Build Commands

The Ultimate Worldedit Companion Tool

General Info

Welcome everyone to Build Commands, which is my latest and greatest addition to my series of map making tools. It is a single worldedit craftscript that offers over 30 tools, and brushes not available in normal worldedit. Most of these tools are things I wished I had when I was working on my own maps, some are from my older scripts, but a lot are brand new ideas I have just came up with recently.

This script, and everything else you see here, has been created 100% by me alone. over the past few months or so. If you enjoyed at all, please consider showing me some support, either by giving me a diamond on this mod, or liking the video, it really does help a lot, and encourages me to continue making new stuff like this.

Majority of all commands work in both single, and multiplayer versions!
Save, load, and perfectly place all new shape objects with a single click!
Easy kit creation that allows loading up to 18 shapes with 1 command!
Support for loading, and placing TerrainContol mod .bo2 objects!
10 new randomly generated tree types!
Paint biomes ingame with a brand new biome brush!
Easily create rivers and ravines with the advanced overlay brush!
Fill and erode terrain away with my terraforming toolkit brushes!
Mirror and flip your selection with greater ease and precision!
Create a customizable flatland anywhere using the new flatten brush!
Create custom 3d lines, laser beams, and massive spikes!
Easily Place large patches of random ground foliage, or hanging vines!
Rotate, and revolve your selection to any angle or increment!
Save surface maps of your current area to an external png image file!
Play the classic, windows puzzle game, Minesweeper, completely ingame!

Installation Guide

Singleplayer

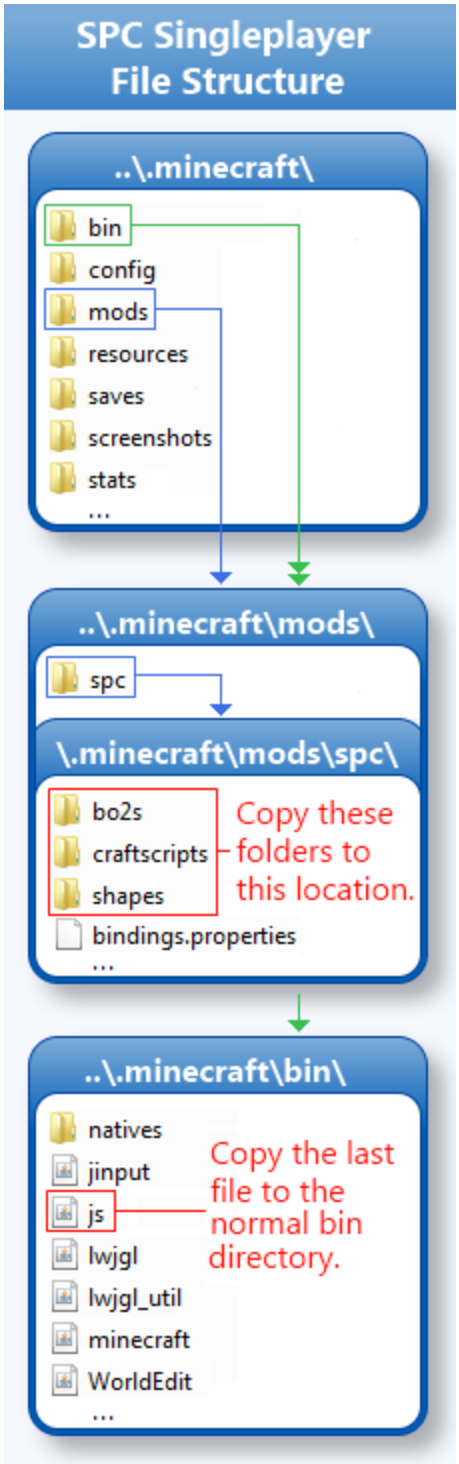
1) Install the SinglePlayerCommands mod, which also comes with WorldEdit.
2) Open your main *"%appdata%/.minecraft"* folder up. (The one with bin and saves folder)
3) From the .minecraft folder navigate to the mods folder then to the spc folder.
4) Copy the unzipped folders "*craftscripts","shapes" & "bo2s"* into it, merging if necessary.
5) Find the worldedit.properties file in the same folder and open with any text editor.
6) Go to the line that reads *"scripting-timeout="* and change the value to 30000.
7) Save the file and close all folders, open Minecraft and test it out with the command *"/cs build"*.

Multiplayer *You must own/have access to server files to install scripts.

1) These scripts all require the worldedit bukkit plugin, make sure to have it installed before attempting.
2) Once installed, locate the *"plugin/worldedit/"* directory where worldedit installed and open the folder.
3) Now just copy the downloaded folders "*craftscripts","shapes" & "bo2s"* into it, merging if necessary.
4) Make sure to increase script timeout value in the *"plugins/worldedit/config.yml"* file to at least 30000.

File Structure Help

See next page.


SPC Singleplayer File Structure
..\.minecraft\
bin
config
mods
resources
saves
screenshots
stats
...
..\.minecraft\mods\
spc
\.minecraft\mods\spc\
bo2s
craftscripts
shapes
bindings.properties
...
Copy these folders to this location.
..\.minecraft\bin\
natives
jinput
js
lwjgl
lwjgl_util
minecraft
WorldEdit
...
Copy the last file to the normal bin directory.

Bukkit Multiplayer File Structure
..\server\
plugins
world
world_nether
world_the_end
banned-ips
banned-players
bukkit.yml
...
..\server\plugins\
WorldEdit
js
worldedit
Copy this from the zip download.
..\plugins\worldedit\
bo2s
craftscripts
shapes
config.yml
worldedit
Copy these folders here.
That's it!

Command Structure & General Info

-Build commands uses an all new "flag" based command style that allows more options and flexibility when setting arguments. This means instead of trying to remember the order arguments belong in, you can now just set a flag specifying which it is.

-Almost all tool arguments are optional, and can be used on an as needed basis.

-Using the tilde("~") sign plus a value will restrict the range of ANY tool, (Eg. ~50, ~25, etc)

-Also, in most tools, clicking blocks above you will invert the created object, building it downward.

Future Plans

My plan for this script is to keep adding new brushes, tools, and possibly even more games as time passes. I already have around 4 more in development already, that weren't quite finished in time for the release. Also, I have recently discovered even more possibilities for multiplayer that uses bukkit only code, that will allow me to do many things that were previously not possible. So, keep your eye out for lots of new content for this in the future!

Array - Stacks the selection in 3 directions.

<u>Command:</u> /cs build array [totalA] [totalB] [totalC]

<u>Description:</u> Allows stacking a selection in up to 3 different directions at once. Great for large repeated structures. This command will construct a special multi-stage brush that will guide you through setting up offset points that control exactly where the selections copies will be placed.

<u>Arguments:</u> The arguments control the total number of copies in each direction. The number of stacked directions will be limited by the number of entered totals.

***a »* totalA - Total number of stacked copies in the first direction.**
***b »* totalB - Total number of stacked copies in the second direction.**
***c »* totalC - Total number of stacked copies in the third direction.**

<u>Sample Use:</u>
/cs build array a5 b10 c13

/cs build array a21 b6

Biome - Brush that paints a custom biome (multiplayer only)

Command: ***/cs build biome [biome] [size]***

Description: **Creates a brush that will allow you to paint a custom biome in your world. The biome changes won't become effective until a chunk reload (leave area or exit world). Snow is used to show where the biome is being changed, and can be easily cleaned up afterward with the worldedit /thaw command.**

Arguments:
biome **- The exact name of the biome to set, use "list" to see all the available types. (Required)**
 # » size **- The size of the brush to use.**

Sample Use:
/cs buid biome swampland #10

Box - Custom hollow, or filled rectangle box brush.

Command: ***/cs build box [xSize] [ySize] [zSize] [hollow] [angled] [insideBlock]***

Description: **Creates a brush that places a custom sized rectangle box that may either be solid, hollow, or filled in with a separate block.**

Arguments:
x » **xSize - Size of the box in the x direction.**
y » **ySize - Size of the box in the y direction.**
z » **zSize - Size of the box in the z direction.**
b » **block - Outside block material.**
h » **hollow - Number of blocks thick the outer box "shell" should be.**
a » **angled - Sets if the box should be angled to align with the player direction (flag only)**
i » **insideBlock - The id of the block that should fill the inside.**

Sample Use:
/cs build box x8 y12 z14 b1
/cs build box x13 y9 z21 b20 h2 a i5

Clear - Clears away all natural blocks from the terrain.

Command: ***/cs build clear [size]***

Description: **Creates a brush that will clear away all nature blocks (flowers, grass, etc)**

from the terrain. Great for prepping an area for future terraforming.

Arguments:
***s »* size - Brush size of the area to be cleared.**

Sample Use:
/cs build clear s25

Commands - Detailed listing of all available commands.

Command: ***/cs build commands***

Description: Prints out a detailed listing of all the commands currently available and all their associated arguments.

Ellipse - Custom hollow, or filled ellipse shape brush.

Command: ***/cs build ellipse [xSize] [ySize] [zSize] [hollow] [angled] [insideBlock]***

Description: Creates a brush that places a custom ellipse shape that may either be solid, hollow, or filled in with a separate block.

Arguments:
***x »* xSize - Size of the ellipse in the x direction.**
***y »* ySize - Size of the ellipse in the y direction.**
***z »* zSize - Size of the ellipse in the z direction.**
***b »* block - Outside block material.**
***h »* hollow - Number of blocks thick the outer ellipse "shell" should be.**
***a »* angled - Sets if the ellipse should be angled to align with the player direction (flag only)**
***i »* insideBlock - The id of the block that should fill the inside.**

Sample Use:
/cs build ellipse x8 y12 z14 b1
/cs build ellipse x13 y9 z21 b20 h2 a i5

Erode - Brush that erodes and smooths natural terrain.

Command: ***/cs build erode [size] [maxFaces] [iterations]***

Description: This command creates a brush that will erode away any natural terrain block based on the number of open faces set in the arguments.

Arguments:

s » size **- Size of the brush tool.**
f » maxFaxes **- Number of open block faces needed to trigger erosion (1-5).**
i » iterations **- Strength of the brush, or the number of times to repeat the erosion.**

Sample Use:
/cs build erode s6 f3
/cs build erode s8 f2 i2

Fill - Brush that fills and smooths natural terrain.

Command: ***/cs build fill [size] [maxFaces] [iterations]***

Description: **This command creates a brush that will fill in any natural terrain blocks based on the number of different blocks there are around it.**

Arguments:
s » size **- Size of the brush tool.**
f » maxFaxes **- Number of open block faces needed to trigger fill (1-5).**
i » iterations **- Strength of the brush, or the number of times to repeat the fill.**

Sample Use:
/cs build fill s6 f3
/cs build fill s8 f2 i2

Flatten - Level and clear all terrain to a custom height.

Command: **/cs build flat [size] [depth] [surfaceBlock]**

Description: **This creates a brush that will allow you to flatten all terrain, and anything else to a custom depth level, and surface block.**

Arguments:
s » size - Size of the brush tool.
d » depth - Exact y coordinate level that everything should be leveled to.
b » surfaceBlock - Block that will be used for the leveled surface.

Sample Use:
/cs build flat s5 d62
/cs build flat s20 d120 b5

Flip - Flips the current selection around the player.

Command: **/cs build flip [shift] [delete]**

Description: This creates a brush that will flip the selection around the clicked point using the players current direction as the axis to flip over.

Arguments:
s » shift - Shifts the selection to follow the flip position. (flag only)
d » delete - Deletes the original selection contents after flipping (flag only)

Sample Use:
/cs build flip s
/cs build flip s d

Fragment - Creates a fragmented sphere shape.

Command: /cs build frag [size] [block] [density] [hollow]

Description: This brush was created to be a sphere brush with a random amount of noise around the outside edge to prevent circular patterns from appearing while doing certain terraforming building. It is most useful when also using with an erode/fill brush to make terrain look more natural.

Arguments:
s » size - Size of the brush tool.
b » block - Block material to use.
d » density - Outside sphere density - Whole %(0-100 | 0=Fragmented | 100=Solid)
h » hollow - Hollow shell thickness - Whole %(0-100 | 0=Solid | 100=Hollow)

Sample Use:
/cs build frag s6 b4
/cs build frag s10 b5 d80 h50

Grass - Creates a random patch of grass and flowers (super bonemeal!)

Command: */cs build grass [size] [density]*

Description: This creates a brush that will cover all grass and dirt blocks with a random assortment of long grass, flowers, and occasionally a pumpkin or melon.

Arguments:
s » size - Size of the brush tool.
d » density - Total density of the grass patch (0.01 - 1)

Sample Use:

/cs build grass s10
/cs build grass s25 d.5

Help - General, or command specific info.
Command: ***/cs build help [command]***

Description: The help command will give detailed info and arguments for any available command. Using it alone will display the normal build commands help message. Additionally, the ? sign may also be used at the end of any command string to give info about that tool.

Arguments:
command - Name of the command to look up.

Sample Use: All of these samples will produce the same help message on the biome tool.
/cs build help biome
/cs build ? biome
/cs build biome swampland #10 ?

Kill - An experimental brush that kills entities. (multiplayer only)
Warning! This brush is very unstable and may crash you, or the sever, use at your own risk!

Command: /cs build kill [entityType] [size]

Description: This creates a brush that will kill specific, or all entities around the clicked area.

Arguments:
entityType - The ID of the entity type you want to kill, leave blank for all.
s » size - Size of the brush tool.

Sample Use:
/cs build kill cow
/cs build kill pig s10

Kit - Loads, and binds a list of custom shapes from a kit file.
Command: /cs build kit [fileName] [angleLock] [excludeID] [select]

Description: This command will automatically load, and bind, many different shapes to items all at once. This is done by loading .kit files, which can be easily generated by dragging and dropping shape (.shp or .bo2) files onto an external batch script located in the shapes folder.

Arguments:
fileName - Name of the kit file to load.
< » angleLock - Locks the shape to a certain angle, use 360 to use exact player direction.
! » excludeID - Block to skip, or exclude when placing down the shape.
$ » select - Updates the selection to the bounding box of the newly placed shape. (flag only)

Sample Use: *All kits except for village are examples only, and are not included.
/cs build kit village
/cs build kit houses <90
/cs build kit ruins <360 !2 $

Laser - Shoots a custom beam of blocks from your fingertips!

Command: /cs build laser [size] [depth] [topBlock] [subBlock]

Description: This brush will allow you to lay down a beam of destruction(air), or any other blocks to wherever you are currently pointing. Several custom parameters allow you to specify the overall beam size, along with surface penetration depth and materials.

Arguments:
s » size - Size of the beam.
d » depth - Depth the beam will penetrate past the clicked surface.
a » topBlock - Material for the block above the surface.
b » subBlock - Material for the block below the surface.

Sample Use:
/cs build laser s7 a5
/cs build laser s5 d30 a18 b17

Line - Draws a custom 3d line in single, continuous, or fixed origin mode.

Command: /cs build line [mode] [size] [block] [extendCnt]

Description: This creates a brush that will allow you to build custom lines segments in 3 different modes: single, continuous, and fixed origin. Holding down the right mouse

button while in continuous mode with a brush mask of !0, allows "drawing" over any surface block.

Arguments:
m » mode - Line mode to use (0 = single line; 1 = continuous; 2 = fixed origin)
s » size - Overall thickness of the line.
b » block - Block material to use.
e » extendCnt - Amount to lengthen, or shorten the line segments.

Sample Use:
/cs build line m1
/cs build line m0 s5 e10
/cs build line m2 s10 b5 e-5

List - List Commands - Short
Command: **/cs build list**

Description: **This will print out a short reference list of all the available commands.**

Map - Saves a map of the area around you to an image file.
Command: **/cs build map [fileName] [size] [heightMap]**

Description: **This tool will attempt to map out, and save a surface image of the blocks around the player to a specified .png image file.**

Notes: **Size seems to be much more limited on multiplayer, at around 256, while SPC has been tested to 1000+. Also, many "un-natural" blocks have not had there color values setup and will only appear as black. (if anyone wants to add more feel free to send me a new list and I will add them in.)**

Arguments:
fileName - The name of the image file to be saved. (required)
s » size - Size of area to be mapped, centered around the players position.
h » heightMap - Adds color shading based on the height of the terrain. (flag only)

Sample Use:
/cs build map myMapName s256
/cs build map otherName s512 h

Mine - Play a game of Minesweeper, Minecraft style!

Command: /cs build mine [xSize] [ySize] [mines] [cheat] [wool] [difficulty] [hardcore]

Description: This tool allows you to play the classic Minesweeper game in Minecraft! The first click after binding the brush to an item will create the minefield with whatever settings were entered. After that the sweeper tool becomes active which allows you to play the game. Right clickin g on the grid surface blocks will set and clear flags. Right clicking on any other block will print the number of mines left and the current time. Left clicking on the surface blocks opens them up, while left clicking on any ore block will open the 8 around it only if the right number of flags is set within the area. A basic block/mine legend is also created at one end of the field showing the set order.

Arguments:
x » xSize -Width of the minefield.
y » ySize - Length of the minefield.
m » mines - Number of mines to place.
c » cheat - Cheat Mode. (flag only)
w » wool - Alternate wool block theme (flag only)
b, i, e » difficulty - Preset settings for beginner, intermediate and expert difficulty. (flag only)
h » hardcore - Turns on hardcore mode, which will ignite all the mines when you lose. (flag only)

Sample Use:
/cs build mine b
/cs build mine x24 y16 m30
/cs build mine c w e h

Mirror - Mirrors your current selection around a selected point.
Command: /cs build mirror [shift] [delete]

Description: This creates a brush that will mirror the selection over the clicked point using the players current direction as the axis to mirror over.

Arguments:
s » shift - Shifts the selection to follow the mirrored position. (flag only)
d » delete - Deletes the original selection contents after mirroring (flag only)

Sample Use:
/cs build mirror s
/cs build mirror s d

Ore - Generates new veins of custom ore, using a brush, or selection.

Command: /cs build ore [size] [overBlock] [density] [region]

Description: This tool will generate new custom ore veins over the specified block, using a brush tool or the current selection region. Ore specific settings can currently only be changed from within the script itself.

Arguments:
s » size - Size of the brush tool.
b » overBlock - Block the ore should replace, or paste over, stone by default.
d » density - General density of the ore veins, by whole percent, 100 is baseline.
r » region - Specifies to use the current selection instead of the brush area (flag only)

Sample Use:
/cs build ore s15 b1
/cs build ore s10 b1 d150
/cs build ore b1 r

Overlay - Advanced overlay tool that can cover 3 layers all to custom depths, many uses!
Command: /cs build overlay [size] [topBlock,depth] [mid,depth] [end,depth] [all]

Description: This is an advanced overlay brush that allows you to set up to three different materials and layer depths at one time. Using air or water as arguments allows easy creation of rivers and ravines!

Arguments:
s » size - Size of the brush tool.
t » topBlock - Top layer block material and depth (comma separated, no spaces)
m » mid - Middle layer block material and depth
e » end - End layer block material and depth
a » all - Override to cover all blocks, not just natural ones. (flag only)

Sample Use:
/cs build overlay s10
/cs build overlay s10 t0,45 m1,2 *Ravine Example
/cs build overlay s6 t0,1 m9,3 e1,1 *River Example

Paint - Attempts to paint shape objects in rapidfire succession.
Command: /cs build paint [fileName] [angleLock] [excludeID]

Description: This tool works exactly like the shape tool but tries to place down the shapes at a faster rate than normal (your results may vary).

Arguments:
fileName - Name of the kit file to load.
< » angleLock - Locks the shape to a certain angle, use 360 to use exact player direction.
! » excludeID - Block to skip, or exclude when placing down the shape.

Sample Use: *Shape file names used are for example only.
/cs build paint ball !0
/cs build paint house <90 !2

Pattern - Replaces all blocks with a custom, predefined set.

Command: /cs build pattern [blockSet] [size]

Description: This brush will replace all blocks within the clicked area to one of a pre-defined internal block set. As of right now the only pattern set available is the "ruin" set, more to come later.

Arguments:
blockSet - Name of the pattern set to use. (currently only "ruin" available)
s » size - Size of the brush tool.

Sample Use:
/cs build pattern ruin
/cs build pattern ruin s15

Platform - Creates a custom platform, or path under your feet.

Command: /cs build platform [size] [block]

Description: This brush will create a platform under your feet wherever you're at. It can also be used to easily create paths on the ground by holding the right button and walking.

Arguments:
s » size - Size of the brush tool.
b » block - Block material of the platform.

Sample Use:
/cs build platform s7
/cs build platform s10 b13

Revolve - Revolves a 2D slice into a 3D solid.
Command: /cs build revolve [count] [useBlock]

Description: This creates a brush that will revolve a 2D selection slice to a full 3D solid object. The clicked position will be used as the revolve center point.

Arguments:
c » count - Number of slices to generate, not setting will default to a full revolve.
b » useBlock - Only revolve a certain block type.

Sample Use:
/cs build revolve
/cs build revolve c32 b5

Rotate - Rotates a 3D selection to a set angle or # of increments.
Command: /cs build rotate [items/-angleInc] [resolution] [single]

Description: This tool is similar to the revolve brush, except this one will rotate or polar array 3D selections, instead of revolving them. You have the option of setting the exact angle increment you want by using a negative number for the increment. Or you can set the exact number of items by using a positive number.

Arguments:
i » items/-angleInc - Increment value, use positive for exact item count, or negative for exact angle.
r » resolution - How well the script fills in missing blocks, defaults to 4 (increase slowly!)
s » single - Limits the rotation to a single copy (flag only).

Sample Use:
/cs build rotate i32
/cs build rotate i-45 r6
/cs build rotate i-155 s

Save - Save the current selection to shape file.
Command: /cs build save [fileName] [excludeBlock] *Selection required

Description: This tool allows you to save the selection to an all new shape file, to be used later on with the shape, paint or kit commands. Orientation and offset(click) point are saved along with block set to allow placing the object the exact same way every time. Shape files all end with a ".shp" extension and will be saved in the "shapes" folder.

Arguments:
fileName - Name of the shape file to save. (no extension)
! » excludeBock - Optional block to exclude when saving shape file.

Sample Use:
/cs build save myFileName
/cs build save otherName !0

Shape - Load a shape object from the selection, or shape file.
Command: /cs build shape [fileName] [angleLock] [excludeBlock] [select]

Description: This tool will allow you to place down a shape object from a saved shape file, or the current selection. Multiple arguments allow for even greater control and options when placing the object.

Arguments:
fileName - Name of the shape file to load (.shp or .bo2), use dash("-") to load from selection.
< » angleLock - Locks the shape to a certain angle, use 360 to use exact player direction.
! » excludeBlock - Block to skip, or exclude when placing down the shape.
$ » select - Updates the selection to the bounding box of the newly placed shape. (flag only)

Sample Use: *All shapes except for "lothouse" are examples only, and are not included.
/cs build shape -
/cs build shape - <90 !0 $
/cs build shape lothouse
/cs build shape bigTree.bo2 <360
/cs build shape myHouse <0 !2 $

Spawner - A brush that creates entity mob spawners. (multiplayer only)
Command: /cs build spawner [spawnerType]

Description: This brush will create a custom mob spawner wherever you click.

Arguments:
spawnerType - Name of the entity to spawn (use list for available entity types).

Sample Use:
/cs build spawner wolf

/cs build spawner villagergolem

Spike - Creates custom, player aligned spikes, with a single click.
Command: /cs build spike [baseSize] [block] [minLength,maxChg]

Description: This tool will allow you to place down many different custom size spikes just by clicking around you. All spikes are orientated to be perfectly inline with the players current direction, allowing greater variety when placing. Min spike length, and max amount of variation can also be set to increase the randomness between spikes.

Arguments:
s » baseSize - Spike base diameter size.
b » block - Block material of the spike.
l » min/max - Minimum spike length, and max amount of variation (comma separated, no spaces)

Sample Use:
/cs build spike s5
/cs build spike s7 b1
/cs build spike s10 b3 l85,12

Spiral - Creates a custom spiral object.
Command: /cs build spiral [radius/-growth] [stretch] [count] [flip] [double]

Description: This tool will

Arguments:
r » radius/-growth - Radius for normal spiral, or radius growth for conical spiral (negative number).
s » stretch - Spiral stretch factor (1= long, stretched - 10+ = short, flatter)
c » count - Number of full spiral segments to generate.
f » flip - Flips the spiral on its side (flag only).
d » double - Specifies to create a double spiral segment (flag only).

Sample Use:
/cs build ore s15 b1

Stickpatch - Creates a random patch of blocks to random custom heights.
Command: /cs build stickpatch [size] [block] [minLength,maxChg] [density]

Description: This brush was designed to place down a random assortment of custom

'stick' patterns spread across the clicked surface area (Eg. random cactus patches).

Arguments:
s » size - Size of the brush tool.
b » block - Block material to use.
l » min,max - Minimum length, and max variation in 'stick' size (comma separated, no spaces)
d » density - Density of the amount of sticks placed (0 - 1)

Sample Use:
/cs build stickpatch s11
/cs build stickpatch s7 b81
/cs build stickpatch s16 b81 l1,3 d.02

Test - Script timeout test.
Command: /cs build test

Description: This will tell you what you current script timeout is, and whether or not it is too low.

Sample Use:
/cs build test

Tree - Creates one of the randomly generated tree types.
Command: /cs build tree [treeType] [size] [woodBlock] [leafBlock] [clump]

Description: This brush can create many randomly generated trees based on a set type, size and custom materials. There are currently 10 different tree types available to use.

Arguments:
treeType - Type of tree to generate (see list below).
s » size - Minimum size the tree should be (some random variation added automatically).
w » woodBlock - Material the wood block should be.
l » leafBlock - Material the leaf block should be.
c » clump - Override to use the 'clump' style leaves instead (flag only).

Tree Types:
bush » Creates a small ground foliage bush.
small » This is similar to standard trees, single block trunk with a leaf clump on top.
medium - A medium sized tree with branches and leaves spreading from the sides.
large » A taller, and more dense version of the medium tree, branches stay closer in.

rainforest » Similar to the large tree except branches are only at the top portion.
branched » Very large multi-branched tree with almost no scaling issues, great at any size.
spike » Similar to the branched tree, but all branches and tree trunk are spikes instead.
stick » Single placement version of the stickpatch tree.
palm » Random palm shape tree with a static leaf shape on top.
mushroom » Not a tree, but a large, and overall simplistic mushroom, with curved base stem.

Sample Use:
/cs build tree small s8
/cs build tree large s25 c
/cs build tree branched s80 w17 l18 c
/cs build tree mushroom s60 w35:5 l35:11

Vine - Smart custom vine placement brush.

Command: **/cs build vine [size] [density] [length] [block]**

Description: **This is a special brush that was designed to place a large amount of hanging vines at once, to the sides of solid blocks, at the appropriate angles. Custom parameters may be entered in to control the overall density and the max random length any vine may grow to. Any other materials may also be used in place of vines (fence, glowstone, leaves, even reeds!).**

Arguments:
s » size - Size of the brush tool.
d » density - Overall vine placement density (0-100).
l » length - Max possible length the vines should grow.
b » block - Alternate block material to use instead of vines.

Sample Use:
/cs build vine s10
/cs build vine s15 d5 l15
/cs build vine s10 d3 l9 b85

Wand - A smarter, more user friendly selection wand.

Command: **/cs build wand**

Description: **This tool works just like the normal worldedit wand tool, except for one major difference. Left clicking will set the selection to the block you clicked like normal, but right clicking will now set the other point to the player's current position. This allows you to make non-cuboid selections very easily, without having to type a single command to adjust for the original clicked positions.**

Sample Use:

/cs build wand

That's it!