Worldedit Terraforming Toolkit - by [inHaze](#)

This is a collection of new tools and brushes for worldedit that will give you access to new building methods and techniques never before possible! It includes a total of 9 brushes and tools that cover many aspects of world terraforming and general map making.

9 new tools never before available in Worldedit!
Works in both singleplayer and multiplayer!
Worldedit based point and click editing, with undo!
Designed to be easy to use, and completely customizable!
Craftscripts will not break on new game updates!

1. Erode - Custom terrain erosion and smoothing brush.
2. Fill - Face based terrain filling and growth brush.
3. Laser - New brush that fires a beam of destruction, or blocks to wherever you point.
4. Vine - Smart custom vine/hanging blocks placement brush.
5. Water - Smart water filling and random lily pad brush.
6. Frag - Random fragmented, and optional hollow sphere material/pattern brush.
7. Ellipse - Custom ellipse shaped material/pattern brush.
8. Rect - Custom rectangle shaped material/pattern brush.
9. Ore - Special randomized vein style ore generator with custom settings/blocks.

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 downloaded, unzipped folder *"craftscripts"* into it, merging with existing folder 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 erode"*.

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

1) These scripts all require the worldedit plugin, make sure to have it installed before attempting.
2) Once installed you just need to copy all the script files to your *"plugin/worldedit/craftscripts/"* directory.
3) Make sure to increase script timeout value in the *"plugins/worldedit/config.yml"* file to at least 30000.

Errors /Troubleshooting

Problem: I can't type at all.
Fix: Singleplayercommands isn't working, trying re installing from a fresh minecraft.jar.

Problem: Error Message "Failed to find an installed script engine." when using script.
Fix: Go to (http://www.mozilla.org/rhino/download.html) download the Rhino Zip file. Unzip it and copy the included js.jar to your "/.minecraft/bin" folder.

Problem: Error Message "Script does not exist .../.minecraft/craftscripts/xxxx.js" when using script.
Fix: Worldedit can't find the script in the right folder, make sure you copied it to the "/.minecraft/mods/spc" folder.

Problem: Error Message "Make a region " when using script.
Fix: Just like it says, you need to select a set of blocks using the //wand or //pos commands.

Important Note for 1.3 Singleplayer

As of right now, these scripts DO currently working in 1.3 singleplayer. The SPC mod has recently gone through a complete rewrite, and has been updated. Most of the worldedit tools/items are functioning correctly, however there is one major flaw with how the game/spc is handling large operations, whether it be a normal sphere brush, or one of these scripts. Before the game would just lag for a few seconds while processing, then everything would be fine. Now it will lag, then bring up a server tick loop error which closes the game. This is not an issue on 1.2.5 and before, it works flawlessly, but for the time being, you may have to use smaller sizes when editing/adding in 1.3.

Another important note is that the path to install the craftscripts has been changed in the latest SPC update. The path used to look like this *"%appdata%/.minecraft/craftscripts"*. It has now been changed to *"%appdata%/.minecraft/mods/spc/craftscripts"*. You must save the scripts here or the game will give you an error saying it can't find them.

0. General Command Info

Use '/cs commandname' in game to bring up the parameter list for each command.
Or use *'/cs commandname 0'* to use default settings for any brush.

All commands should be entered with only a single space separating the parameter values.
When setting brushes, you must have a non-usable item currently in your hand to bind the brush to.
Most brushes can have the size, material and mask changed at any time with /mat , /size and /mask.
All brushes can also use normal worldedit materials and patterns. (must set with /mat command)

1. Erode Brush

<u>Description</u>:

This and the fill brush were originally based on the voxelsniper erosion brush, and work almost the exact same way. The idea for this is that it either fills or erodes a block based on the number of faces you set. Its primary use is for cleaning up rough areas of terrain, but can also be used to completely reshape any current landscape.

Usage: */cs erode **size, faces, strength***

size - Brush radius size.
faces - Min # of exposed faces needed to trigger erosion. (1-6)
strength - Brush strength multiplier.

*Note - The faces parameter can also be changed at anytime using the /mat command followed by the value.

2. Fill Brush
Description:
This works in the same way as the erode brush, except it will try and fill in new blocks instead of eroding. It is most useful for enlarging, pulling or blending current terrain together. It will copy its most common neighbors block type when filling to make the new section fit in better.

Usage: */cs fill **size, faces, strength***

size - Brush radius size.
faces - Min # of exposed faces needed to trigger fill. (1-6)
strength - Brush strength multiplier.

*Note - The faces parameter can also be changed at anytime using the /mat command followed by the value.

3. Laser Beam Brush
Description:
This is a completely new and original brush that 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. Lots of fun to play with!

Usage: */cs laser **size, depth, mat***

size - Overall size or width of the beam.
depth - Depth the beam will penetrate past the clicked block.
mat - Beam material.

4. Vine Brush
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!).

Usage: */cs vine **size, density, length, mat***

size - Brush radius size.
density - Vine wall placement density - whole %. (0-100)
length - Maximum random vine length.
mat - Alternate material to use.

5. Water Flood Brush
Description:

This brush was designed to be a smart water flood/filling tool that can also be used generate random lily pads. It differs from a normal brush in 2 distinct ways, the first being that it will only fill areas that are at or over than the ylimit parameter, the second is that it will actually fill down all open air till it hits a solid bock.

Usage: */cs water **size, yLimit, lilypad, mat***

size - Brush radius size.
yLimit - Maximum water level y limit. (Important!)
lilypad - Lily pad density %. (0-100)
mat - Alternate fill material to use.

Use *'/cs water 0'* to create with default settings.
Stand on the water 'shore' when setting with defaults

6. Fragment Brush

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.

Usage: */cs frag **size, mat, density, shellSize***

size - Brush radius size.
mat - Brush material to use.
density - Outside sphere density - Whole %(0-100 | 0=Fragmented | 100=Solid)
shellSize - Hollow shell thickness - Whole %(0-100 | 0=Solid | 100=Hollow)

7. Ellipse Brush

Description:
Custom size ellipse shaped brush that can be used for any normal brush operations. Also features a hollow option.

Usage: */cs ellipse **xSize, ySize, zSize, mat, hollow***

xSize - Brush radius size in x direction.
ySize - Brush radius size in y direction.
zSize - Brush radius size in z direction.
mat - Brush material.
hollow - Brush inner fill. (Solid=1 | Hollow=0)

8. Rectangle Brush

Description:
Custom size rectangle shaped brush that can be used for any normal brush operations.

Usage: */cs rect **xSize, ySize, zSize, mat***

xSize - Brush radius size in x direction.
ySize - Brush radius size in y direction.
zSize - Brush radius size in z direction.
mat - Brush material.

9. Vein Style Ore Generator

Description:

This tool was designed to populate a completely solid stone area with a new vein style of ore distribution. This was mostly intended to be used on new terrraforming creations where the base of the land was all stone and needed to be used for survival. When used on a selection, it will generate new random ore veins that are completely customized by a special set of parameters for each type of ore, set within the script. These values may easily be modified to any other value along with adding new items to generate.

Usage: */cs ore **block, density**   *requires an active selection containing the blocks to be overwrote.

block - Block type the ore will generate in. (Defaults to stone, can use 0 to test ore gen in the air.)
density - Overall ore spawn density. (Whole %, 100 is baseline.)

Custom Ore Settings:
You will need to open the ore.js script file with any text editor to change ore distribution settings. All parameters can be found at the bottom of the script in the . It is important to keep the same formatting when altering or adding new items to the list!

Thanks to Simo_415 for his continuous efforts in making SPC the best mod ever.
Credit and thanks to sk89q and team, for making worldedit so amazing, and easy to script!
Thanks & credit to blockhead for posting this miracle brush and inspiring me to make this.
Credit to echurchhill and his awesome fab script for showing an easier way to save blocks.
Thanks to everyone at voxelsniper for their amazing tools, and additional inspiration.