

---

# Hops Documentation

*Release 0.0.1*

**Hops**

**May 14, 2017**



<b>1</b>	<b>Getting Started</b>	<b>1</b>
<b>2</b>	<b>Hard Ops</b>	<b>3</b>
2.1	Hard Ops Introduction Guide . . . . .	3
2.2	Hard Ops Instalation . . . . .	4
2.3	Hard Ops Menus . . . . .	4
2.3.1	Hard Ops hud . . . . .	4
2.3.2	Hard Ops menu . . . . .	4
2.3.3	Hard Ops Pie Menu . . . . .	5
2.3.4	Hard Ops helper . . . . .	5
2.3.5	Hard Ops material menu . . . . .	5
2.3.6	Material List / Material Helper . . . . .	5
2.3.7	How to use the Material List . . . . .	5
2.3.8	Where do you get more materials? . . . . .	6
2.3.9	How to bring up the Material Helper. . . . .	6
2.3.10	Hard Ops T Panel . . . . .	6
2.3.11	Hard Ops options . . . . .	6
2.4	Hard Ops Operators . . . . .	7
2.4.1	Sharpeners . . . . .	7
2.4.1.1	SSharpener . . . . .	7
2.4.1.2	CSharpener . . . . .	7
2.4.1.3	SStep . . . . .	8
2.4.1.4	CStep . . . . .	8
2.4.1.5	Clear SSharpener . . . . .	8
2.4.2	Cutters . . . . .	9
2.4.2.1	CSplit . . . . .	9
2.4.2.2	Re-Bool . . . . .	9
2.4.3	Edit mode operators . . . . .	9
2.5	Hard Ops Options . . . . .	9
2.6	release log . . . . .	9
<b>3</b>	<b>BoxCutter</b>	<b>11</b>
<b>4</b>	<b>Faq</b>	<b>13</b>
<b>5</b>	<b>examples</b>	<b>15</b>

<b>6</b>	<b>Hard Ops Legacy Guide Pages</b>	<b>17</b>
6.1	Hard Ops Introduction Guide	17
6.2	Welcome To the Hard Ops Documentation	17
6.3	Table Of Contents	18
6.4	Getting Started	18
6.5	Understanding The Functions	18
6.6	How To Install Hard Ops	18
6.7	Windows 10 / 7	18
6.8	Mac	19
6.9	Linux:	19
6.9.1	Add-On Tab	19
6.9.2	Additional Notes	19
6.10	Auxillary Addons	19
6.11	Enable Looptools	20
6.12	Using The Add-On	20
6.13	UI Tab	20
6.14	Drawing Tab	21
6.15	Info	21
6.16	Keymap Tab	21
6.17	Links / Help Tab	21
6.18	Add Ons Tab	22
6.19	These plugins are highly recommended	22
6.20	These plugins are not required but are recommended. They dont get checkboxes.	22
6.21	How to get started with Hard Ops	22
6.22	Example #1	23
6.23	Ssharp / Csharp Workflow	23
6.24	Example #2	23
6.25	Is there a hotkey guide?	25
6.26	Yes!	25
6.27	What is the difference between Ssharpen and Csharpen?	25
6.28	Operations Menu	25
6.29	Operations options	26
6.30	UV Preview	26
6.31	Segments	26
6.32	Bwidth	26
6.33	QArray	26
6.34	Tthick	26
6.35	Ssharpen	27
6.36	Csharpen	27
6.37	Cstep / Sstep	27
6.38	Clear Ssharps	27
6.39	Clean Mesh (E)	27
6.40	Sstatus System	28
6.41	Sstatus Types	28
6.42	Undefined	28
6.43	Csharp	28
6.44	Boolshape	29
6.44.1	Sstatus In Action	29
6.44.2	Mesh Override / Sstatus Reset	30
6.45	Status Override	30
6.46	Status Reset	30
6.47	Understanding C/Sharpen	30
6.48	How does Sharpening work?	30
6.49	SSharpen	30

6.50	Csharpen . . . . .	31
6.51	Hard Ops Boolean system . . . . .	31
6.52	Boolean Basics . . . . .	31
6.53	Booleans And Curvature . . . . .	32
6.54	Pokeball example . . . . .	32
6.55	More On Booleans And Curvature . . . . .	33
6.56	CSharpen . . . . .	34
6.57	What does Csharp do that Ssharp doesn't? . . . . .	35
6.58	Why does Csharp apply all my modifiers? . . . . .	35
6.59	F6 Menu . . . . .	35
6.60	<i>General Parameters</i> . . . . .	35
6.61	<i>Sharpening Parameters</i> . . . . .	36
6.62	Csharpen Advanced . . . . .	36
6.63	Csharp In action . . . . .	36
6.64	Cstep (Basics) . . . . .	36
6.65	What is Cstep? . . . . .	36
6.66	How does Cstep work technically? . . . . .	37
6.67	Alternate Use Of Cstep / Sstep . . . . .	37
6.68	Insert System . . . . .	38
6.69	What are the Inserts? . . . . .	38
6.70	Insert Types . . . . .	38
6.71	Red Inserts Details . . . . .	38
6.72	Inserts / Materials Sharing . . . . .	39
6.73	FAQs / SFQs . . . . .	39
6.74	How Do I Install Hard Ops / Boxcutter? . . . . .	39
6.75	See Install . . . . .	39
6.76	Why is the add-on checker not working? . . . . .	39
6.77	Why am I having issues when I boolean certain areas? . . . . .	40
6.78	What is Cstep / Sstep? . . . . .	40
6.79	Why is my mesh hidden when I Cstep? . . . . .	40
6.80	Can I get out of Cstep? . . . . .	40
6.81	Why is it called Sstep / Cstep / Ssharpen / Cssharp? . . . . .	41
6.82	Can I add my own inserts to Hard Ops? . . . . .	41
6.83	What do the different inserts mean? . . . . .	41
6.84	Help! Why doesn't this add-on show activated in the add-on checker! . . . . .	41
6.85	Material List / Material Helper . . . . .	42
6.86	How to use the Material List . . . . .	42
6.87	Where do you get more materials? . . . . .	42
6.88	Is there a way to adjust shaders as well? . . . . .	43
6.89	How to bring up the Material Helper. . . . .	43
6.90	Machin3 Drone Breakdown Shots . . . . .	43
6.91	Mirroring And Symmetry . . . . .	43
6.92	Mirror Mirror . . . . .	43
6.93	Automirror Addon . . . . .	44
6.94	Symmetrize . . . . .	45
6.95	Links . . . . .	45



# CHAPTER 1

---

## Getting Started

---





### Hard Ops Introduction Guide



This guide is intended to explain the basics and ideas behind Hard Ops and give some insight to its intentions and future plans.

Hard Ops initially began as a collaboration between myself and AR however over time, it has expanded to encompass a team from all over the world.

*The goals initially were to make a tool that would make our personal lives easier when it came to modelling. But this expanded to be an entire workflow.*

Now the goal of Hard Ops can be simplified into one line.

**Hard Surface modelling and workflow assistance. A helping hand.**



There is no magic button. It won't make a better artist. All our tool can do is enable you to be the maximum you.



HardOps also has an older wiki. [Version 8](#) was out at the time I was writing this.

I hope that this guide answers more questions than it asks and makes the experience of using HardOps as easy as possible. I also hope it can give some insight into our hard surface process as well as indicate ideas for future plans and developments.

I would have never been able to make it to where it is if not for the team. They are the driving force behind the development and without a team the work would not have been possible. The collaborative nature of the team and the willingness to invest time has been astounding and has helped this project live far longer than even I initially expected.

## Hard Ops Instalation

1. Download Hard ops
2. Unzip folder to  
global path (we suggest to use that one)  
C:\Users\user\AppData\Roaming\Blender Foundation\Blender\2.78\scripts\addons  
or local blender directory (will be visible in seperate blender wersions)  
C:\your directory\blender-2.78a-windows64\2.78\scripts\addons  
..
3. in blender settings unlock the addon  
![install](..\installation/img/install.jpg)
4. save user settings  
![install2](..\installation/img/install2.png)

## Hard Ops Menus

### Hard Ops hud

Hardops recognise selected object's state, and based on that displays available operators You can see current objects state in top left corner

Undefined - classic objects

CSHARP- applied after csharpen operator, indicates bevel workflow

CSTEP- applied after cstep operator, indicates multi bevel workflow

### Hard Ops menu

Menu is hardops basic way to call operators

it can be called by pressing 'Q'

Hotkey for menu can be changed in input settings just like other blender menus

Hops menu differs a bit from standard blender menus. Most of options are dynamic and change depending on selection, mode or objects state. here are few of the examles. look for operators for specific operators and asigned to them menus.

classic menu

no selection

2 obj selected

csharpen state

edit mode

## Hard Ops Pie Menu

Pie Menu is hardops alternative way to call operators

it can be called by pressing 'shift + Q'

Hotkey for pie menu can be changed in input settings just like other blender pie menus

Hops pie menu offers same dynamic options as standard hops menu.

classic menu

no selection

2 obj selected

cs sharpen state

edit mode

## Hard Ops helper

Hops helper is special hardops menu allowing You fast access to modifiers and materials options

You can call helper by pressing ctrl + `

helper requires 3dbuffer option to work correctly

## Hard Ops material menu

material menu allows You to see list of currently available object materials

You can open it by pressing alt + M

## Material List / Material Helper

---

### How to use the Material List

Q >> Settings >> Material List

This will display the materials present in the scene. If you have no materials created none will be shown.

In my scene I have defaults that I use for quick colorization. This is available per object in object mode and on a geometric selection via edit mode.

Here is an example of the material menu being used in edit mode.



In this example you can also see how it works in edit mode.



It's almost important to note that multi materials on the same mesh using geometric separations and bevels will look better with an even number bevel segment amount. You can see this in the above example as well.

---

### Where do you get more materials?

I get my materials from a variety of sources.

- Disney Ubershader
- Metal Pack
- Car Paint
- Ubershader
- Free PBR Material
- PBR Material
- Low Noise Pack

Shaders are beyond the scope of this plugin but these locations should be enough to get you started. These days I use materials more for color blocking and then swap them out for the intended ones after baking or downpacking.

In my default scene I keep a sphere on layer 0 with materials held that allow me to hold some quick preferences. From here I am able to delete the sphere later and only the used materials are kept.

Usually my goal is just a quick presentation render. Similar to this.

![image](/img/basics1/bas\_11.png)

---

### How to bring up the Material Helper.

\*\* Ctrl + ~ \*\* brings up the modifier helper. The 2nd tab is the material helper.

![/img/settings/mm3.gif]

Currently we still are developing this so there may be some issues however it does exist as an option to experiment with in the meantime. The modifier helper is quite useful for managing modifiers and we hope the material helper is just as useful.

---

![/img/settings/mm4.gif]

### Hard Ops T Panel

T panel is standard way to call operators in blender You can find all Hops operators under HardOps tab

### Hard Ops options

Hardops options like color display can be accessed by addon preferences panel

## Hard Ops Operators

### Sharpeners

#### SSharpen

Soft Sharpener is the basic sharpening tool, it apply hard edges and correct weight to Your active selected model.

#### sSharpen use cases

recalculating sharpen edges on modified geometry

#### SSharpen F6 options

1. sharpness
  - edge angle to witch sharpening is applied - default 30
2. auto smooth angle
  - value for autosmooth angle for selected object - default
3. additive mode
  - ON - apply defined sharpness and keeps other already existing sharp edges
  - OFF - clears all sharp edges before applying its own sharpness
4. sub-D mode
  - ON - do not apply sharp edges and creases making it easier to work with subd after

#### CSharpen

Complex sharpener, is the 2nd sharpening option in hops. It not only creates sharp edges but also apply bevel to it. Csharpen puts selected object in new CSHARP state, that indicates the mesh is beveled

#### cSharpen use cases

applying bevel to simple not beveled object

recalculating bevel on csharpen object that has pending booleans

#### cSharpen F6 options

1. modifiers ignored by Csharpen
  - this is the list of modifiers that can work with Csharp
  - -by default only booleans and solidify are applied whenever the csharpen is used
2. general parameters
  - (a) sharpness

- edge angle to which sharpening is applied - default 30
  - (b) auto smooth angle
    - value for autosmooth angle for selected object - default 60
  - (c) segments
    - number of segments of bevel modifiers
  - (d) bevel width amount
    - bevel size
3. sharpening parameters
- (a) additive mode
    - ON - apply defined sharpness and keeps other already existing sharp edges
    - OFF - clears all sharp edges before applying its own sharpness
  - (b) sub-D mode
    - ON - do not apply sharp edges and creases making it easier to work with subd after

### **SStep**

### **CSstep**

### **Clear SSharpen**

clear sharpen allows You to reomve sharpening egdes, weights and pending modifiers from selected mesh  
It also clears object stats moving it back to undefined

### **clear sharp use cases**

removing sharpening from the mesh

### **clear sharp F6 options**

1. remove modifiers
  - reomoves all bevel modifiers from selected mesh
2. clear sharps
  - remove all sharp edges
3. clear bevels
  - clears all bevel weights from edges
4. clear creases
  - remove creases from edges

**Cutters**

**CSplit**

**Re-Bool**

**Edit mode operators**

**Hard Ops Options**

**release log**





## CHAPTER 3

---

BoxCutter

---



## CHAPTER 4

---

Faq

---



# CHAPTER 5

---

examples

---



---

## Hard Ops Legacy Guide Pages

---

### Hard Ops Introduction Guide

This guide is intended to explain the basics and ideas behind Hard Ops and give some insight to its intentions and future plans.

Hard Ops initially began as a collaboration between myself and AR however over time, it has expanded to encompass a team from all over the world.

*The goals initially were to make a tool that would make our personal lives easier when it came to modelling. But this expanded to be an entire workflow.*

Now the goal of Hard Ops can be simplified into one line.

#### **Hard Surface modelling and workflow assistance. A helping hand.**

There is no magic button. It won't make a better artist. All our tool can do is enable you to be the maximum you.

HardOps also has an older wiki. [Version 8](#) was out at the time I was writing this.

I hope that this guide answers more questions than it asks and makes the experience of using HardOps as easy as possible. I also hope it can give some insight into our hard surface process as well as indicate ideas for future plans and developments.

I would have never been able to make it to where it is if not for the team. They are the driving force behind the development and without a team the work would not have been possible. The collaborative nature of the team and the willingness to invest time has been astounding and has helped this project live far longer than even I initially expected.

### Welcome To the Hard Ops Documentation

---

We hope this guide is able to help you more efficiently find your way around the tool and and put it work making you own ideas come to fruition quicker.

---

Train by AR - Modelled with Hard Ops

## Table Of Contents

### Getting Started

- Introduction
- Preferences
- Install
- Installation
- Getting Started

### Understanding The Functions

- Ssharpen Basic
  - Csharpen Basic
  - Sharpening Basics
  - Csharpen Advanced
  - Mirroring And Symmetry
  - Cstep Basics
- 

### How To Install Hard Ops

---

Using install from file will more than likely **not work**. I would recommend installing the add-on manually by copying the contents of the zip into the add-on directory.

Here is where you would locate it on your system. It goes without saying that putting the zip you received in your add ons folder will also not work.

### Windows 10 / 7

When using Blender a folder is created deeply in your PC for add-ons. Putting it here instead of the branch you are using ensures the next updates also have it installed.

C:/Users/ **USERNAME** /AppData/Roaming/Blender Foundation/Blender/ **2.XX** /scripts/addons/HOps



## Mac

Locate Blender in your applications folder

RMB on the blender.app and select show package contents

## Linux:

I assume linux users know their PC.

~/config/blender/ **2.XX** /scripts/addons

---

Most of this will be assuming Windows is used. So once you open the zip you should see just one folder.

This is what the contents of the zip look like. Its just a folder called HOps. Which is also the internal codename.

All you do is copy the folder into the add ons folder. Then upon opening Blender you are able to go into add-ons.

ctrl + alt + u >> is the hotkey to access use preferences. You can then search for Ops and you will isolate Hard Ops. From here you can enable the plugin.

Now onto discussing the preferences and setup inside of Hard Ops.

---

## Add-On Tab

The recommended addons will show checkboxes when the supported plugins are present.

This can have issues if the naming is different than what it is looking for. BoxCutter for example requires the folder is named "BoxCutter" The namings being searched for are as follows.

- "BoxCutter"
- "Auto Mirror"
- "Mira Tools"

This can be an issue to some however it is recommended to just check the folder naming. We are currently looking into improving it but it was intended to be a quick checklist to assist with troubleshooting.

To troubleshoot further you also can run the following command in the scripting area of the scripting tab.

**bpy.context.user\_preferences.addons.keys()**

This will show a list of all the add ons present. This is how we troubleshoot this area when the plugin detection is not working.

---

## Additional Notes

### Auxillary Addons

There might be a zip file inside of Hard Ops called auxiliary addons.

This has some of the auxiliary add-ons that are recommended. They are able to be installed by placing the contents of that zip in your add-ons folder similar to Hard Ops or BoxCutter.

## Enable Looptools

Looptools is a plugin that is already built into Blender. It comes in handy for a variety of things however for Hard Ops it powers the circle tool behavior.

In fact here is how you can make a circle manually using loop tools and vertex bevel.

- ctrl + shift + b >> vertex bevel
- w >> looptools >> circle
- ctrl + tab >> face mode
- ctrl + x >> dissolve faces
- ctrl + . >> set origin point to individual
- e >> alt + s >> e

That can be a large amount of hotkeys so instead Meshtools Circle exists. Which utilizes looptools.

With circle I can add circle in multifold the speed which I was doing it in the previous image.

Now you're ready to begin!

## Using The Add-On

---

Below is an image of the plugin being enabled and configured. The main thing is to check the addons tab to ensure all required plugins are present. Hard Ops is able to be used without it however it does affect what options are available.

## UI Tab

---

- Pro Mode - this enables additional options that will be explained throughout the guide. Leaving this unchecked is recommended for beginners however enabling it enables all the options.
- Extra Options - this is a series of options ranging from right handed mirroring to additional pie menu preferences such as a ghost button and the addition of the F6 menu. The F6 pie options is enabled by default. So really extra options is just for pie menu users.
- Pie: F6 Option at Top - adds F6 to the top of the pie menu. Enabled by default.
- Pie: Add Boolean Options - adds boolean options to the pie menu for cases in which more than 1 mesh is selected.
- Asset Manager Expansion - for those who purchased the asset manager have the option to allow it to connect to hard ops for easier usage. This options is considered depreciated and may be changed in future releases.

## Drawing Tab

---

- Enable tool overlays - enables the display of the drawing notification system we use for operators. Csharpen / SSharpen / Cstep / Sstep / Merge / Mirror all have a notification drawing system to allow you to see what is going on and then proceed with working.

Here's an example of the drawing system as of version 8 of Hard Ops.

Below are 3 buttons for themes you can use for Hard Ops.

- Hard Ops - My theme for Hard Ops. The default color system.
- AR - Adrian's Theme for Hard Ops. He uses a much plainer system of display with his hard ops.
- Themegrabber - My favorite. This one will color Hard Ops according to your personal theme. I recommend clicking this then saving user settings so the theme is based on you.

You also can set the colors yourself manually and we have put forth many options to customize. The idea was to make this something you can customize and make your own.

## Info

---

This area contains just a note from me.

## Keymap Tab

---

This area tells you that Q is the basis of Hard Ops and that is pretty much it there.

/addons/HOps/keymap.py

In the rare event you need to change the keymap you can drag the keymap.py file into a text editor and edit the hotkey. Or use the button like the following gif.

The hotkeys as of this writing are.

- Q - Hard Ops Menu
- Shift + Q - Hard Ops Pie Menu (may be changed in the future)
- Ctrl + ~ - Modifier/Material/Misc Helper

## Links / Help Tab

---

This area contains links to different areas.

Clicking these will take you to the respective pages.

## Add Ons Tab

---

Hard Ops has become great due to the support of many great tools already existing that are used to minimize certain tasks. Clicking these buttons will link you to downloading them.

### These plugins are highly recommended

- [BoxCutter](#) - is a separate plugin to Hard Ops created by AR and is a secondary project focusing on just cutting alone. I was not the creator of this project and therefore I was unable to add it into Hard Ops.
- [Mira Tools](#) - Mira Tools is a great too for managing lines of complicated points via curve based interpolation. This hosts a plethora of immensely useful tools and is worth checking into on its own.

### These plugins are not required but are recommended. They dont get checkboxes.

- [Batch Operations](#) - Batch Operations is a one stop shop for managing multiple objects worth of modifiers, materials, groups, and more. This is another tool that should be default in Blender but is worth looking into just on it's own. Not to mention the creator has assisted our team with Boxcutter and many other fine contributions.
- [AutoMirror](#) - Lapineige is the creator of this fine tool and is just the finest mirroring solutions. My favorite aspect of this behavior is the bisect and deletion of one half in order to set up the mirroring. This has been integrated under meshtools for quick access. This is one of my favorite tools. However all of these tools are my favorites. Hence my efforts to put them together.
- [EasyLattice](#) - Lattices are not the easiest thing to add to objects. This tools makes lattice application as easy as it would be to add a booltool.
- [Blender Updater](#) - This is for windows users. Its just a quick updater for Blender. I love it. This is not an addon. This is an application.
- [Edge Equalize](#) - This is just an add on for equalizing edges. It has it's uses. I have come to enjoy it and recommend it as well.

The rest of these links also point to their respective pages and provide links to acquiring them. Having these present in hard ops will make the tools even better and we seek to support even more. When it comes to making tools it became apparent that investing in remaking the wheel when one already exists is less than optimal in the scheme of things.

## How to get started with Hard Ops

Hard Ops can seem daunting however all the commands do basic things. So I recommend learning the ropes on basic objects to practice just to get used to the menus. In the next few examples I'll be going over how I use Hard Ops in my day to day work.

## Example #1

### Ssharp / Csharp Workflow

Try out modelling basic shapes and using the sharpeners. I usually do a small trial run to warm up with the tools.

In this example I right clicked my x-symmetrize and remapped it to alt + x in object mode. This is part of my default scene but we allowed some options to be able to be right clicked and allowed hotkeys.

Here you see me using basic shapes and the Q menu to get started.

Its just a matter of my making mesh changes and adjusting the bwidth. Usually as the model gets more detailed the bevel size gets shrunk. There is another workflow involving cstep however this is a bit more advanced. For now we will focus on the csharp / ssharp boolean workflow.

Many of the tools perform ssharp or csharp after the fact in order to do a post operation surfacing of sorts.

One example would be the cslice. Cslice was made by AR as an alternative to my rebool. Both of them have their cases and can be used interchagably depending on the situation.

- rebool requires a pending boolean. Meaning you must set up a boolean with ctrl + (numpad +/-)
- cslice is the process. select two meshes and it will cut. That is all.

In some cases it's easier to cslice however in the example you can see it did not work out so quickly so therefore setting it up was necessary. It's not a matter of one over the other but using whichever works best for your needs.

So continuing on. As we make more cuts and the details become closer. Its important to lower the bevel width distance to keep the mesh looking good.

---

## Example #2

Let's start with a cube.

In the top corner it specifies that this object is UNDEFINED. This ensures when I press Q the options are for meshes that are not considered beveled.

When I first press Q these are the options I am presented.

- SSharpen (soft sharpening without bevelling)
- CSharpen (soft sharpening with bevel modifier added)
- Tthick (add thickness to the volume via solify)

I will start out Csharpening the object then use B-Width to adjust the bevel.

Bwidth will show up on the main menu when an object is beveled via Csharpen

From this point I will use booleans to make additional cuts into the object.

The hotkeys used are:

- ctrl + numpad - (Subtractive Boolean)
- ctrl + numpad / (Slice Boolean)
- alt + shift + x (mirror mirror X axis)

This can be a lot to take in at once so I'll explain what I did here.

- (shift + a) >> insert cube and used it with Slice Boolean to slice the center piece out.
- Added another cube and used Q >> operations >> b-width to put a simple bevel on the object without going into csharp.

This was necessary because it is not recommended to use csharp objects to cut into csharp objects. This will result in bevel weight information being baked in that is not wanted. Here is an example.

- (Ctrl + Numpad -) >> Subtractive Boolean then I used Csharp to apply the boolean to the object and calculate the sharpening.
- Deleted one side, selected the piece to mirror and then the body to mirror across. (alt + shift + x)

By mirroring it across the other side I am able to work on one side while the other is updated automatically. This allows me to cut more things out and the objects will retain their properties.

So continuing with these ideas in mind you can make something complex quickly even in just object mode.

However the combination of both object mode and edit mode tools can allow for some interesting ideas. Especially when you focus on particular shapes and areas.

In this example I used a plane with knife project to force a plane in the ngon surface that resulted from the boolean operation.

Here is an example of me using edit mode to add some custom details.

In edit mode I have shift + ~ mapped to select boundary loop. This allows me to select a region then convert it to only the exterior loop. This isn't a hard Ops feature so you will have to map it manually.

- Isolated the face with only 4 points with shift + h to hide alternate geometry
- W >> Subdivide (x2) / A >> select all >> shift + ~ >> select boundary.
- ctrl + shift + tab to switch to vert selection >> ctrl + I >> invert selection

This step is useful because converting the exterior points to circles would not look as good. So I have become used to separating isolating the edge area for a buffer.

- Q >> Meshtools >> Circle (Nth)
- E >> Extrude >> Alt + S >> Push on normal

While this sounds like a lot of buttons. I press these within a few seconds and can only say that it becomes easier as you get used to it.

In this example I used demote to remove the modifier bevelling so that I could bevel it myself and give it a larger bevel. I also used another cube to cut more out of it.

Even this cube that I used to cut in the front section still has a use. I will simply scale it and then use it to slice out those pieces into manifold meshes of their own.

You might have noticed an error that happened after the initial cslice. Its important to note that sometimes if you cut close to edges you have to lower the bevel width to keep it working correctly.

After cutting you might have also noticed that I used alt + x to symmetrize the object (destructive symmetry) then used alt + shift + x to symmetrize the front cutout to the other side of the front base. The cutout received non destructive symmetry because the two pieces do not touch. This keeps the bevel, mirror, and booleans behaving predictably.

Using the basic tools a decent amount of detail is able to be accomplished in a short time.

This is just one way to use the tool but I hope this helps in getting started!

More Getting Started

## Is there a hotkey guide?

**Yes!**

---

## What is the difference between Ssharpen and Csharpen?

**SSharpen** is short for soft sharpen. We called it that because it does the following.

- edit mode >> select sharps >> mark sharps, bevel weight, subdiv crease
- sets mesh to smooth
- enables autosmooth

To do it manually it looks like this.

Alternatively Q >> Ssharpen does all of the steps for you.

See Ssharpen

**CSharpen** is short for complex sharpen. Complex sharpen does all that Ssharpen does and more.

What else does it do?

- applies modifiers that are not excluded from the list
- adds a bevel modifier to the object
- changes the mesh state to Csharp which changes the options in the Q menu.

The initial options are based on the most likely options for basic meshes. When the mesh is in Csharp state you have a higher likelihood of wanting to see options for adjusting the bevel alternatively if you added a solifidy after bevelling it via csharp it would fold in on itself.

All I can recommend for now is also reading the additional chapters on Csharpen and Ssharpen and play with them yourself to best understand them.

See Csharpen

---

## Operations Menu

The operations contain all the options that can sometimes show up in the dynamic Q menu.

These are the default options that show up under the q menu. There are also additional options that will show up due to multiple meshes being selected or other context based reasons. For example when a bevel modifier is present like in csharp, segments will show at the top of this menu.

---

## Operations options

### UV Preview

UV preview will display the uvs of the mesh in the 3d view.

*This option will only show if UVs are present on the mesh.* The quickest way to get some quick uvs is **Q >> Meshtools >> X-Unwrap**. Otherwise you will need uvs to be present in order for this to work.

Here is an example of using X-Unwrap to get some quick UVs.

The idea behind this tool was to make UV previewing easier for working with only the 3d view. Previously I was using the UV image editor frequently for this purpose but with this I am now able to focus on just modeling for elements that don't specifically need attention.

### Segments

This allows for you to adjust the bevel segments on the object. This option is older and not needed since Bwidth also handles the bevel segments amount via mouse roll.

This will only be shown if a bevel modifier is present on the object.

### Bwidth

Bwidth allows you to dynamically adjust the bevel if one is present.

- if a bevel modifier is not present it will add one however using settings that are different than the one added by csharpen.
- if a bevel modifier is present it allows the modal operation of adjusting the bevel width and segment amount
- it's important to note that the Q menu doesn't show this until a bevel modifier is in place for the initial 3 options. As of now this happens in the csharp / cstep sstate.

### QArray

Qarray is a modal operator for adding array modifiers dynamically without the modifier panel. This is capable of arraying in multiple matrixes via the Q >> axis modifier.

Here is an example of this in action.

You can also use it with multiple axis as previously mentioned.

### Tthick

Tthick adds a solidify modifier self destructively.

This option shows up in the main menu if the mesh status is undefined. Otherwise it is located in the Operations >> Tthick.



This option comes in handy in situations where you just need spacing or thickness quickly that can be adjusted on the fly.

In this example I used (ctrl + numpad -) to cut a cube from another cube. While the boolean was like I used Q >> Thick on the bool shape and used it to make a paneling cut instead of a simple subtractive cut.

## Ssharpen

See Ssharpen

## Csharpen

See Csharpen

## Cstep / Sstep

See Cstep

## Clear Ssharps

This option clears all ssharp information and resets the mesh back to default.

When ran Clear S/C/Sharps does the following,

- removes modifiers from the mesh
- clears all bevel weight / bevel weight / sharp marked edges
- removes auto smooth
- sets mesh shading back to flat
- sets sstatus to undefined / resets sstatus

## Clean Mesh (E)

This option performs a limited dissolve with 0.5 degress and a remove double with a threshold of .02.

This is a new feature added in Hard Ops 8 and is just an experimental way of optimizing planar meshes for continuous boolean.

In some instances. I set the hotkey to alt + ctrl + X.

After using Cslice to make the separation you can see a series of skewed edges.

These can be dissolved manually.

I used C for circle select and then pressed ctrl + x to dissolve. Alternatively you can select all and do a limited dissolve through the X menu.

So now let's see the Clean Mesh in action.

There's still additional things I want to add to this such as temporarily showing the wire of the model during the F6 operation. This may be expanded in further releases.

I must also add that while it dissolves edges to simplify surfaces. This is not always ideal since it goes against the idea of using guidance edges to control shading. So use at your own risk and always check the mesh after the operation. This was only intended to make it easier to do subsequent booleans without mesh error.

## Sstatus System

---

Sstatus is displayed in the left corner of the 3d view.

Depending on the sstatus of the mesh the menus will be different. This is intended to provide the most relevant options based off of the context.

---

## Sstatus Types

### Undefined

- this is the default state of meshes. All options are default and behaviors are also at default.

In the above example you can see that ssharpen does not change the sstatus or options.

### Csharp

- this is the live state for HardOps and means that the bevel modifier has been added. This will cause tools to work with that in mind. This is the most used state and when active changes the Q menu as well.

You can see from the addition of the bevel and csharpening the options change to the more relevant ones.

- Ssharpen (now an updater for Csharpen and the sharp edges)
- B-Width (adjust the bevel interactively for the Csharpened mesh)
- Cstep (for baking the bevel into the mesh)

Cstep - this is for times when bevels are being baked while booleans are also being baked. This is considered an advanced mode and was a tough call to even add to the main menu. To put it simply, with this state the mesh is automatically hidden after operations like cstep / symmetrize / step / csharpen / ssharpen.

In this example you can see how in edit mode the mesh is hidden after subsequent operators. This allows for the bevels to get isolated when booleans are applied. Its a bit of a hack but it works until certain features are added.

## Boolshape

- this was added as a result of Boxcutter. This was done so that when modshapes are made they also have a special q menu. Normally B-width is inaccessible from the q menu unless you go under operations but in this case its best to bevel it without sharpening in order to prevent from booling crease, bevel weight or crease information that is added from Csharpening.

Box cutter has the ability to make mod shapes which have a Q menu of the following options.

- tthick (solidify modifier)
- bwidth (bevel modifier)
- array (array modifier)

This will also be expanded on more in the future when the boolean workflow is expanded.

---

## Sstatus In Action

### Q >> Meshtools >> Sstatus Override

Sstatus override will allow you to set the mesh to another sstatus which will change tool behavior and how operators behave as well. Below is an example of two objects.

- the left cube is taken from undefined >> csharp >> cstep
- the right cube was thickened using Tthick which is available in the undefined menu but not in csharp / cstep menus

As you can see the context changes a bit. This was intended to prevent undesired behaviors. For example if a cstepped mesh is csharpened it will cause issues with the mesh. This system prevents it from occurring.

In the Q menu you can see the context options changing as I am moving through the states. Because Csharpen and Ssharpen involves unhiding the mesh it also causes issues with the behavior of Cstep which requires the mesh be hidden to isolate booleans for Sstep sharpening.

So now that the left cube is in Cstep mode the behavior for mirroring and the Q menu will be different. In the below example I am using alt + x as my custom symmetrize x shortcut but whenever it is ran you can see the mesh is rehidden in edit mode due to the cstep behavior. This advantage can allow you boolean quickly and without having to tend to the mesh in edit mode.

Even though the mesh is in cstep mode I can use the mesh override to reset the mesh back to default. For this example since the mesh's baked bevels are at 12 segments initially. I know overriding to csharpen again won't cause any issues.

I am able to work in Csharp mode again with those options and I can even go back into cstep if I wanted to bake the bevels and go in further.

So with this I can use the tools with different behaviors and shift it around to my advantage. I would recommend trying this system with simple shapes to grasp an understanding.

---

## Mesh Override / Sstatus Reset

### Status Override

Mesh override is located in the q menu under mesh tools.

Status override will provide a pop up window to allow you to change the state of the mesh.

Below is an example of me using the mesh override in action.

Above the mesh was

- csharpened which added a bevel
- cstepped to bake the bevels
- mesh override >> undefined to reset the mesh
- tthick which was available from the undefined main Q menu
- csharpened to bevel the newly solidified mesh

In action has allowed me another level of speed in creating basic hard surface shapes and detailing.

### Status Reset

Status reset is located in the q menu under mesh tools and right below Mesh override

This option simplifies the process of Status Override into being just a quick way to reset the mesh back to undefined.

I use this option almost more than the status override

---

## Understanding C/Sharpen

---

### How does Sharpening work?

Ssharpen is just an automated process that marks sharp edges for hard surface. While also setting up Autosmooth with an angle of default 60.

### SSharpen

Initially whenever a mesh was set to smooth the surfaces would look quite bad.

This was improved with auto-smooth being added later. However it still needed a little more attention. You can see it looks better this way but not perfect.

This can be good enough for most cases however autosmooth on lower numbers might get undesired results. It can be easier to mark certain edges sharp using **ctrl + e >> mark sharps** than to lower the autosmooth threshold.

Sometimes when you are working you want a little finer control and would then benefit from being able to select sharp edges and mark them manually in order to control what is being smoothed exactly. Luckily there are controls for that. **select >> sharp edges**. The select menu has an assortment of options worth looking into.

Ssharpen will do the function of all the above gifs in one step. And thats it.

---

## Csharpen

So to recap Ssharpen is a “sharpener” it doesn’t affect modifiers. It simply “sharpens” the mesh. Ssharpen is a function in the background of many of the functions in Hard Ops. It’s important to know how it works manually to best understand it.

Even this can be a but of work especially when you bring bevelling into the mix. For this reason Csharpen was created. This particular function is much different than Ssharpen but it’s best to explain it manually.

This shape as you can see has many live booltools still active.

I can press **ctrl + ~** for the modifier manager and apply them manually but that is more clicks than I would like to do. This is an example of what the csharpen process will look like manually.

After applying modifiers and calculating ssharps is the next part of Csharpen. Putting a bevel modifier that is set to bevel only marked bevel weight edges. This will give the mesh a nice bevel on sharp edges and make the model look better. Before Hard Ops this was quite a process and was tedious.

So now to demonstrate I will take this mesh with still live booleans and use Csharpen. Afterwards I can use B-width to adjust the now ssharp object.

These two cylinders are examples of the difference between the two operators.

Hopefully this recap on the underlying functions assists in helping make better cubes!

## Hard Ops Boolean system

---

### Boolean Basics

With the Hard Ops 8 update. Booltool is no longer a required dependency. In fact it is recommended to use the new system instead since it’s more catered to our own tools.

**If Booltool is present it will respect it and not assign our hotkeys.**

The hotkeys are as follows

- ctrl + numpad + / performs union
- ctrl + numpad - / performs subtraction
- ctrl + numpad \* / performs slicing

There is also a display for the operator to let you know details about the operation.

As of now the boolean tool has 3 alternate solutions.

- CarveMod (most stable, worst topo)
- Carve (same as Carvemod except it applies it immediately)

- B-Mesh (least stable, best topo result)

CarveMod is the default and will operate similar to the booltool itself. Carve is the same thing as carve mod except it applies it and doesn't keep the boolean live. Below is an example of CarveMod versus Carve.

Bmesh will require there to be guidance edges in order to ensure a successful cut. This is something that's best practiced on a simple object to best practice. Not having guidance edges with bmesh will result in cuts not working properly.

---

## Booleans And Curvature

Curvature can be a tricky thing to use with booleans and to be honest there is no shortcut to efficient geometry however there are some understandings that can make it easier.

Using a boolean on a cylinder is the same in Hard Ops as it would be with just booleans alone.

As you can see the geometric result was quite bad. This is due to the surfacing / curves ngons and the bevel on top. There are a couple of ways to fix this.

-adding geometry around the area to isolate the normal shading

-removing doubles for double vertices in the same areas while merging near miss vertices

This can require a small amount of work however we are always looking into ways to improve this.

The difference between them can be quite immense.

This is how it was cleaned up.

This video was a study of sort of curved surfaces and how it works with inserts and slicing.

<https://youtu.be/B8X7BDMCJ40>

---

## Pokeball example

A nice demonstration of the inadequacies of booleans for hard surfaces can be shown in making a pokeball. I did a small study about how it can be done quickly with booleans versus efficiently with a cast modifier and subdivision blocking.

Booleans

You can see with the surfaces that there is still a little touchup work to be done. However I must say that Hard Ops is not just for booleans it also is a tool for helping get a "finished" result. So now let's try using a cast modifier with subdivision to get the shape more sharper.

Cast Modifier With Subdivision Blocking

In this example the finishing was done via the Csharpen while the initial shape is blocked in using modifiers efficiently and the cast modifier. While this can't work in all cases. It just serves as an example of an alternative way to approach such a shape.

---

## More On Booleans And Curvature

Let's start by converging a cube to a sphere.

And then slicing it.

All the cross edges that are present will make the csharpen not operate correctly without some cleanup. This is due to the fact that the geometry solved the bevel modifier so if the edge flow is less than optimal so will the behavior of the bevel modifier.

Using the CleanMesh(E) function can help in this situation however for the exterior piece we see that cleanup will be needed. So lets clean it up.

I used symmetrize y and then z to put a symmetry division loop that allowed me to rip the geometry with V and then mirror it using mirror mirror which is alt + shift + y then opened up the modifier helper with ctrl + ~ and added the z axis as well.

From there is was a matter of cleanup and merging verts. Alt + M >> Merges points. (its important to merge them in a way that doesnt affect silhouette.) GG >> Edge slide. (when dealing with form you want to move points without interrupting the form. GG >> Edge slide is my favorite change to Blender)

As the geometry gets finer you can have a large bevel width because you are working in finer and finer detail. So lower the bwidth as you work lower and lower. That is why it is part of the main Q menu when the bevel modifier is present.

Let's add an additional cylindrical cutout in the front.

A few things to note in this example.

- when you add a subdivision to a cylinder it turns to something not nice. Just ssharpen it. And then it will hold its form. That is due to the creasing.
- I use ssharpen on spherical shapes to smooth them. I know they have no hard edges and therefore will get smoothed only. If subdivision is there the underlying mesh will be creased so keep that in mind.
- I used the boolean hotkey of ctrl + numpad minus to subtract it then used Csharpen to apply the boolean and make the geo real. This is one of the main behaviors of csharpen.

I can clean it up much easier if I use the same symmetry tricks. However I can also use the auto-mirror to quickly mirror as well and just add the Z axis as well so lets do that.

After the mesh cleaning we got rid of some planar elements at the base but for the most part the bevel still looks bad. So once again its time to fix that geometry.

Merge here, merge there. As you do it more and more it gets easier however this is the foundation of using booleans, and bevels with curved surfaces.

I took the original cylinder used for the bevel and made it real again. First I made it solid in the Settings menu. Then I reset the sstatus which made it a regular shape in the eyes of Hard Ops. From here I was able to do the following.

- apply subdivision using the modifier helper
- csharpen to add a bevel and set the mesh to a csharp form
- clean mesh(E) to dissolve planar verts like the junction on the flat faces

First I will add a cube and rotate it 45 to make a cut. Then use cslice in the Q menu. As you can see the geo is pretty bad. Lets finish cleaning that up.

The main functions used here were the following.

- ctrl + X >> dissolve geometry
- G + G >> edge slide
- alt + m >> merge

And in less than a minute the area was looking much better. All that's left is the sliced piece. Most of these gifs are less than a minute long and I'm also trying my best to do them at a speed that allows readers to follow.

Now this mesh is looking great. But there is no time to admire. Time to mess things up again.

The main piece still has the mirror modifier on the main axis of the form. That limits my ability to work and boolean in those areas because the geo is not real. Because of that I often mirror things. Apply mirror. Re-mirror.

The mirror has some advantages though. Like the previous slice. But let's also do something additive.

And now for the cleanup.

For this double curvature merger the process is more difficult but requires a steady hand and patience. A few notes on this process

- I press alt + m and then to do it subsequently I use shift + r to repeat last command. In my case I used to merge to last and then repeated after selecting the right parts with always the point to meet at being last.
- Also the `ssharpener` didn't completely mark it so I had to do that in edit mode. This is a common thing with curvature. Degress less than 30. It's easier to manually correct than to tweak the `Csharpen` F6 menu which might make it misbehave on other objects.

Now to apply the mirror on the main body and make a larger cut.

For this one it was just a matter of clean mesh and symmetrize and my work was done.

Just to do a few more cuts on this surface. Using the same tools as previous.

And to top it off. Let's detail the center a bit.

And from here mirror mirror. `Cslicing` and then `CleanMesh` after a few merges.

By this point you should have a pretty good idea.

But it never ends.

Once you start assigning materials using the Alt + M in object mode and the Q menu in edit mode you can get some pretty nice results. In fact I use slicing as a way to separate materials and pieces by using this color blockout stage as a way to assist with design.

By the time I am rendering. `Cycles` does all the work of making it look pretty.

Rendering is how I judge the surfacing of the model and the reflections. If the surface was optimized poorly there will be distortions and pinches in the reflection. So it is quite the balancing act. But the goal behind `Hard Ops` is to make surfaces that aren't afraid of the closeup.

This file is available with the other `Hard Ops` files if you want to see it for yourself. The filename is, (008- Drone).q

`Hard Ops` provides no materials but they are easy to acquire. In fact the inserts have placeholder materials that are intended to be replaced by you. I also wrote a blog post about my materials and which ones I use in addition to how my default scene is set up.

### [My Blender Setup](#)

And with this thought process you would continue working on the shape. Cleaning things up and refining things until you have result you like.

## CSharpen

`Csharpen` is the first tool based on the sharpening system and is a crucial tool in `Hard Ops`. The idea of `Csharpen` came about through a need to quickly apply booleans and then sharpen the mesh to keep working forwards.



*We could have called it apply and move on but Csharpen sounds better*

---

## What does Csharp do that Ssharp doesn't?

- apply modifiers that are not listed otherwise as exceptions in the F6 menu
- set up a bevel modifier on the mesh with a specific profile and segment count
- also it ssharpens the mesh after applying the correct modifiers / basically updating the mesh with accurate bevels via weights

## Why does Csharp apply all my modifiers?

It only applies certain modifiers. By pressing F6 you have options on what is being applied when Csharpen is ran.

If the modifier is not highlighted in this list it will be applied. It's as simple as that.

Luckily I also discussed Csharpen in action in the same video as the one where I discussed Csharpen.

## F6 Menu

The F6 Menu is also worth explaining in depth since this is the beauty of this tool.

### *Modifiers Ignored By Csharp*

This area lists all the modifiers that are currently ignored by the Csharp operation. By default it ignores the following modifiers.

- array
- bevel
- boolean
- mirror
- solidify
- subdivision

Shift + clicking additional options will add additional modifiers to the list. This may be expanded in the future to be a user preset list with allowable toggles.

## General Parameters

Sharpness - specifies what angles receives sharp, bevel weight(1), crease(1) for the SSharp process.

Autosmooth Angle - specifies what angle the autosmooth receives in the Object Data tab.

Segments - specifies amount of segments the bevel modifier that is added will receive.

Bevel Width Amount - specifies the bevel width of the bevel modifier

## Sharpening Parameters

Additive Mode - this is an important option. This is also the main reason I use the F6 panel.

- unchecked: this option with recalculate ssharp information every time Csharp is ran. This will remove your previous sharpening and recalculate it. This is used when its easier to recalculate the sharps than it is to correct.  
in this example you can see that since I bevelled in edit mode the bevel weight marked edges of offset into flat surfaces. These surfaces don't benefit from bevelling because they are flat. Since the form is simple additive mode cleans it up easily.
- checked: this option will add new ssharp information and leave and previous information Sharp / Crease / Bevel information that existed before. This is the default behavior.

It is generally preferable to not have your sharpening overwritten. That is why it is checked by default. I recommend this as well however experimentation will make it apparent which tool will work for you.

---

To better understand there is some additional reading.

- Csharpen Advanced
- Ssharpen
- Sharpening Basics

## Csharpen Advanced

---

To best understand Csharpen in action it's best to see and use the tool on something simple to understand how it applies to you.

## Csharp In action

In this example I csharpened the main cube then added a duplicate and used the booltool shortcut of **ctrl + (numpad -)** to cut the cube into the first one. Then with the primary object selected I used Csharpen to apply the shape while respecting the modifiers in place in order to calculate the new bevelling information as I worked.

---

So now the settings have been covered lets see it in action once more.

After setting up Csharpen I was able to cut the shape and mirror it while also doing subsequent cuts on the object to get additional detail. The important thing to note is how I don't need any additonal UI or buttons. There is hardly a need for the modifier panel or even the T panel. The goal is to make the workflow fast and efficient.

## Cstep (Basics)

---

## What is Cstep?

---

Cstep is basically a workflow involving baking bevels destructively and working forward with subsequently smaller bevels.

The workflow of cstep was developed primarily by AR and is highly destructive however can get rather unique results. I still consider this workflow experimental but I also highly recommend trying it out.

---

## How does Cstep work technically?

---

Let's say you csharp and object and adjust the bwidth.

From this point all I can do is adjust the bwidth at this level. Even when I do subsequent cuts they will be at this bevel level. Even further cuts that are csharpened will be at this bevel level. In order to maintain. I will have to lower the bwidth to accommodate.

This is generally not the best choice since I am having to adjust in order to compensate. So now let's see how cstep helps with this situation. By making the bevels I can get more and more micro.

In this example you can see I used ctrl + numpad minus to cut the cube into the main shape after activating cstep on it. This technically

- applied the bevel modifier
- added a new bevel modifier set to weight
- hid the mesh in edit mode and removed marked bevel weights

This is how we "bake the bevels". So Cstep in a way is "Bake Bevels"

This is why when you cstep you no longer have access to bevel width. You are baking the bevel then re-adding it while removing previous bevel information.

In the above example you also see me use Sstep. Sstep is the process of calculating the bevels based off either modifiers or the mesh that is visible in edit mode. Unlike csharp cstep focuses on the mesh that is visible and does not unhide unless it is being baked and cleaned (bevel weight removed).

In this example take note of the following.

- Cstep is how I bake each level before going further
- Sstep is how I calculate the sharps after setting up the booleans for that level
- Sstep has an F6 menu for which modifiers it can ignore but really it doesn't need to be messed with unless you are focusing on the sharpening amount which generally is also left at default.

---

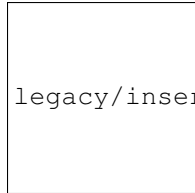
## Alternate Use Of Cstep / Sstep

In this example I start out with the Csharpen then use B-width to increase the segments to 12. I then cstepped it in order to bake the bevels.

## Insert System

### What are the Inserts?

Hard Ops has an insert system for using premade meshes for quick greebling and detail work. It's intended to be used in a pinch when detail is needed quickly.



The insert system can be brought up via the Q menu via inserts and also has a top option for an asset scroller which allows you to cycle through assets.

---

### Insert Types

Orange Inserts are just basic inserts. These are just meshes you place.

Orange inserts can be inserted on a series of faces and can be scaled immediately after selection via a modal operation. This allows for scaling to perfection. It even sets the pivot points to individual origins so they retain their position. These work in both object and edit mode.

Red Inserts are inserts made to be boolean cut into surfaces.

Adjustable inserts are red inserts that can be adjusted prior to being applied.

Red inserts also do not get inserted on faces. They always are inserted at the 0,0,0 of the 3d view. This is because they don't behave well being placed automatically due to the hook modifiers that the adjustable ones have. These inserts also have an AP that allows for surface snapping while also seeing the outside perimeter. This allows for precise placement.

### Red Inserts Details

Red inserts are built up out of these 3 pieces.

- AP or alignment plane (used for aligning the whole insert)
- BB of BoolBox (used for the boolean to make room for the OB)
- OB or Object (the object being inserted)

[Update Log Explaining Red Inserts](#)

---

## Inserts / Materials Sharing

Inserts come with basic materials assigned. These are placeholders that are just indexes for materials that you would want to replace them with.

In this example I will add some inserts to a cube.

As you can see the materials have the .00X numbering due to them being subsequently added. If you select both inserts and go to the operations menu there is an option for Material Link.

This is here just for material localization however most of the time I replace these with the placeholders I have present and then use the shift + L >> material link.

This is one of the more hidden options however it was made for just this purpose.

---

## FAQs / SFQs

### How Do I Install Hard Ops / Boxcutter?

#### See Install

### Why is the add-on checker not working?

The recommended addons will show checkboxes when the supported plugins are present.

This can have issues if the naming is different than what it is looking for. BoxCutter for example requires the folder is named "BoxCutter" The namings being searched for are as follows.

- "BoxCutter"
- "Auto Mirror"
- "Mira Tools"

This can be an issue to some however it is recommended to just check the folder naming. We are currently looking into improving it but it was intended to be a quick checklist to assist with troubleshooting.

To troubleshoot further you also can run the following command in the scripting area of the scripting tab.

**bpy.context.user\_preferences.addons.keys()**



This will show a list of all the add ons present. This is how we troubleshoot this area when the plugin detection is not working.

---

## Why am I having issues when I boolean certain areas?

---

### What is Cstep / Sstep?

In short.

**Cstep** - bakes bevels to the mesh and hides it in edit mode for future booleans.

Cstep is similar to Csharpen aside from the fact that it applies the bevels before re-adding it. This is where it differs from csharpen. The moment you cstep you will notice you no longer have the ability to adjust the bwidth. **This is because the bevel was applied and readded.** This means that now you can boolean the object while being able to adjust the bevel for those singular areas.

**SStep** - calculates sharpening from the mesh that is visible in edit mode. Compared to SSharpen which will unhide the mesh to calculate ssharps.

SSharpen - unhides the mesh before ssharpening.

Sstep - only calculates ssharp from visible mesh in edit mode

Sstep is basically the same as ssharpen except for un hiding the mesh. I also use this when I dont want to unhide my selection to calculate however the main purpose of this tool was for cstep workflows.

See CStep

---

### Why is my mesh hidden when I Cstep?

Cstep hides the mesh when used. See [How does Cstep work technically](How does Cstep work technically)

This example shows what happens when you cstep. After hitting the button in the Q menu the mesh type will say Cstep in the top left. Also the bwidth will no longer be fun. This is because it has been applied. When you go in edit mode it will be hidden.

- press alt + h to unhide mesh
- press h to hide mesh

In the example it shows me undo to get back to csharp state but understanding the states is essential to using the tools correctly and understanding the context.

### Can I get out of Cstep?

Absolutely. We added the ability to override states which can be useful to putting meshes back to the beginning for using alternate workflows. Here is an example.

**q >> meshtools >> status reset** will reset any mesh back to a default state. Causing the Q menu dynamic options to also be default instead of cstep oriented.

In this example I use cstep to block it in sort of an bake a bevel while then using the Status reset located under **q >> meshtools >> status reset**. By resetting the sstatus I was able to use Tthick which is a basic option under meshes that have no sstatus. From there I went back through a csharp / rebool workflow and ended up back in a csharp workflow to finish off the shape with finer details and a bevel with 1 segment for the edges.

## Why is it called Sstep / Cstep / Ssharpen / Cssharpener?

The naming made sense at the time but the functions are best described like so. Someday the tools will be unified into something more logical however in the meantime we needed some sort of terminology for naming.

---

## Can I add my own inserts to Hard Ops?

No. However it is possible but not recommended since it's not user friendly. For custom inserts I recommend the [Asset Management](#). This is a temporary solution but we do have plans to expand this down the road.

---

## What do the different inserts mean?

Orange Inserts are just basic inserts. These can be inserted in edit mode oriented to faces.

Orange inserts can be inserted on a series of faces and can be scaled immediately after selection via a modal operation. This allows for scaling to perfection. It even sets the pivot points to individual origins so they retain their position.

Red Inserts are inserts made to be boolean cut into surfaces.

Adjustable inserts are inserts that can be adjusted for additional length prior to being applied.

Red inserts also do not get inserted on faces. They always are inserted at the 0,0,0 of the 3d view. This is because they don't behave well being placed automatically due to the hook modifiers that the adjustable ones have. These inserts also have an AP that allows for surface snapping while also seeing the outside perimeter. This allows for precise placement.

Red inserts are built up out of these 3 pieces.

- AP or alignment plane (used for aligning the whole insert)
- BB of BoolBox (used for the boolean to make room for the OB)
- OB or Object (the object being inserted)

[Update Log Explaining Red Inserts](#)

---

## Help! Why doesn't this add-on show activated in the add-on checker!

By using this line in the scripting area of blender you can see all the add-ons activated.

```
bpy.context.user_preferences.addons.keys()
```

The Boxcutter add-on folder requires the naming be "BoxCutter"

You can also find the foldername in the addon file area.

This is how it would be fixed in the add-ons directory.

Now I can start boxcutter with Alt + W

Mira tools also is the same way.

In my add ons folder it is just called mira\_tools. For this addon it can get complicated since the naming is rather specific.

I must add that the add on checker doesn't really matter. It was attempted to be a convenience thing. However some issues that users find with it is due to the naming.

## Material List / Material Helper

---

### How to use the Material List

Q >> Settings >> Material List

This will display the materials present in the scene. If you have no materials created none will be shown. In my scene I have defaults that I use for quick colorization. This is available per object in object mode and on a geometric selection via edit mode.

Here is an example of the material menu being used in edit mode.

In this example you can also see how it works in edit mode.

It's almost important to note that multi materials on the same mesh using geometric separations and bevels will look better with an even number bevel segment amount. You can see this in the above example as well.

---

### Where do you get more materials?

I get my materials from a variety of sources.

- Disney Ubershader
- Metal Pack
- Car Paint
- Ubershader
- Free PBR Material
- PBR Material
- Low Noise Pack

Shaders are beyond the scope of this plugin but these locations should be enough to get you started. These days I use materials more for color blocking and then swap them out for the intended ones after baking or downpacking.

In my default scene I keep a sphere on layer 0 with materials held that allow me to hold some quick preferences. From here I am able to delete the sphere later and only the used materials are kept.

Usually my goal is just a quick presentation render. Similar to this.

---



## Is there a way to adjust shaders as well?

### How to bring up the Material Helper.

\*\* Ctrl + ~ \*\* brings up the modifier helper. The 2nd tab is the material helper.

Currently we still are developing this so there may be some issues however it does exist as an option to experiment with in the meantime. The modifier helper is quite useful for managing modifiers and we hope the material helper is just as useful.

---

## Machin3 Drone Breakdown Shots

MACHIN was so kind as to allow me to use his drone as the image for Hard Ops 8 as had operatives previous however he also shared some images of his process that is worth sharing.

He has the drone for purchase as an asset [here](#). It is definitely worth a study and up close look. It was amazing to see someone push the limits even further.

Also congratulations to his [top row](#).

## Mirroring And Symmetry

Mirroring is important when it comes to keeping a mesh symmetrical however in Boolean workflows having active mirror modifiers isn't always ideal. So we attempted to provide many options for how to best approach symmetry.

---

When it comes to Hard Ops there are 3 ways mirroring is dealt with.

- Mirror Mirror - (integrated) non destructively via modifiers. works based off of object selection
- [AutoMirror](#) - (separate) this plugin is capable of bisecting a model and adding a mirror modifier
- Symmetrize - these options use blender's own symmetrize for all axis and is built in. This is my default choice for symmetry

The tools depend on the case however they all provide unique solutions for particular issues pertaining to mirroring.

---

## Mirror Mirror

This was one of the first plugins I ever was involved with. It was made by me and Robert Fornoff who also was a part of the building of Hard Ops. Initially this was a separate plugin but now it has been merged into Hard Ops to use the drawing system and reduce outside dependencies.

Mirror Mirror has 3 hotkeys.

- alt + shift + x (X Symmetry)

- alt + shift + y (Y Symmetry)
- alt + shift + z (Z Symmetry)

The beauty of this tool is that the intended behavior was selecting two objects and after pressing the before mentioned hotkeys would mirror the object across another object's axis. You can also use the mirroring hotkeys on a single object and that would give it a mirror on itself across an axis.

To demonstrate MirrorMirror in my workflow.

After slicing off one half. I used my alt + x symmetrize shortcut to make the body symmetrical. However for the piece I sliced off I was able to use **alt + shift + x (X Symmetry) to mirror it across the body** and therefore only have to work on one half. This is something I do quite a bit and also use MirrorMirror in blocking to quickly have mirrored halves of a character that I can then later adjust easily and with the original origin axis still kept.

In fact just to show how I go about character blocking using mirror mirror, I provided an example.

I personally can't work without this enabled so it was only natural to add it into the core of Hard Ops 8. I would recommend getting used to it and spending some time with it in order to better understand how it works.

---

## Automirror Addon

Automirror is a tool made by [Lampaigne](#). This tool fullfills a rather unique need and does it so well I need not bother making a tool of my own. I cannot recommend this add on enough for mirroring. However this tool has a rather specific behavior that is worth going over in depth.

The support for automirror is in the pie/tpanel/qmenu.

In this menu you can either just click Auto-Mirror to mirror on the default X-axis or choose and axis as well as set positive or negative and set up the mirroring type before initializing. Generally it is set up to mirror x by default.

You can see now after setting up the Automirror after the block in that the model was automatically mirrored for the following sectional cuts that were made. Its also important to note that I was unable to cut across into mirrored territory. This is something that can be fixed by just applying the mirror modifier but it's important to note:

- You cannot cut mirrored meshes using booleans if the mirror is not applied. This is because one half of the mesh is not real.
  - you can fix it by applying it
  - you can work around it by not cutting near the mirror point. This is where the fun begins. But if you do want to cut it globally just apply the mirror and automirror again if needed.

You can use the automirror with these rules in mind to pull off some awesome stuff.

For example, Here's a quick 2 minute example of me using this particular behavior with Hard Ops.

In this example I set up both x and y symmetry which isn't always ideal.

So when utilized properly you can get some quick results rather fast.

In this example I set up just x symmetry which is more ideal.

Within 2 minutes I can get pretty far with the basic operators. And this is due to the speed and time saved with the auto bisect behavior. I cannot recommend [AutoMirror](#) enough.

---

## Symmetrize

So last but definitely not least is blender's own symmetrize. This tool is fantastic. I am personally a big fan of how quick it is and how accessible it is. Basically our operator is just a frontend of the one that is located in the w menu by default.

Here is blender's own symmetrize. All we did was make it more accessible to our needs. The reason this is used often is because it is mirror and done. This prevents boolean errors and allows for more veratility. Even while being the most destructive.

I am still a big fan of this and even have it mapped to alt + shift + x in edit mode in my version of Blender. However now let's see symmetrize in action. Keep in mind I have mine mapped to alt + x via right clicking it in the q menu. There are some differences that I will go over more after this example. I also usually combo this with mirror mirror as well so look out for the shortcut (alt + shift + x)

By working this way I can make vertical cuts without limit and less issues than with a dynamic modifier in place. So theres a time and place for all of them but to recap.

- mirror mirror - is used for separate objects that need to be mirrored across another object's x/y/z axis
- symmetrize - is used to mirror objects across themselves without regard for what is on the other side. I consider it mirror and be done.

So with that out the way the symmetrize we created is also able to behave slightly different due to mesh sstatus. One example is cstep meshes will be hidden automatically after being symmetrized. Keeping the user moving if aware of the situation and sstatus.

Here is an example of that.

## Links

---

[Setting Up Blender](#)

[Substance Painter Baking With Blender](#)

---

[Hard Ops 8 Update](#)

[Hard Ops 0079 Update](#)

[Hard Ops 007 Update](#)

[Hard Ops 0065 Update](#)

---

[BoxCutter 4 Update](#)

[BoxCutter 1 Update](#)