “0” will transfer the entire stack, an amount of “-1” will transfer a random amount)

<input checked="" type="checkbox"/> 4	0	onActivate	Self	IS_TransferItem	Brick_to_Player	Bazooka	<input type="checkbox"/> 1
---------------------------------------	---	------------	------	-----------------	-----------------	---------	----------------------------

**IS\_ifTransferred** – executes ISConditionTrue or ISConditionFalse depending on the situation from last GUI transfer.

**Parameters:**

Situation (MENU) – if specified item was the first selected, last selected, was added to the brick's inventory or removed from the brick's inventory.

Specified item (MENU)

Any item (CHECKBOX) – if the specified item is any existent item.

Relative conditional event execution (TEXTBOX) – executes specified ISConditionTrue or ISConditionFalse events (leaving this textbox blank, with no space, will execute all ISConditionTrue/False events, depending if true or not).

<input checked="" type="checkbox"/> 9	0	onActivate	Self	IS_ifTransferred	FirstSelected	NONE	<input checked="" type="checkbox"/> 0 1 2 3 4 5 6 7 8
---------------------------------------	---	------------	------	------------------	---------------	------	---

**IS\_ifItemPresent** – Checks if a certain item is present in the specified inventory; executes ISConditionTrue if found and ISConditionFalse if not (this event can be used to check if an inventory is full, empty, or contains at least one item).

**Parameters:**

Specified inventory (MENU)

Item to search for (MENU)

Toggle to instead check if any item is present (CHECKBOX)

Relative conditional event execution (TEXTBOX) – executes specified ISConditionTrue or ISConditionFalse events (leaving this textbox blank, with no space, will execute all ISConditionTrue/False events, depending if true or not).

<input checked="" type="checkbox"/> 8	0	onActivate	Self	IS_ifItemPresent	PlayerInv	Bazooka	<input type="checkbox"/> 0 1 2 3 4 5 6 7 8
---------------------------------------	---	------------	------	------------------	-----------	---------	--

**IS\_ifInvSameAs** – Checks if the brick's inventory is the same as another brick's inventory; executes ISConditionTrue if they match and ISConditionFalse if they don't.

**Parameters:**

Action to perform (MENU) – match item positions, match stack amounts, match both positions and stack amounts or just items.

Toggle to match inventory sizes (MENU) – if enabled, and if the brick's inventory doesn't match the size of the other brick's inventory, this event will automatically return false. Otherwise, if disabled, the brick inventory sizes will be ignored.



String name of other inventory to check (TEXTBOX) - (Very useful for crafting systems)  
Relative conditional event execution (TEXTBOX) – executes specified ISConditionTrue or ISConditionFalse events (leaving this textbox blank, with no space, will execute all ISConditionTrue/False events, depending if true or not).

<input checked="" type="checkbox"/> 6	0	onActivate	Self	IS_ifInvSameAs	MatchPos	<input checked="" type="checkbox"/>	GSFContainerB	0 1 2 3 4 5 6 7 8
---------------------------------------	---	------------	------	----------------	----------	-------------------------------------	---------------	-------------------

**IS\_ifItemInCells** – Same as IS\_ifItemPresent event, except more detailed – allows you to check for a specific stack value and within a cell range; executes ISConditionTrue if found and ISConditionFalse if not.

**Parameters:**

Specified inventory (MENU)

Item to search for (MENU)

Stack amount and cell range (TEXTBOX: Three Words) –\_stack amount to match for specified item and the start/end cells to check within (leaving the stack amount blank or setting it to a value of “0” will ignore matching the stack amount; leaving both the start and end cells blank or setting both to “0” will check the entire inventory, instead of a specified cell range; leaving the end cell blank or setting to “0”, however, will cancel the function and won’t execute ISConditionTrue or ISConditionFalse).

Relative conditional event execution (TEXTBOX) – executes specified ISConditionTrue or ISConditionFalse events (leaving this textbox blank, with no space, will execute all ISConditionTrue/False events, depending if true or not).

<input checked="" type="checkbox"/> 7	0	onActivate	Self	IS_ifItemInCells	BrickInv	NONE	999 0 32	0 1 2 3 4 5 6 7 8
---------------------------------------	---	------------	------	------------------	----------	------	----------	-------------------

**IS\_CopyInv** – Copies items from one inventory to another.

**Parameters:**

Choose to copy items from or to the specified inventory (MENU)

String name of specified inventory (TEXTBOX)

<input checked="" type="checkbox"/> 5	0	onActivate	Self	IS_CopyInv	To	GSFContainerB
---------------------------------------	---	------------	------	------------	----	---------------

**IS\_UndoTransfer** – Undoes the last item transfer/swap action, whether if done through inventory GUIs or events (actions can only be undone once). (It’s highly recommended to set a delay between **33-250 milliseconds**)

<input checked="" type="checkbox"/> 11	0	onActivate	Self	IS_UndoTransfer	
--	---	------------	------	-----------------	--

**IS\_VCESetBrName** – Sets the brick's name to what you specify; supports VCE variables (can be used to open certain inventories relative to each player).

#### Parameters:

Specified name (TEXTBOX)

<input checked="" type="checkbox"/> 10	0	onActivate	Self	IS_VCESetBrName	<var.cl:name>
--	---	------------	------	-----------------	---------------

## VCE Variables

All variables are **<var.cl:NAME>**, replace NAME with a variable name below:

**itemA itemAam itemAsk itemAw invA** (first selected item's: name, stack amount, stack limit, weight, and inventory selected in)

**itemB itemBam itemBsk itemBw invB** (last selected item's: name, stack amount, stack limit, weight, and inventory selected in)

**itemAdd itemAddAm itemAddSk itemAddW invAdd** (item added to brick inventory's: name, stack amount, stack limit, weight, and inventory selected in)

**itemAdd itemAddAm itemAddSk itemAddW** (item removed from brick inventory's: name, stack amount, stack limit, and weight)

**isEnc PIW HasMod BrOpn** (is player over-encumbered, player's encumbrance/weight, if they have the add-on (0=no, 1=outdated, 4=current), is the brick inventory open)

**HasBp BpSvd BpOwrN BpOwrID BpOpn** (if player has backpack item, if backpack inventory is new or has been saved at least once, backpack owner's name, backpack owner's ID, if backpack inventory is open)(Note: some backpack vars might not update if the backpack GUI closes while other inventory GUIs are open; but they should update when the GUIs are refreshed).

**BrNme BrID BrSz BrOwrN BrOwrID BrEditN BrEditID cellA cellB SendAm** (brick inventory name, brick ID, brick inventory size, brick inventory owner ID, owner name, last person who updated brick inventory's name, their ID, first cell number selected, last cell number selected, amount from stack transferred via GUI only(?))

**EncEnab BpEnab BpReq EvtRes maxW WSym** (is encumbrance enabled by host, backpack storage enabled, backpack item required, event restrictions disabled, max player encumbrance, weight symbol)