

# SignShop

*A Powerful and Easy to Use Shop Plugin*

## Quick Reference

*for version 2.2.4*

This reference was made to help you understand some of the more advanced features of SignShop. It assumes you already know what the plugin does and how to set up a shop. You can always go to our Bukkit page at: <http://dev.bukkit.org/server-mods/signshop/> to watch tutorial videos, ask questions, leave comments (we love those), or report bugs. Thanks and enjoy the plugin. ~Devinish & wargamer2010

## Index

[Config Options](#)

[SignShop Blocks and Custom Signs](#)

[Variables](#)

[Using Signs to Run Commands](#)

[Limiting the Number of SignShops](#)

[Using Price Multipliers](#)

[Permissions](#)

[Donations and Sponsors](#)

## Config Options

There are a few basic options to modify how SignShop acts globally. You can edit any of these options and use the command **/signshop reload** to reload the configuration.

**MaxSellDistance:** Maximum distance between a SignShop and chest/lever/etc., in blocks (Infinite/Default = 0)

**MaxShopsPerPerson:** Maximum number of SignShops allowed per person, not including admin (playerIsOp) shops (Infinite/Default = 0)

**TransactionLog:** Whether or not to use transaction logging (Default = false)

**OPOverride:** Allow OP to override permission plugin (Default = true)

**AllowVariableAmounts:** Allow players to sell partial amounts to signs and get partial money (Default = false)

**AllowEnchantedRepair:** Allow enchanted items to be repaired with [\[Repair\]](#) signs (Default = true)

**AllowUnsafeEnchantments:** Enable support for unofficial enchantments (Only to be modified if you know what you are doing. Default = false)

**DisableEssentialsSigns:** Allow SignShop to disable Essentials signs (Only to be modified if you know what you are doing. Default = true)

**EnableSignStacking:** Allow inventories to stack signs in groups of 64 (Default = false)

**AllowMultiWorldShops:** Allow shops with the chest and sign in different worlds (Default = false)

**EnablePermits:** Disables shops for owners without the **SignShop.Permitt** permission node (Default = false)

**PreventVillagerTrade:** Allow players to right-click villagers and trade with them (Default = false)

## SignShop Blocks and Custom Signs

To create customized signs, you can add a new sign name in the config.yml under the "signs:" section. Don't forget to also create the appropriate entries in the "messages:" sections. You can add as many or as few of these Blocks as you like to your sign, separated by commas. These Blocks are activated in order from left to right when the sign is used (which is important for a few of them).

**takePlayerItems:** Takes the items designated during the sign set up from the user

**givePlayerItems{ignorefull}** OR **{oneslot}**: Gives the items to the user, optional parameters to ignore full inventories **{ignorefull}** and to require at least one slot of available space in the inventory **{oneslot}**.

**takeVariablePlayerItems;** Allows the selling of partial and full amounts if **AllowVariableAmounts** is true

**takeItemInHand:** Takes the items in the user's hand from the player

**takePlayerInventory:** Takes all of the items in the user's inventory

**takeShopItems:** Takes the items designated during the sign set up from the shop chest

**giveShopItems:** Gives the items designated during the sign set up to the shop chest

**takePlayerMoney:** Takes the amount of currency on the 4th line of the sign from the user

**givePlayerMoney:** Gives the amount of currency on the 4th line of the sign to the user

**takeOwnerMoney:** Takes the amount of currency on the 4th line of the sign from the owner of the shop

**giveOwnerMoney:** Gives the amount of currency on the 4th line of the sign to the owner of the shop

**setRedstoneOn:** Turns (a) redstone lever(s) on

**setRedstoneOff:** Turns (a) redstone lever(s) off

**setRedStoneOnTemp{#}:** Turns (a) redstone lever(s) on and then turns it/them off after "#" seconds

**toggleRedstone:** Flips (a) redstone lever(s) from on to off, or vice-versa

**playerIsOp:** Makes it so the player needs to be OP or have the **SignShop.Admin.\*** Permission node

**setDayTime:** Sets the time in the current world to the beginning of the day

**setNightTime:** Sets the time in the current world to beginning of the night

**setRaining:** Sets the weather in the current world to rain with the chance of thunderstorms

**setClearSkies:** Sets the weather in the current world to clear

**healPlayer:** Sets the user's health to full

**repairPlayerHeldItem:** Sets the item in the user's hand to full durability

**enchantItemInHand:** Enchants the item the user is holding with the enchantments defined during the sign setup

**disenchantItemInHand:** Removes any enchantments from the item the user is holding

**takePlayerXP{raw}:** Takes the amount of experience from the user defined on the 3<sup>rd</sup> line of the sign, optional parameter to take raw XP instead of XP levels

**givePlayerXP{raw}:** Gives the amount of experience to the user defined on the 3<sup>rd</sup> line of the sign, optional

parameter to take raw XP instead of XP levels

### **Special Blocks**

These affect the Block that comes immediately after them. The order in which they are placed is important.

**Chest{#}**: Makes a Block affect only the chest number replacing “#”

**RandomItem**: Makes a Block select an item at random out of the items designated during the shop setup

[Back to Index](#)

## **Variables**

These are used in the “messages:” “errors:” and “commands:” section to replace a word with a variable. Not all variables can be used in the “errors:” section.

**!items** The items and amounts being bought or sold by the SignShop

**!price** The price listed on the 4<sup>th</sup> line of the SignShop, after any modifications or multipliers

**!enchancements** The enchantments applied to the item, not including the item itself

**!xp** The amount of experience levels listed on the 3<sup>rd</sup> line of the SignShop

**!hasxp** The amount of xp a player has

**!chest#** This is like !items, but only in specific chests (replace “#” with the chest number)

**!customer** The person who bought from a SignShop

**!max** Used only in specific error messages to list maximum distances, amounts of things, etc.

**!player** The user of the SignShop, can be used in the “commands:” section to enact commands on players

**!owner** The owner of the SignShop, can be used in the “commands:” section to enact commands on the owner

**!world** The world of the SignShop, can be used in the “commands:” section to enact commands on worlds

**!line#** The writing on the line of the sign corresponding to the “#”, used for the “commands:” section

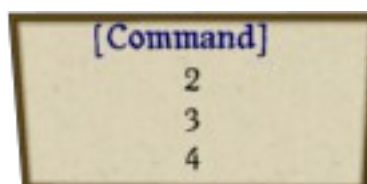
[Back to Index](#)

## **Using Signs to Run Commands**

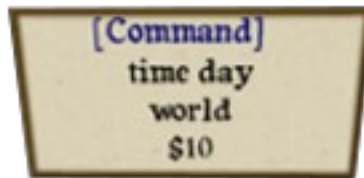
You may want to have SignShop sell something intangible from another plugin. You can use SignShop to sell a rank, a teleport to a specific place, or maybe you want to make a funny suicide booth involving lightning. As long as you have a plugin that allows you to run the commands you want in console, you can use SignShop to have users pay for the commands! There are 2 ways to do it in SignShop.

### **Using a [Command] Sign in SignShop**

This is the easiest way to run commands with SignShop is by writing them on a [Command] sign. A command sign is broken into 4 lines.



The top line (line1) has the title, "[Command]", and is there to indicate to SignShop that this sign will run commands. The remaining lines can be used to type commands. Here is an example of a completed command sign.



In the configuration you will see the following lines.

```
213 - tp !player !owner
214 Command:
215 - \!line2 !line3
216 pricemultipliers:
217
```

In this example the command being run is "time day world". "time day" is taking the place of the variable !line2, and "world" of !line3. The fourth line is the price of using the sign. That is the basic premise of the command sign. You can also use the !line# variables in other sign commands if you like, just **make sure to put a "\" before any "!" when starting a new line** like the Command example has, otherwise Java gets confused and will tell your config.yml to regenerate! If you don't have enough space to type commands on a sign, there is another option.

### Setting Up a Custom Sign

Sometimes the 3 available lines on a sign may not be enough. Setting a sign to run multiple commands, or a really long command might be an example of that. Here's how to set up a sign to run the default Bukkit command **/tp**:

- 1) Create a custom sign by making a new sign under the "signs:" section of the config.yml. We are going to call our sign **TpToOwner**.

```
9 enableSignstacking: true
10 signs:
11 TpToOwner: takePlayerMoney,playerIsOp
12 Buy: takePlayerMoney,takeShopItems,giveOwnerMoney,givePlayerItems
13 Sell: takeVariablePlayerItems.takeOwnerMoney.giveShopItems.givePlayerMoney
```

- 2) You will notice I added the Blocks "takePlayerMoney" and "playerIsOp." I did that to make it so only administrators can make the **TpToOwner** sign, and that anyone who uses the sign will be charged money. You can add whatever Blocks you like, but be careful that you don't accidentally set up signs that anyone can make if they grant infinite items, or run commands in console.

```

68      DeviceLogger: You have setup a device logger that costs !price!
69      confirm:
70      TpToOwner: Teleport to the owner of this SignShop for !price?
71      Buy: Buy !items for !price?
72      Sell: Sell !items for !price?

```

- 3) Make sure to add messages to the config so players know what they are getting!
- 4) Now go to the "commands:" section in your config and type the name of the sign you want to have run commands and underneath it you want to put the command(s) to be executed. Here's what it should look like for **TpToOwner**:

```

199      commands:
200      TpToOwner:
201      - tp !player !owner
202

```

- 5) Note how "commands:" is formatted. The sign name is **indented by 2 spaces** and the command **does not have the "/"** used to send commands in game. In game the command is `/tp Player1 Player2`, but because we want whoever used the sign to teleport to whoever owns the sign we substitute "Player1" with **!player** and "Player2" with **!owner**. And that's all there is to it! This sign will now teleport players to you for whatever price you put on the 4<sup>th</sup> line.

[Back to Index](#)

## Limiting the Number of SignShops

There are several ways to limit the number of shops a player can create. Let's take a look at a few.

The first and most obvious way is using the **MaxShopsPerPerson** setting in the config. This is a global setting that affects all players. There are other ways to limit player shops, but this one takes priority over all of them. You can allow a player to bypass this permission by giving them the permission node **Signshop.IgnoreMax**.

The second way offers more control over individual groups. For this you can use the "limits:" section in the config. You will see two entries already set under "limits:", **DefaultEx** and **MemberEx**. These examples would correspond to the player groups defined in your permissions plugin. Undefined groups here will be ignored. If you want to add your own groups, you can do that just like the examples are set up. If the amount set in "limits:" is higher than the one set by **MaxShopsPerPerson** it will be ignored. "limits:" will not override the global maximum. Again, if you need to bypass these settings for an individual user, you can give them the permission node **Signshop.IgnoreMax**.

[Back to Index](#)

## Using Price Multipliers

Price Multipliers are a way to give user groups special buffs or nerfs to specific types of SignShops. Let's look at a few examples. We have 2 user groups defined in the "pricemultipliers:" section: **VIPEx** and **BlacksmithEx**.

```

202
203 pricemultipliers:
204   VIPEx:
205     iBuy: 0.7
206     iSell: 1.2
207   BlacksmithEx:
208     Repair: 0.5

```

Let's first look at **BlacksmithEx**. We can see that "Repair" has a multiplier of 0.5, this means they only pay 50% of the price for [Repair] signs.

Above that, we can see the **VIPEx** group. You will notice that this group gets a 30% discount at [iBuy] signs, and a 20% bonus for selling to [iSell] signs. You can list as many number of signs here as you like. Be careful however, if you give discounts to certain groups for using player shops, and not server shops, the owner of the shop only gets what the discounted group paid.

[Back to Index](#)

## Permissions

There are a few basic SignShop permission nodes. The "\*" symbol indicates either the name of a sign or a type of block (depending on the permission node). The permission will then affect that sign or type of block. This also works for custom signs. Here are the basic nodes:

### Signshop.DenyUse.\*

This permission denies usage of signs.

**Example:** You may wish to deny a group's ability to sell to an infinite shop (i.e. an [iSell] sign), in which case you would give them: **Signshop.DenyUse.iSell**

### Signshop.DenyLink.\*

This permission denies the linking of shops to certain in-game blocks.

**Example:** You may wish to deny a group's ability to link their shops to a furnace to prevent automatic smelting, in which case you would give them: **Signshop.DenyLink.Furnace**

**List of linkable items:** Chest, Sign, Lever, Dispenser, Furnace, Brewingstand, Enchantmenttable, Slab

### Signshop.Signs.\*

This permission allows players to create signs.

**Example:** You might want to disallow a group from creating the sign to toggle redstone levers, you can remove that ability by negating it with the following permission node (assuming your permission plugin allows negating permissions): **-Signshop.Signs.Toggle**

### Signshop.CopyPaste

This permission allows players to click on signs with black dye to copy information onto an already active

SignShop.

**Example:** If you want to update the price of an item, create a new sign and put the new price on the bottom line. Leave the other 3 lines blank, then click with **black dye**. You can modify the description, the price, and the type of sign this way. Blank lines are ignored. You cannot, however, change a Device sign to a Buy sign, as the operations are incompatible with one another. You can also allow moderators and admins to edit other players' signs with **Signshop.CopyPaste.Others**

### Signshop.Permitt

If the **AllowPermits** setting in the global options is set to true, players must have this node in order for their shops to work.

**Example:** You can use SignShop and a permission plugin to sell permits allowing users to be merchants. Without a permit, the shop will be disabled and they will need to buy another in order for their shops to continue functioning.

### Signshop.ChangeOwner

This permission allows a player to click on another player with redstone to change the owner of a SignShop.

**Example:** If you would like to set up a player account as a bank, or transfer a store to another player, you would punch them with redstone, then punch the sign you would like to modify. If you do not own the sign that is being modified, you will also need the permission **Signshop.ChangeOwner.Others** to do so.

### Signshop.IgnoreMax

This permission bypasses any defined maximum shop settings in the [config](#).

**Example:** You can make it so normal players can only create something like 10 signs, while donators can create infinite, by giving them **Signshop.IgnoreMax**

### Signshop.IgnoreRepair

This permission bypasses AllowEnchantedRepair setting in the [config](#).

**Example:** You can make it so "Blacksmiths" can repair enchanted items with **Signshop.IgnoreRepair**

### Signshop.Admin.\*

This permission allows players to create administrative signs (such as shops with infinite items for global shops).

**Example:** You can use this permission to grant the ability to make signs with the `playerIsOp` tag (defined in the [config](#)). **Signshop.Admin.Heal** allows a player to create a healing station.

[Back to Index](#)



## Donations and Sponsors

We will never charge you anything to use SignShop, we truly believe in free and open software. We do it solely for the Minecraft community, and server owners like you. If your server couldn't do without SignShop, why not thank us by [sending us a few bucks](#) for beer and/or coffee?



If you are unable to donate, we understand. But please also feel free to visit our sponsors, [Beastnode](#). They are the best server hosts I've ever had the opportunity of dealing with. If you are looking for a host, or already have a one, you might be paying too much for a service that isn't nearly as good. Beastnode runs their Minecraft servers off SSDs and offer 99.9% uptime. I've been with them since the beginning, and I'm glad I'll never have to deal with bad service again. Plus, if you visit them using our affiliate link, you can help support the development of SignShop! You get an awesome server, we make an awesome plugin. Everybody wins!



[Back to Index](#)

