SignShop 2

A Powerful and Easy to Use Shop Plugin

Quick Reference

for version 2.10

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

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{acceptdamaged}; Allows the selling of partial and full amounts if the **AllowVariableAmounts** setting is true, optional parameter to allow selling of used tools for partial money.

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

takePlayerInventory{clearArmor}: Takes all of the items in the user's inventory, optional parameter to clear player armor as well

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

giveShopItems{allowemptychest}: Gives the items designated during the sign set up to the shop chest, optional parameter to create shops that take any item (useful for takeItemInHand)

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

takeTownMoney: Takes the amount of currency on the 4th line of the sign from the Towny town, to which the owner of the shop belongs

giveTownMoney: Gives the amount of currency on the 4th line of the sign to the Towny town, to which the owner of the shop belongs

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 by 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{variablecost}: Sets the item in the user's hand to full durability, optional parameter to charge partial money for partially damaged items

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 3rd 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 3rd line of the sign, optional parameter to take raw XP instead of XP levels

playJukebox: Looks for records in a chest and plays them in order

oneTime: Makes it so a player may only use that type of sign once

resetOneTime{NAME}: Resets the shop type listed as the optional parameter, replace "NAME" with the name of a sign that should be affected.

promotePlayer: Promotes players to the rank listed on the 2nd line of the sign. (Note: most permission plugins consider inherited groups to be part of the users group, so SignShop won't allow you to promote players to a lower rank.)

cooldown{SECONDS}: Sets a cooldown (replace "SECONDS" with the number of seconds) for individual signs, so you can make a sign that players can only use once every hour (3600), or day (86400), or any other number of seconds.

runCommand{asUser} OR **{asOriginalUser}**: Runs commands in the "commands:" section for this sign, optional parameters to make it as if the player using the sign typed the command (Note: the {asUser} optional parameter temporarily gives the player a "*" permission while SignShop runs the command, then removes the permission. Support for the "*" permission is required by your permission plugin for the {asUser} feature. {asOriginalUser} runs the command without any additional permissions, and is the same as if they typed it in chat.)

runTimedCommand{SECONDS}: Runs the commands in the "timedCommands:" section for this sign after a certain number of seconds (replace "SECONDS" with the number of seconds). These commands can ONLY be run in the console.

Special Blocks

ShareSign: A Block only used for [Share] signs

RestrictedSign: A Block only used for [Restricted] signs

BankSign: A Block only used for [Bank] signs

HotelSign: A Block only used for [Hotel] signs (must have SignShopHotel installed)

givePlayerGuardians: A Block only used for [Guardian] type signs (must have SignShopGuardian installed)

OnExpiration{SOMEOTHERBLOCK}: This block runs “SOMEOTHERBLOCK” when a hotel expires. A good use of this would be to replace “SOMEOTHERBLOCK” with runCommand and do some additional things when the hotel expires (must have **SignShopHotel** installed)

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 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 4th line of the SignShop, after any modifications or multipliers

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

!xp The amount of experience levels listed on the 3rd 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

!x, !y, !z Used to list coordinates of a shop, mostly for error messages, but can be used elsewhere as well

!profits The list of people on lines 2 and 3 of a share sign

!profitshops The locations of signs linked to a [Share] sign

!restrictedshops Used to display the shops a [Restricted] sign is restricting

!blacklisted_item Used to display an item on the blacklist

!param Used to display the optional parameter from a Block in a message

!cooldownleft Used to display the amount of time remaining before a player can use a shop (in ms)

!cooldownminutesleft Used to display the amount of time remaining before a player can use a shop (in minutes)

!cooldownhoursleft Used to display the amount of time remaining before a player can use a shop (in hours)

!cooldowndaysleft Used to display the amount of time remaining before a player can use a shop (in days)

!cooldownyearsleft Used to display the amount of time remaining before a player can use a shop (in years)

!destroymaterial Used to display the item set in DestroyMaterial

!signtype The name of the sign as set in the config.yml

!shopinventory The items inside the chests linked to a shop

!bank The bank listed on the second line of a bank sign

!times The amount of times a message was suppressed

!permgroup The permission group a player is in at the beginning of the transaction

!promoteto The permission group a player will be added to, listed on the 2nd line of the sign

SignShopHotel (These variables are only used for the SignShopHotel plugin)

!hotel The name of the hotel, as shown on the second line of the sign

!roomnr The room of the hotel that will be rented

!renttime The length of time for which a room will be rented

SignShopGuardian (These variables are only used for the SignShopGuardian plugin)

!guardians The amount of Guardians listed on the second line of the sign

!currentguardians The amount of guardians you currently have

Back to Index

Using Signs to Run Commands

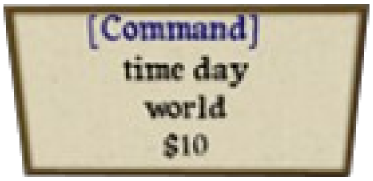
You may want to have SignShop sell something intangible from another plugin. You can use SignShop to sell a rank, a teleport to a specific place, or maybe you want to make a funny suicide booth involving lightning. As long as you have a plugin that allows you to run the commands you want in console,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

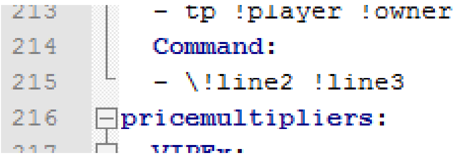
This is the easiest way to run commands with SignShop is by writing them on a [Command] sign. A command sign is broken into 4 lines.

[Command]
2
3
4

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.



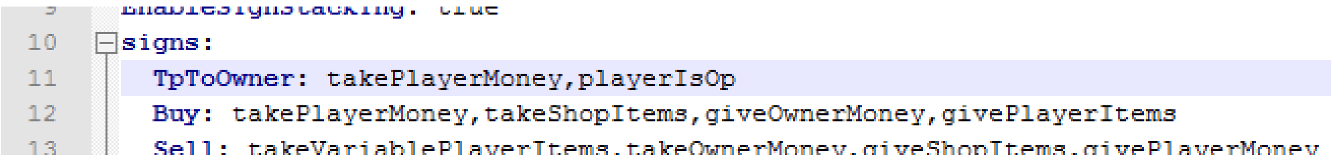
```
  - tp !player !owner
  Command:
  - \!line2 !line3
pricemultipliers:
  VIPEx:
```

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

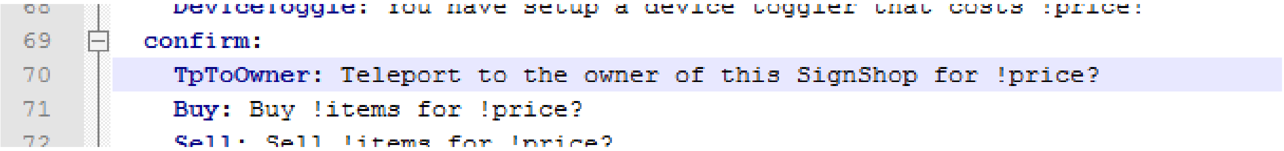
Sometimes the 3 available lines on a sign may not be enough. Setting a sign to run multiple commands, or a really long command might be an example of that. Here's how to set up a sign to run the default Bukkit command **/tp**:

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**.



```
EnableSignStacking: true
signs:
  TpToOwner: takePlayerMoney,playerIsOp
  Buy: takePlayerMoney,takeShopItems,giveOwnerMoney,givePlayerItems
  Sell: takeVariablePlayerItems,takeOwnerMoney,giveShopItems,givePlayerMoney
```

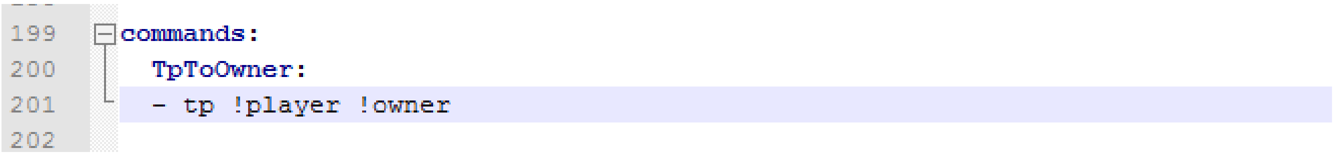
2) You will notice I added the Blocks "takePlayerMoney" and "playerIsOp." I did that to make it so only administrators can make the **TpToOwner** sign, and that anyone who uses the sign will be charged money. You can add whatever Blocks you like, but be careful that you don't accidentally set up signs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



```
    DeviceToggle: You have setup a device toggler that costs !price!
  confirm:
    TpToOwner: Teleport to the owner of this SignShop for !price?
    Buy: Buy !items for !price?
    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**:



```
commands:
  TpToOwner:
  - tp !player !owner
```

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 4th 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.

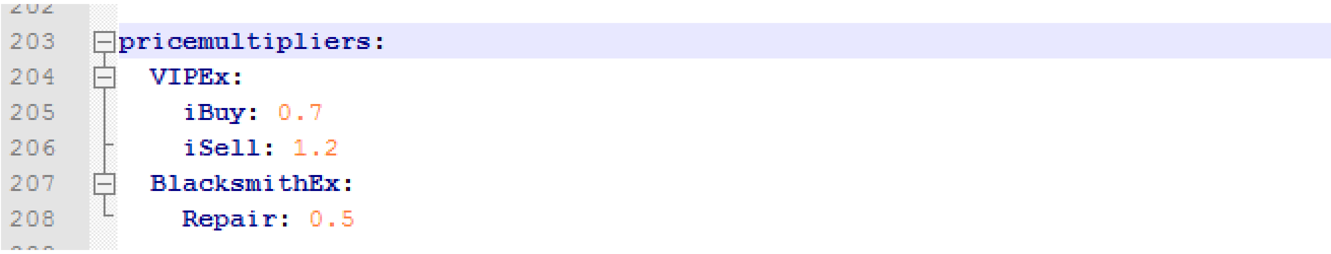
The first and most obvious way is using the **MaxShopsPerPerson** setting in the config. This is a global setting that affects all players. There are other ways to limit player shops, but this one takes priority over all of them. You can allow a player to bypass this permission by giving them the permission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



```
pricemultipliers:
  VIPEx:
    iBuy: 0.7
    iSell: 1.2
  BlacksmithEx:
    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. All permission nodes are case insensitive. 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.Permit

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. You can also do* **Signshop.Permit.Stone** *to only allow shops containing the material "Stone". Material names can be found at* http://jd.bukkit.org/rb/apidocs/org/bukkit/Material.html *(Make sure to replace any spaces with an underscore, i.e. "BAKED POTATO" becomes "BAKED_POTATO")*

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.BypassShopPlots.*

This permission bypasses EnableShopPlotSupport setting in the config.

Example: *You can make it so VIPs are allowed to create shops is regions with the "allow-shop" flag set to "deny" in Worldguard using* **Signshop.BypassShopPlots.Worldguard**

Supported plugins: *Worldguard's "allow-shop" flag, Towny's shop plots*

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.*

Signshop.SuperAdmin

This permission makes it so players are seen by SignShop as OPs.

Example*: You can use this permission to grant the ability to bypass the blacklist, break other users' shops, use /signshop reload, etc.*

Back to Index

Donations

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?

Back to Index