
Source2Blender

Release 1.0

TF2 Blender Community

Nov 04, 2023

TABLE OF CONTENTS

1	Credit	3
2	Contents	5
2.1	Installing Programs	5
2.2	Team Fortress 2 Vanilla	6
2.3	Team Fortress 2 Workshop	19
2.4	Source Filmmaker	20
2.5	Garry's Mod	25

This documentation will serve as a beginner-friendly guide to porting assets from games featuring Valve's game engine, Source, to Blender, the most popular free and open source 3D computer graphics software.



Note:

This project is under active development and will receive constant updates. If you need any help, or want to give us suggestions, feel free to join the [TF2 Blender Discord server](#) and tell us about it!

Feel free to contribute to this documentation by submitting a pull request on the [GitHub repository](#)

CHAPTER ONE

CREDIT

This documentation was made by [Woha](#), [Hisanimations](#), and [Meer](#), with additional assistance from the members of the [TF2 Blender Discord server](#). It was adapted to ReadTheDocs by Espion ([espiondev on GitHub](#)) and Woha ([WilliamWoha on GitHub](#))

CONTENTS

2.1 Installing Programs

2.1.1 Installing Blender Source Tools

Blender Add-on, Imports multiple file types, including .SMD which holds Animation Data. Used alongside Plumber for importing of Maps.

- Download [Blender Source Tools](#), it's a .ZIP file. Don't open or extract it. Make sure your Blender version is up to date, as the download link says it's supported for versions 2.92 and higher. However, older versions of the Add-on going back as far as Blender 2.66 can be [found here](#).
- In Blender, go into *Edit > Preferences*.
- In the Add-ons menu, click on the *Install...* button.
- Select the .ZIP file you downloaded from the [Blender Source Tools](#) website.
- Click on the check box to enable it.

Blender Source Tools is now installed. You'll notice its Import option shows up under *File > Import > Source Engine* (.smd, .vta, .dmx, .qc).

2.1.2 Installing Plumber

Blender Add-on, Imports multiple file types, including .VMF. Used alongside Blender Source Tools for importing of maps.

- Go to the [Plumber](#) download page, click on the big bold text in front that says Plumber. Scroll down, download the .ZIP that's appropriate for your system. It's available for MacOS, Linux, and Windows. Do not extract or open the .ZIP when downloaded.
- Make sure your Blender Version is up to date. The site says it supports 2.82 and higher but for compatibility with Blender Source Tools, make sure you have 2.92 or higher.
- In Blender, go into *Edit > Preferences*.
- In the Add-ons menu, click on the *Install...* button.
- Select the .ZIP file you downloaded from the [Plumber](#) releases page.
- Click on the check box to enable it.

- It should automatically find any Steam products it's compatible with, such as Team Fortress 2, Left 4 Dead, Portal, Half Life 2, etc. It depends on what you have installed. Make sure you have a properly working copy of Team Fortress 2 and it's in a valid Steam location otherwise it won't detect it.

Plumber is now installed. You'll notice its Import option show up under *File > Import > Plumber*.

2.1.3 Installing SourceIO

Blender Add-on, Imports every file type except the ones used for Animation.

- Go to the [SourceIO](#) download page, click on the big bold text that says SourceIO. Scroll down, download the file called SourceIO.zip. Do not extract or open the .ZIP when downloaded.
- In Blender, go into *Edit > Preferences*.
- In the Add-ons menu, click on the *Install...* button.
- Select the .ZIP file you downloaded from the [SourceIO](#) releases page.
- Click on the check box to enable it.

SourceIO is now installed. You'll notice its settings show up under *File > Import > Source Engine Assets*.

2.1.4 Installing GCFscape, Crowbar, BSPSource

These programs are either unpacked into folders or have an installer. Just go to their download links by clicking their names in the section they're referred to. They're programs that run outside Blender, and not as Add-ons.

2.2 Team Fortress 2 Vanilla

Throughout the pages, you'll find clickable text where a link is provided. Just click on it or right click and 'Open in new tab' to see the contents. Most mouses can click with the scroll wheel. Clicking links like this will also open them in a new tab.

Depending on the category you'll find two methods or only one. Where there are multiple methods (Such as in the [Maps and included props](#) Section), the benefits and drawbacks will be listed. Sites with only one method means other methods aren't as worthwhile, or there only is one way to do it.

It's best if you have an up-to-date copy of Team Fortress 2 installed as nearly all of these methods will be relying on obtaining files present from within the game.

2.2.1 Contents

Quickstart

This page covers useful add-ons made by Hisanimations that streamlines importing certain Source engine assets to Blender.

The TF2 Trifecta

Importing characters and cosmetics has been simplified by [Hisanimations'](#) Blender add-on, the [TF2 Trifecta](#). This add-on has numerous useful features, such as:

- Importing mercenaries in either IK (easier for general animation and posing) or FK (compatible with importing animations).
- Searching for cosmetics and automatically parenting them to a selected mercenary.
- Updating the Blender files that contain cosmetics, mercenaries, and props to the latest version.
- Applying paints to cosmetics, fixing broken cosmetic materials ([See here](#)).

The YouTube tutorial for the trifecta can be found [here](#). The original TF2 Collection tutorial can be found [here](#).

The Buildings port

[Hisanimations](#) has made a buildings port, including a fully-functional sentry, dispenser, and teleporters ([See here](#)).

Maps and included props

Method 1 (Recommended) uses three tools, all linked in that section. This method is recommended as it makes the maps significantly easier to work with. All imported items are organized into collections and very easy to work with and customize.

Method 2 (Alternate) uses only one addon, called SourceIO. It's a one click solution and way easier than *Method 1 (Recommended)*, but the names of objects becomes messy, and there's extra cleanup required as extra objects such as the map hitbox is also imported. It's closer in looks to TF2 as it uses its own shader, so if you want the true TF2 look, then use this. *Method 1 (Recommended)* also works fine but is better suited for applying your own style or flair to your renders and animations.

Both methods require a fully working copy of Team Fortress 2 and a recent copy of Blender. If you are unable to get these methods to work, it is recommended to use the latest version of Blender.

Method 1 (Recommended)

Summary of Method 1

- Download BSPSource, Plumber, and Blender Source Tools. Install Plumber and Blender Source Tools as Add-ons into Blender (This step is only necessary on first time use).
- Locate the map you wish to bring into Blender by going to [game_directory] / tf / maps.
- Convert the .BSP file into a .VMF file using BSPSource.
- Import the .VMF file from *File > Import > Plumber > Valve Map Format (.vmf)* in Blender.
- Disable all lights except `light_environment`, and set Color Profile from *Filmic* to *Standard*.

Full Guide of Method 1

Note: Always get the most recent version of a program or Add-on linked here.

Blender Source Tools, BSPSource, and Plumber will be used for Method 1. The steps to install these can be seen at the *Installing Programs* section.

Plumber was originally called `io_import_vmf`, and it's by the same developers. It's currently in Beta but significantly superior to `io_import_vmf` and that's why we've linked the page to download that instead.

Convert a .BSP to .VMF

Important: You need to install [Java](#) to run BSPSource.

- Download [BSPSource](#) and extract all files to a folder.
- Open `bspsrc.jar` from this folder.
- There's many options in the program. For now, just click the button for *Add*. From there, go to [game_directory] / tf / maps and choose the specific map file you want to convert.
- Go to the *Other* tab in BSPSource, and enable the checkbox labelled *Extract Embedded Files*.
- Once done, just click the *Decompile* button in the bottom right. There's no need to edit the other settings, though you're free to play around if you know what you're doing.
- A file browser will show up for where to put the .VMF file. You can choose any location, but it's best if it's a place you can easily come back to.
- Click on *Decompile*.

Your .VMF file has now been decompiled and is in your output folder. You'll notice another folder in that location with the same name as the .VMF file. We'll use this later. If such a folder doesn't exist, then don't worry about it.

Important: During the time that BSPSource is Decompiling the map, it will show logs of what it's doing. There is an **Errors & Warnings** box visible. If the process worked, then this box should be mostly empty. If [Warning] is shown in this, then it should be fine. If an [Error] is shown in this, however, then it may have failed. You'll have to come back to this step if the map doesn't work later.

Bringing The Map In

You can now import a .VMF file from the *File > Import > Plumber > Valve Map Format (.vmf)* button (Make sure Plumber and Blender Source Tools are installed). Browse to the location you stored your .VMF file which you Decompiled using BSPSource earlier.

- Make sure that in the Import settings, the game is set to Team Fortress 2.
- Set the Scale to be set to 0.1, and the Light Brightness set to 10. This is so the map is compatible with the Hisanimations Characters port and TF2 Collections Port.
- In the folder space underneath the *Game*, type the name of the folder with the same name as the .VMF. So if for example, your map file is called `pl_pier_d.vmf` then there should also be a folder called `pl_pier_d`. Write `pl_pier_d` in that space. If you don't have such a folder, you can skip this step.
- Then click the .VMF file, and click *Import*. That's it, you should have everything.

The installation steps are not necessary to do again. It's really just as simple as, Turn .BSP into .VMF with BSPSource, Bring .VMF into Blender.

Go to *Finishing Touches (Both Methods)* for advice on clean up and additional useful things to know about within Blender regarding these maps.

If you wish to familiarize yourself with the whole process, or see an overview, a *Summary of Method 1* is written which you can refer to.

If any problems occur you can ask for help on the [TF2 Blender Discord server](#).

Method 2 (Alternate)

Summary of Method 2

- Download SourceIO and install it as an Add-on into Blender (This step is only necessary for first time use).
- Go to *File > Import > Source Engine Assets > Source Map (.bsp)* in Blender.
- Locate the map you wish to bring into Blender under `[game_directory] / tf / maps`.
- Select everything by pressing A, then Shift Click an Empty.
- Press N under the 3D Viewport to open the sidebar, go to SourceIO, and click *Load Entity*.
- Disable all lights except `light_environment`, and set Color Profile from *Filmic* to *Standard*.

Full Guide of Method 2

SourceIO will be used for *Full Guide of Method 2*. The steps to install these can be seen at the *Installing Programs* section.

Bringing the Map in

Important: Carefully follow these instructions. If you make a mistake, you will have to create a new, blank, project, as this addon directly reads off the .BSP in real time and doesn't allow that file to be changed or edited. There's also a lot of cleanup work to be done so, it's recommended to have a completely empty Scene for this process.

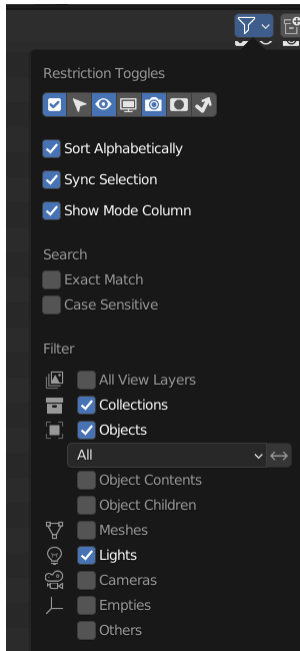
- Go to *File > Import > Source Engine Assets > Source map (.bsp)* (Make sure SourceIO is installed).
- Select your map of choice. The map **MUST** be in your TF2 game directory. It will be in [game_directory] / tf / maps. You can use the name filter to narrow down the results.

Once loaded in, maps will be quite bare-bones. The lighting will most likely be too dark, and the props aren't there. There are a few things to set up.

- Press A to select all objects within the viewport. Then Shift Click on an Empty.
- An Empty is a placeholder. You'll notice a lot of these in places where Props are supposed to be.
- Everything should be selected, with an Empty being highlighted as Orange.
- Hovering over the 3D Viewport, press N to open the side panel. There will be a *SourceIO* tab. Click on that to open it.
- Click on *Load Entity*.
- It might take some time so please be patient. If done right, all props should show up without any error messages.

The names of the props will be a huge mess, and every single prop will end up having its own Collection. For this reason, *Method 1 (Recommended)* is recommended, but otherwise, the overall Map and the Props on the map are pretty much the exact same.

The lighting is going to appear strange because in Eevee (Blender's default render engine) has a maximum of 128 lights. Filter the Outliner (the place where all objects and things in the scene are shown) by lights with the following settings:

**See also:**

For a full list of Eevee's limitations, you can consult [this page](#) from Blender's official manual.

Finishing Touches (Both Methods)

- Use *Material Preview* mode to confirm that all materials are actually fully functional before you do anything else. All textures should be visible and no part of the map should be white.
- Use Eevee if you want a true TF2 look. Cycles will get you very different results.
- There's unfortunately a limit of Eevee which there's no way around. It can only have 128 active lights at once, while a lot of maps in TF2 end up having significantly more than that. Unfortunately the only way around this is to use Cycles, which doesn't have a light limit, but another alternative is to maintain the majority of the look by turning off every light except the one which starts with the name `light_environment`. This is the 'Sun' light and is responsible for nearly all outdoor lighting and shadows present on the map.
- If you want more accurate TF2 colors, go to Color Management, and set the Color Profile from *Filmic* to *Standard*.
- To apply the Skybox, check the [Applying Skybox](#) section.

Note:

In some maps, for example `p1_badwater`, some universally used props will look a bit off, such as the rocks used in the starting area for the payload cart. This is because these props have multiple different skins used by different maps. A script is being developed to make it easy to change skins, but if you currently want to do it manually, then go to the Materials section of this object and make it so all the assigned faces are of a different material slot instead. If you know how Materials and Assigning works, this shouldn't be too difficult for you to do.

If you used SourceIO to import the map, in the sidebar (brought up by pressing the N button), there should be the option to change through different skins easily.

Applying Skybox

It is necessary for Plumber to be installed for this next part. SourceIO cannot do this.

To apply a Skybox, first find it in the corner, then select all the objects that are near it. Make sure none of the objects are linked to the rest of the map. If they are, then you have to go into Edit mode to separate it.

After that, Shift Click the Empty. This is important because otherwise it won't line up.

Finally, click Object, then Transform VMF 3D sky.

A GIF has been attached to show the process visually.

Individual Props

This section is written as a way to obtain individual props that are universally used in maps stored in the TF2 files, such as Barrels, Control Points, or Gates. Some maps will have props that aren't used universally, and are exclusive to them. These can still be obtained with both methods.

Method 1 (Recommended) is the better of the two, as the work is already done. [Hisanimations](#) has already made a fully working Props, Weapons, and Cosmetics Ports file that you can use for yourself. His [YouTube video](#) explains what it is and how to use it. If you have questions or need help with this port, you can ask on the [TF2 Blender Discord Server](#) as he's an active member there.

Method 1 (Recommended) is also significantly more space effective. The download of it takes up 5.2 GB while doing it using *Method 2 (Alternate)* will add 7.7 GB to your TF2 game directory.

Method 1 (Recommended)

Watch the [Hisanimations TF2 Blender Weapons, Cosmetics, and Props port video](#) and follow the instructions.

Method 2 (Alternate)

Summary of Method

- Download GCFScape, and SourceIO. Install SourceIO as an Add-on into Blender.
- Use GCFScape to extract the necessary files from `tf2_misc_dir.vpk` and `tf2_textures_dir.vpk` into `[game_directory] / tf`.
- Use SourceIO to import .MDL file of choice from the extracted folders.

Full Guide of Method

The process is rather simple. It only requires a bit of setup, then the importing of the prop should be doable with a few clicks.

- Download [GCFScape](#), and [SourceIO](#). Install SourceIO into Blender (installation guide listed in [Installing Programs](#))
- Go to `[game_directory] / tf` and open the file called `tf2_misc_dir.vpk`. It should open through GCFScape.

- In GCFScape, right Click the **Models** folder, click *Extract*, and Extract it to [game_directory] / tf. Don't try to Drag and Drop as it's extremely laggy and buggy. The extraction will be 2.5 GB in size so make sure you have the space for it.
- After that, go back a step, then go into the **Materials** folder. Inside of this is another folder called **Models**. Extract this to [game_directory] / tf as well.
- Close GCFScape. Go to [game_directory] / tf and open the file called tf2_textures_dir.vpk. It should open through GCFScape, just like the previous .VPK file.
- This next step will add 5.3 GB to your game folder size, so make sure you have the space for it. There should be only one folder inside, called **Materials**. Open this, then find the **Models** folder. Extract this folder to [game_directory] / tf. You can now close GCFScape.

All of that was for setting things up. Once that's completed, all you have to do for bringing a Model in is to open Blender, click *File > Import > Source Engine Assets > Source model (.mdl)*, and choose the .MDL file you're after inside the **Models** folder. It should have textures set up and everything. The above steps don't have to be repeated.

Character and Rig

Method 1 (Recommended) is the best of these, as the work is already done. [His animations](#) from the [TF2 Blender Discord server](#) has already made a fully working Character Ports file that you can use for yourself. His [YouTube video](#) explains what it is and how to use it. If you have questions or need help with this port, you can ask on the [Discord server](#). Using this method is recommended in most cases, but if you intend on animating, then *Method 2 (Alternate)* is slightly better, as it gives better FPS when working with keyframes, but you lose out on the ease of use and the support.

Method 2 (Alternate) is to get the stuff directly from the in-game files. This method is recommended for animation work where you need the maximum possible performance. You can still animate completely fine with *Method 1 (Recommended)*, but this one just gives a slightly higher FPS number.

Method 2 (Alternate) requires a functioning copy of Team Fortress 2 and a recent copy of Blender. You don't need a copy of the game for method 1.

Note:

There is also a difference in quality for these methods. Method 1 uses the characters obtained from Source Filmmaker, which has higher quality models intended for animation work. Method 2 will use the models present in the game, which are of lesser quality. If you want to use Method 2, and also want the higher quality models, then the process for the files is the same, except they must be obtained from tf_movies from SFM.

Method 2 having a higher performance compared to Method 1 isn't because of the difference in the models used, but also because Method 1 has extra features packed into it and actively running scripts. The performance is small, though, so unless you REALLY need that small gain because of a weaker computer or if you're animating a very complex scene, just use Method 1. A funny thing is, if you play an Animation with the Armature selected, it straight up halves the performance, so do make sure that nothing is selected when playing something back.

Method 1 (Recommended)

Watch the [Hisanimations TF2 Blender Character port](#) video and follow the instructions.

Method 2 (Alternate)

Summary of Method 2

- Download GCFScape, and SourceIO. Install SourceIO as an Add-on into Blender (This step is only necessary for first time use).
- Use GCFScape to extract the necessary class files from `tf2_misc_dir.vpk` into a folder of your choice (This step is only necessary for first time use).
- Import the .MDL of the character from *File > Import > Source Engine Assets > Source Model (.mdl)*.
- Clean up the import by renaming the appropriate files and deleting any extra Objects that aren't required.

Full Guide of Method 2

- Download [GCFScape](#), and [SourceIO](#). Instructions for installing are under [Installing Programs](#).
- Go to `[game_directory] / tf` and open the file called `tf2_misc_dir.vpk`. It should open through GCFScape.
- This next step will add 2.5 GB to your game folder size, so make sure you're not low on space. Extract the Models folder into `[game_directory] / tf`. Don't try to Drag and Drop as it's extremely laggy and buggy. Right click the folder and click *Extract* so you may extract it. Once done, close GCFScape.
- In Blender, go into *File > Import > Source Engine Assets > Source Model (.mdl)* (Make sure SourceIO is installed).
- Go to `[game_directory] / tf / models / player`. Here you'll find a bunch of files that have the names of the TF2 mercenaries, such as `heavy.mdl` or `spy_animations.mdl`, and so on. Only focus on the one that doesn't have animations in the name, as the other files are for the [Animations](#) section.
- For the class you want to import, click the `(class).mdl`. If you want the imported model to be compatible with taunts or animations (the process of which is explained further down the page), then make sure to set the *World scale* to 1.
- If everything was done right, you should now have the model in Blender with a fully working rig and textures. Make sure to use Material Preview to confirm that the textures are functional.

Note:

This process is identical to that used in [Method 2 \(Alternate\)](#) of the [Individual Props](#) Section, but needs less files to be extracted. As for why, we don't know. SourceIO is very mysterious.

If you used TF2's in-game files, then inside GCFScape when you're extracting the files from `tf2_misc_dir.vpk`, you might have noticed that similar files were also under `models / player / hwm`. The difference between these files and the ones inside `models / player` is only of the mouth supposedly having HWM properties. HWM, or HardWare Morph System, is used by VALVe for facial reflexes and stuff. But according to Hisanimations, they aren't used in TF2, despite their files being present. Whether you use files under `models / player` or `models / player`

/ hwm, won't matter. Other than the mouth, both have the exact same mesh and their quality will be the same. | Again, as mentioned earlier, if you want better quality models, you need to get the files from `tf_movies` from SFM, or just use `characterandrig_method1` for the highest quality models and ease of use.

Animations

Regardless of what method you use to import the TF2 characters and their appropriate rigs, be it the Hisanimations port, or the TF2 in game models, or the SFM models, all use the same method for applying in-game animations (But for the Hisanimations port, you do have to make sure you get the one that's compatible with taunts. That one is available under the `#community-ports` channel of the [TF2 Blender Discord server](#)). There's no other method hence only one method is listed.

This is a bit long and tedious so, make sure to follow every step carefully, but at least you won't have to do these animations yourself from scratch. The method works.

Note:

Not all animations from TF2 can be imported with ease. It depends on which specific animation you want to import. Some animations in TF2 are additive, instead of independent, meaning that you need a base animation and the new animation adds on top of it. For example, to bring in the animation of shooting the shotgun, you first need to have the idle animation of that shotgun brought in.

This is possible in SFM. However, in Blender, a script is required. It's currently being developed by Hisanimations and not ready right now.

Method

Summary of Method

- Download GCFSScape, Blender Source Tools, and Crowbar. Install Blender Source Tools as an Add-on into Blender (This step is only necessary for first time use).
- Using GCFSScape, extract the necessary class files from `tf2_misc_dir.vpk` into a folder of your choice (This step is only necessary for first time use).
- Open the appropriate `.MDL` file in Crowbar and Decompile it into another folder.
- Use Blender Source Tools to import the `.QC` file.
- Remove or hide any unnecessary objects such as the hitbox or extra LOD models.

Full Guide of Method

- Download [GCFSScape](#), [Crowbar](#), and [Blender Source Tools](#). Instructions for installing are under *Installing Programs*.
- Go to `[game_directory] / tf` and open the file called `tf2_misc_dir.vpk`. It should open through GCFSScape.
- From GCFSScape, extract the `models` folder to `[game_directory] / tf`. If you've already done this step from previous guides, there's no need to do it again. Otherwise, make sure you have space, as this step will add 2.5 GB to your TF2 folder.

- Close GCFScape. Go to the folder you just extracted, which is `models`, and go to the `player` folder. Copy `(class)_animations.mdl` to another location, preferably a new folder. This is the file that holds almost all animation data for that specific class.
- Repeat the process for the `.MDL` present in `models / workshop / player / animations`. Just in case the specific animation can't be found in that first `.MDL` file, we'll get the remaining ones from here too.
- Open Crowbar, and go to the *Decompile* tab. For the MDL file, select the `.MDL` from the files you just extracted through GCFScape.
- For the Output Folder, make a new folder or choose an existing one to Decompile to.
- You don't need to change any settings, click *Decompile* in the bottom left. If done right, the folder should have a very large amount of `.smd` or `.dmx` files.
- Finally, In Blender, click on the specific skeleton you want to apply an animation to (You do have to import the Character first. You can't just bring the animation into an empty scene).
- After that, go into *File > Import > Source Engine (.smd, .vta, .dmx, .qc)*.
- Go to the folder where Crowbar Decomplied the files. In there you should find multiple files, all with a lot of names. These files should all have the extension of `.dmx` or `.smd`. Find the one that you're after, and import it. The names might be familiar, so you do have to do some searching, such as `taunt_russian` for Kazotsky Kick.
- If everything was done right, the Timeline in Blender should adjust itself and by pressing play, the animation should be visible.

Note:

In some cases, you may find two identically named files, one name starting with `taunt_` and the other name starting with `layer_taunt_`. These are different files. As of writing, it is uncertain which is the one to use but, if one file doesn't give the wanted results, try the other. Also, not every animation is guaranteed to work, even if it's not an Additive one.

Animations are designed to be played back at 30fps or 24fps. You can use the NLA Editor to change the speed of the animation.

Weapons and Cosmetics

[Hisanimations](#) made a [video](#) explaining how to use his TF2 collection with every TF2 Weapon, Cosmetic, and Prop. Watch the video and follow the instructions.

Sound effects & Voice lines

Download [GCFScape](#). With GCFScape, open `tf2_sound_vo_english_dir.vpk` for voice lines, and `tf2_sound_misc.vpk` for sound effects, both located in `[game_directory] / tf`, to search for whatever you're after. You can extract it then browse it or just go through it directly in GCFScape.

Replays (demos)

Automatic conversion with AutoAGR

This guide contains instructions for setting up and using AutoAGR. It is based on [this video](#) by Hisanimations

Setup

The following add-ons are required:

- [afx-blender-scripts](#)
- [Blender Source Tools](#)
- [The TF2 Trifecta](#)

Software needed:

- [Team Fortress 2](#)
- [HLAE](#)
- [GCFSScape](#)
- [Crowbar](#)
- [Blender](#)

You can download AutoAGR in the *# announcements* channel in the [TF2 Blender Discord server](#)

Steps

Create a folder structure that resembles this one. The GameplayModels folder's name is arbitrary.

```
GameplayModels/
├─ models/
├─ models2/
│  └─ workshop/
```

Open [game_directory] / tf / tf2_misc_dir.vpk with GCFSScape.

Click on root / models in the left panel.

Open the models2 folder in a file explorer.

From GCFSScape, drag the weapons and player folders into the models2 folder.

Open the models2 / workshop folder.

Double click on workshop in GCFSScape.

Drag the weapons folder from GCFSScape to the models2 / workshop folder.

Your folder structure should look like this once you've done all this:

```
GameplayModels/
├─ models/
├─ models2/
│  └─ player/
│     └─ weapons/
```

(continues on next page)

(continued from previous page)

```

| | workshop/
| | weapons/

```

Open crowbar and click on *Decompile*

Enter the full path for your `models2/` folder under the MDL Input field. Change the *MDL Input* dropdown to **Folder** and subfolders

Set the output folder to the full path for the `models/` folder

Under the Options Section, enable **Folder for each model** and click *decompile*

Delete `models2/` when the process is finished

Danger: Do **NOT** join a multiplayer game with HLAE or AutoAGR. This will VAC ban your Steam account.

Run AutoAGR

Set AutoAGR to this recommended configuration:

```

FRAMERATE: 30 / 24

RECORD PLAYER CAMERA: YES

RECORD OTHER PLAYERS: NO

RECORD WEAPONS: YES

RECORD PROJECTILES: YES

RECORD FIRST PERSON MODELS: YES

EXIT GAME AFTER PLAYBACK FINISHES: NO

STARTING TICK: 1

```

Set the paths for the input `.dem` file, the `.agr` output file, the `[game_directory] / tf` folder, and the folder containing HLAE.

Click on the “write” button, then the “play” button.

Once the entire demo has been played, TF2 should go back to the home screen. Type “`mirv_agr stop`” and close the game.

Click on the Output properties panel and set the output frame rate to the one you chose in AutoAGR

Click on *File > Import > HLAE afxGameRecord (.agr)*

Set the import settings to this recommended configuration:

```

Asset path: folder containing the "models" folder.

Add interpolated key frames: No

Scale: 0.1

```

(continues on next page)

(continued from previous page)

Scale invisible to zero: No

Skip Physic, LOD **and** Shared_Player_Skeleton meshes: YesSkip Stattrack **and** Stickers: Yes

Bones (skeleton) only: Yes

Model instancing: Yes

Spawn an FK rig using the TF2 Trifecta and use Bonemerge to bind it to a rig in the scene.

If the eyes are white, select the empties in the eyes and set their Object constraint influence to 0

2.3 Team Fortress 2 Workshop

The TF2 workshop is slightly complicated. Unlike other Workshops, such as that of Garry's Mod or Source Filmmaker, only Maps can be Subscribed to. Other Workshop uploads are only for displaying pictures or videos, and for VALVe to view.

The good news is, with an official VALVe program called SteamCMD, you can get some TF2 workshop items.

The bad news is, only the items that have actually been Accepted can be obtained. All maps, and Accepted items, are the only ones that can be downloaded. Otherwise, your best chance for obtaining the model would be to contact their original creators.

If you know how to use SteamCMD then this shouldn't be a problem. Otherwise, we recommend [This Youtube Video](#) (*Downloading Steam Workshop Files - 2022 edition* by Open3DLab) that explains how to use it. Maps work completely fine and you can Subscribe to them. Everything else is where it gets somewhat arduous.

2.3.1 Contents

Maps and included props

The process of importing TF2 workshop maps is **exactly** the same as that of *Team Fortress 2 Vanilla's Maps and included props* section. There are two methods to do so, both are explained there. The only thing to know about, is that if a Map is obtained by Subscribing to it from the TF2 Workshop on Steam, it will be saved to Steamapps / workshop / content / 440. There will be multiple folders in this location, one of them holds the map. You'll have to do a bit of searching for it, but the .BSP file has a proper name. From this location, copy the .BSP file into [game_directory] / tf / maps. Then the process to import it into Blender is exactly the same, and instructions are provided on *Team Fortress 2 Vanilla's Maps and included props* section.

If instead you obtained the map file by a download on a mod site such as Gamebanana or through downloading it from a workshop downloader such as the one built in Crowbar, the one available [Online](#), or through SteamCMD, then just move it from wherever you downloaded, to [game_directory] / tf / maps. After that, the process is exactly the same, and you should follow the steps listed on the *Team Fortress 2 Vanilla* section.

2.4 Source Filmmaker

Note: If you want to make sure *everything* works, be it maps, models, etc. then you'll want to make sure you have all the DLCs of SFM installed. First, launch Steam. Then go to *Library* and right click on Source Filmmaker. Click *Properties*, and go to *DLC*. Check the ones you need, and exit this menu. Launch SFM and create a session, it'll prompt you for the downloads. It's gonna take a while as the DLCs can go up to 7 GB in size individually, but you only have to download it once.

Throughout these pages, you'll find clickable text where a link is provided. Just click on it or right click and 'Open in new tab' to see the contents.

It's recommended for SFM to be installed for the best experience. You can also obtain the files using Crowbar's downloader, and most should work fine using this, but read the description of the specific map. If it says it needs other things like Prop Packs or to have DLC's, then downloading SFM is your best option.

This entire page is written with the intent that the reader will use it for Workshop and Mods, as SFM Vanilla offers assets that are already fully available in other Source Titles, such as in Team Fortress 2. It doesn't have anything extra in Vanilla except for files present under `tf_movies` as Character Models, and the maps called `stage` and `stage_big`. These can be obtained with the methods shown as well, just with different directories.

Note: Unlike Team Fortress 2 and Garry's Mod, all the files and folders are pretty much already prepared and you don't need to do anything extra, such as turning the Add-on file into regular folders or using GCFscape to extract from .VPK files. So the steps are easier here. However, if using Workshop Add-ons or Mods that are specifically **maps**, the only extra step necessary is to set up Plumber. Here's how to set it up.

For importing maps, Plumber won't autodetect the files from SFM like it does with other titles, so some manual work is needed. Open Blender, then go to *Edit > Preferences...* > *Add-ons*, and find Plumber. Expand it. You'll find two boxes, one on top of the other. There might be some Source Titles present in this box, such as Team Fortress 2 or Garry's Mod. If not, don't worry about it, but if it's there then it'll be helpful to view them to see how to set it up. For this box, click the + sign on the right. Label this new entry as Source Filmmaker, or as SFM. That's your choice. In the second box below, you have to manually add the directories of folders present in the Source Filmmaker folder. If you're confused by what that means, you're basically setting up Plumber so it pulls data from these locations and the textures all work. These will be folders with names such as `steamapps / common / SourceFilmmaker / game / tf`, `steamapps / common / SourceFilmmaker / game / hl2`, `steamapps / common / SourceFilmmaker / game / tf_movies`, etc. All folders should be added. You do also have to be careful that Plumber writes the directories with `\\` and **NOT** with `/`. Look at the other titles for reference if you need to. Once this is set up, Plumber should be good to go if you want to import an SFM map.

2.4.1 Contents

Maps and included props

This *Method* uses three tools, all linked in that section.

Unlike Team Fortress 2 and GMod, SFM maps are a bit complicated and pull resources from multiple different folders. SourceIO has not been successful for the importing of maps, but has been fine for everything else. For this reason, the Map Import section only has one method.

This method requires a fully working copy of Source Filmmaker and a recent version of Blender. If you are unable to get this method to work, it is recommended to use the latest version of Blender.

Also make sure to follow the note under *Source Filmmaker* for how to set up Plumber so it actually reads SFM files.

Method

Summary of Method

- Download BSPSource, Plumber, and Blender Source Tools. Install Plumber and Blender Source Tools as Add-ons into Blender (This step is only necessary on first time use).
- Set up an entry in Plumber for it to read all of SFM's folders present under [sfm_directory] / game (This step is only necessary on first time use).
- Locate the map you wish to bring into Blender by going to [sfm_directory] / game / workshop / maps.
- Convert the .BSP file into a .VMF file using BSPSource.
- Import the .VMF file from *File > Import > Plumber > Valve Map Format (.vmf)* in Blender.
- Disable all lights except light_environment, and set Color Profile from *Filmic* to *Standard*.

Full Guide of Method

Note: Always get the most recent version of a program or Add-on linked here.

Blender Source Tools, BSPSource, and Plumber will be used for this method. The steps to install these can be seen at the *Installing Programs* section.

Plumber was originally called io_import_vmf, and it's by the same developers. It's currently in Beta but significantly superior to io_import_vmf and that's why we've linked the page to download that instead.

Convert a .BSP to .VMF

Important: You need to install [Java](#) to run BSPSource.

- Download [BSPSource](#) and extract all files to a folder.
- Open `bspsrc.jar` from this folder.
- There's many options in the program. For now, just click the button for *Add*. From there, go to `[SFM_directory] / game / workshop / maps` and choose the specific .BSP map file you want to convert.
- Go to the *Other* tab in BSPSource, and enable the checkbox labelled *Extract Embedded Files*.
- Once done, just click the *Decompile* button in the bottom right. There's no need to edit the other settings, though you're free to play around if you know what you're doing.
- A file browser will show up for where to put the .VMF file. You can choose any location, but it's best if it's a place you can easily come back to.
- Click on *Decompile*.

Your .VMF file has now been decompiled and is in your output folder. You'll notice another folder in that location with the same name as the .VMF file. We'll use this later. If such a folder doesn't exist, then don't worry about it.

Important: During the time that BSPSource is Decompileing the map, it will show logs of what it's doing. There is an **Errors & Warnings** box visible. If the process worked, then this box should be mostly empty. If [Warning] is shown in this, then it should be fine. If an [Error] is shown in this, however, then it may have failed. You'll have to come back to this step if the map doesn't work later.

Bringing The Map In

You can now import a .VMF file from the *File > Import > Plumber > Valve Map Format (.vmf)* button (Make sure Plumber and Blender Source Tools are installed). Browse to the location you stored your .VMF file which you Decompiled using BSPSource earlier.

- Make sure that in the Import settings, the game is set to Source Filmmaker (If this option is not available to you, then you have to go to the [Source Filmmaker](#) homepage to set up Plumber).
- Set the Scale to 0.1, and the Light Brightness set to 10. This is so the map is compatible with the Hisanimations Characters port and TF2 Collections Port.
- In the folder space underneath the *Game*, type the name of the folder with the same name as the .VMF. So if for example, your map file is called `fnaf1_sfm_d.vmf` then there should also be a folder called `fnaf1_sfm_d`. Write `fnaf1_sfm_d` in that space. If you don't have such a folder, you can skip this step.
- Then click the .VMF file, and click *Import*. That's it, you should have everything.

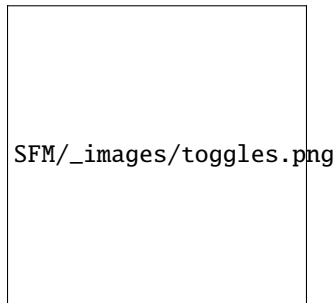
The installation steps are not necessary to do again. It's really just as simple as, Once you set up Plumber so it reads the SFM folders, all you have to do is turn the .BSP map file into .VMF with BSPSource, and bring .VMF into Blender with Plumber.

Go to *Finishing Touches* for advice on clean up and additional useful things to know about within Blender regarding these maps.

If you wish to familiarize yourself with the whole process, or see an overview, a *Summary of Method* is written which you can refer to.

If any problems occur you can ask for help on the [TF2 Blender Discord server](#).

The lighting is going to appear strange because in Eevee (Blender's default render engine) has a maximum of 128 lights. Filter the Outliner (the place where all objects and things in the scene are shown) by lights with the following settings:



See also:

For a full list of Eevee's limitations, you can consult [this page](#) from Blender's official manual.

Finishing Touches

- Use *Material Preview* mode to confirm that all materials are actually fully functional before you do anything else. All textures should be visible and no part of the map should be white.
- Use Eevee if you want a true TF2 look. Cycles will get you very different results.
- There's unfortunately a limit of Eevee which there's no way around. It can only have 128 active lights at once, while a lot of maps in TF2 end up having significantly more than that. Unfortunately the only way around this is to use Cycles, which doesn't have a light limit, but another alternative is to maintain the majority of the look by turning off every light except the one which starts with the name `light_environment`. This is the 'Sun' light and is responsible for nearly all outdoor lighting and shadows present on the map.
- If you want more accurate TF2 colors, go to Color Management, and set the Color Profile from *Filmic* to *Standard*.
- To apply the Skybox, check the *Applying Skybox* section.

Applying Skybox

To apply a Skybox, first find it in the corner, then select all the objects that are near it. Make sure none of the objects are linked to the rest of the map. If they are, then you have to go into Edit mode to separate it.

After that, Shift Click the Empty. This is important because otherwise it won't line up.

Finally, click Object, then Transform VMF 3D sky.

A GIF has been attached to show the process visually.

Individual Models

This section is written as a way to obtain individual models that can be found in the SFM Workshop. This includes characters, weapons, vehicles, cosmetics, the method for importing will all be the same.

Method

Summary of Method

- Download SourceIO. Install SourceIO as an Add-on into Blender.
- Use SourceIO to import .MDL file of choice from the extracted folders.

Full Guide of Method

It's amazing how simple SourceIO has made it. First install [SourceIO](#) as an Add-on into Blender. The steps for this can be found at the [Installing Programs](#) Section. Once done, open Blender, click *File > Import > Source Engine Assets > Source model (.mdl)*, and choose the .MDL file you're after inside the Models folder. The Models folder can be found inside any of the directories within the [sfm_directory] / game folders. It should have textures set up and everything. Workshop items and mods can be found under [sfm_directory] / game / workshop, and the MDL files should be in the Models folder.

Animations

WIP



2.5 Garry's Mod

Throughout this page, you'll find clickable text where a link is provided. Just click on it or right-click and 'Open in new tab' to see the contents.

Depending on the category you'll find multiple methods or only one. Where there are multiple methods (Such as in the [Maps and included props](#) Section), the benefits and drawbacks will be listed. Places with only one method means the other method just wasn't as good/worth it, or there only is one way to do it.

It's best if you have an up-to-date copy of Garry's Mod installed as nearly all of these methods will be relying on obtaining files present from within the game.

Note: The methods shown throughout this page are all effective on both the Vanilla files of Garry's Mod, as well as Mods and Workshop items. If using Workshop Add-ons or Mods, the only extra step necessary is to move the files in question to appropriate locations. A program called GMad can be used for turning .GMA files, which are used as Add-on files and are present under [game_directory] / garrysmod / addons, into regular files. After that, the folders can be copied to [game_directory] / garrysmod, and then the methods can be applied for bringing them in. If instead you obtained the files from a mod download, then you can extract those directly to [game_directory] / garrysmod without having to use GMad.

2.5.1 Turn Addons into Regular Files

To turn a .GMA file into the files you need so that they can be brought into Blender, open two windows of the Game Directory, side by side. In one window, go into [game_directory] / garrysmo d / addons. This is where your .GMA file will be. In the other window, go to [game_directory] / bin. Here, you'll find gmad.exe.

Drag and drop the .GMA file onto gmad.exe. It will create a new folder with the same name as the .GMA file, in the Add-ons folder. This folder will contain everything you need for all the steps on the rest of the page.

2.5.2 Contents

Maps and included props

Note: If you know the Team Fortress 2 methods for bringing maps in, it is exactly the same process here except the location is changed to [game_directory] / garrysmo d / maps instead of [game_directory] / tf / maps. The fully detailed guide has still been listed here, similar to the *Team Fortress 2 Vanilla* section, as some users may not be experienced with Team Fortress 2 and may only be interested in Garry's Mod. This is why you will find the instructions here and on that page to be extremely similar, as they're the same process.

Method 1 (Recommended) uses three tools, all linked in that section. This method is recommended as it makes the maps significantly easier to work with. All imported items are organized into collections and very easy to work with and customize.

Method 2 (Alternate) uses only one addon, called SourceIO. It's a one click solution and way easier than *Method 1 (Recommended)*, but the names of objects becomes messy, and there's extra cleanup required as extra objects such as the map hitbox is also imported. It's closer in looks to Gmod as it uses its own shader, so if you want the true Gmod look, then use this. *Method 1 (Recommended)* also works fine but is better suited for applying your own style or flair to your renders and animations.

Both methods require a fully working copy of Garry's Mod and a recent copy of Blender. If you are unable to get these methods to work, it is recommended to use the latest version of Blender.

Method 1 (Recommended)

Summary of Method 1

- Download BSPSource, Plumber, and Blender Source Tools. Install Plumber and Blender Source Tools as Add-ons into Blender (This step is only necessary on first time use).
- Locate the map you wish to bring into Blender by going to [game_directory] / garrysmo d / maps.
- Convert the .BSP file into a .VMF file using BSPSource.
- Import the .VMF file from *File > Import > Plumber > Valve Map Format (.vmf)* in Blender.
- Disable all lights except light_environment, and set Color Profile from *Filmic* to *Standard*.

Full Guide of Method 1

Note: Always get the most recent version of a program or Add-on linked here.

Blender Source Tools, BSPSource, and Plumber will be used for Method 1. The steps to install these can be seen at the *Installing Programs* section.

Plumber was originally called io_import_vmf, and it's by the same developers. It's currently in Beta but significantly superior to io_import_vmf and that's why we've linked the page to download that instead.

Convert a .BSP to .VMF

Important: You need to install [Java](#) to run BSPSource.

- Download [BSPSource](#) and extract all files to a folder.
- Open `bspsrc.jar` from this folder.
- There's many options in the program. For now, just click the button for *Add*. From there, if you repacked a file using EspionRepacker, go to the folder where the Output file is (the one which is a .BSP file) and select it. Otherwise, go to `[game_directory] / garrysmud / maps` and choose the specific map file you want to convert (these are also .BSP files).
- Go to the *Other* tab in BSPSource, and enable the checkbox labelled *Extract Embedded Files*.
- Once done, just click the *Decompile* button in the bottom right. There's no need to edit the other settings, though you're free to play around if you know what you're doing.
- A file browser will show up for where to put the .VMF file. You can choose any location, but it's best if it's a place you can easily come back to.
- Click on *Decompile*.

Your .VMF file has now been decompiled and is in your output folder. You'll notice another folder in that location with the same name as the .VMF file. We'll use this later. If such a folder doesn't exist, then don't worry about it.

Important: During the time that BSPSource is Decompileing the map, it will show logs of what it's doing. There is an **Errors & Warnings** box visible. If the process worked, then this box should be mostly empty. If [Warning] is shown in this, then it should be fine. If an [Error] is shown in this, however, then it may have failed. You'll have to come back to this step if the map doesn't work later.

Bringing The Map In

You can now import a .VMF file from the *File > Import > Plumber > Valve Map Format (.vmf)* button (Make sure Plumber and Blender Source Tools are installed). Browse to the location you stored your .VMF file which you Decompiled using BSPSource earlier.

- Make sure that in the Import settings, the game is set to Garry's Mod.
- In the folder space underneath the *Game*, type the name of the folder with the same name as the .VMF. If you don't have such a folder, you can skip this step.
- Then click the .VMF file, and click *Import*. That's it, you should have everything.

The installation steps are not necessary to do again. It's really just as simple as, Turn into .VMF with BSPSource, Bring .VMF into Blender.

Go to *Finishing Touches (Both Methods)* for advice on clean up and additional useful things to know about within Blender regarding these maps.

If you wish to familiarize yourself with the whole process, or see an overview, a *Summary of Method 1* is written which you can refer to.

If any problems occur you can ask for help on the [TF2 Blender Discord server](#).

Method 2 (Alternate)

Summary of Method 2

- Download SourceIO and install it as an Add-on into Blender (This step is only necessary for first time use).
- Go to *File > Import > Source Engine Assets > Source Map (.bsp)* in Blender.
- Locate the map you wish to bring into Blender under [game_directory] / garrysmod / maps.
- Once the map is brought in, select everything by pressing A, then Shift Click an Empty.
- Press N under the 3D Viewport to open the sidebar, go to SourceIO, and click *Load Entity*.
- Disable all lights except `light_environment`, and set Color Profile from *Filmic* to *Standard*.

Full Guide of Method 2

SourceIO will be used for *Full Guide of Method 2*. The steps to install these can be seen at the *Installing Programs* section.

Bringing the Map in

Important: Carefully follow these instructions. If you make a mistake, you will have to create a new, blank, project, as this addon directly reads off the .BSP in real time and doesn't allow that file to be changed or edited. There's also a lot of cleanup work to be done so, it's recommended to have a completely empty Scene for this process.

- Go to *File > Import > Source Engine Assets > Source map (.bsp)* (Make sure SourceIO is installed).

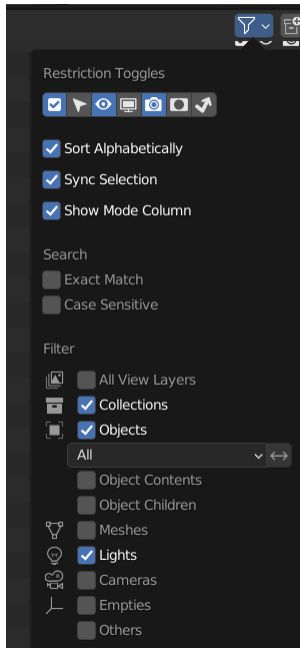
- Select your map of choice. The map **MUST** be in your Gmod game directory. It will be in [game_directory] / garrysmod / maps. You can use the name filter to narrow down the results.

Once loaded in, maps will be quite bare-bones. The lighting will most likely be too dark, and the props aren't there. There are a few things to set up.

- Press **A** to select all objects within the viewport. Then Shift Click on an **Empty**.
- An **Empty** is a placeholder. You'll notice a lot of these in places where Props are supposed to be.
- Everything should be selected, with an **Empty** being highlighted as Orange.
- Hovering over the 3D Viewport, press **N** to open the side panel. There will be a *SourceIO* tab. Click on that to open it.
- Click on *Load Entity*.
- It might take some time so please be patient. If done right, all props should show up without any error messages.

The names of the props will be a huge mess, and every single prop will end up having its own Collection. For this reason, *Method 1 (Recommended)* is recommended, but otherwise, the overall Map and the Props on the map are pretty much the exact same.

The lighting is going to appear strange because in Eevee (Blender's default render engine) has a maximum of 128 lights. Filter the Outliner (the place where all objects and things in the scene are shown) by lights with the following settings:



See also:

For a full list of Eevee's limitations, you can consult [this page](#) from Blender's official manual.

Finishing Touches (Both Methods)

- Use *Material Preview* mode to confirm that all materials are actually fully functional before you do anything else. All textures should be visible and no part of the map should be white.
- Use Eevee if you want a true TF2 look. Cycles will get you very different results.
- There's unfortunately a limit of Eevee which there's no way around. It can only have 128 active lights at once, while a lot of maps in TF2 end up having significantly more than that. Unfortunately the only way around this is to use Cycles, which doesn't have a light limit, but another alternative is to maintain the majority of the look by turning off every light except the one which starts with the name `light_environment`. This is the 'Sun' light and is responsible for nearly all outdoor lighting and shadows present on the map.
- If you want more accurate Garry's Mod colors, go to Color Management, and set the Color Profile from *Filmic* to *Standard*.
- To apply the Skybox, check the [Applying Skybox](#) section.

Applying Skybox

It is necessary for Plumber to be installed for this next part. SourceIO cannot do this.

To apply a Skybox, first find it in the corner, then select all the objects that are near it. Make sure none of the objects are linked to the rest of the map. If they are, then you have to go into Edit mode to separate it.

After that, Shift Click the Empty. This is important because otherwise it won't line up.

Finally, click Object, then Transform VMF 3D sky.

A GIF has been attached to show the process visually.

Individual Props

This section is written as a way to obtain individual props that are universally used in maps stored in Gmod files, such as Barrels. Some maps will have props that aren't used universally, and are exclusive to them. In this case it's best to just import the map, find the prop, and separate it.

Method

Summary of Method

- Download SourceIO. Install SourceIO into Blender (This step is only necessary on first time use).
- (Vanilla) Download GCFscape. Use GCFscape to extract the necessary files from `garrysmo_dir.vpk` into `[game_directory] / garrysmo` (This step is only necessary on first time use).
- (Workshop and Mods) Use GMod to obtain the files of the addon or mod, and move the folders into `[game_directory] / garrysmo` (This step is only necessary for first time use).
- Use SourceIO to import .MDL file of choice from the extracted folders.

Full Guide of Method

The process is rather simple, it only requires a bit of setup, then the importing of the prop should be doable with a few clicks.

- Download [SourceIO](#). Install SourceIO into Blender (Installation guide listed in [Installing Programs](#)).

Steps for Vanilla:

- Download [GCFScape](#) and install it.
- Go to [game_directory] / garrysmod and open the file called garrysmod_dir.vpk. It should open through GCFScape.
- In GCFScape, Right Click the Models folder, click *Extract*, and Extract it to [game_directory] / garrysmod. Don't drag and drop as it is laggy and can bug out.
- Then in GCFScape, Extract the folder called Materials to [game_directory] / garrysmod as well.

Steps for Workshop and Mods:

- (Addon) Use Gmad (Instructions for how to use it are at the [Turn Addons into Regular Files](#) Section) to obtain the files from the .GMA file. Open the newly created folder, and move all the folders inside to [game_directory] / garrysmod.
- (Mods) If it is a .GMA file, use Gmad to obtain the folders. If it is regular files, extra work isn't required. Just move the folders into [game_directory] / garrysmod.

All of that was for setting things up. Once that's completed, all you have to do for bringing a Model in is to open Blender, click *File > Import > Source Engine Assets > Source model (.mdl)*, and choose the .MDL file you're after inside the Models folder. It should have textures set up and everything. The above steps don't have to be repeated.

Character and Rig

Follow the initial steps for setting up [Individual Props](#).

The .MDL files are all present under [game_directory] / garrysmod / models. Just use SourceIO to bring in the one you want. Click *File > Import > Source Engine Assets > Source model (.mdl)*, and choose the .MDL file you're after. If you import a character, they should have a rig and textures set up.

Animations

WIP

Weapons and Cosmetics

Follow the initial steps for setting up *Individual Props*.

The .MDL files are all present under [game_directory] / garrysmo~~d~~ / models. Just use SourceIO to bring in the one you want. Click *File > Import > Source Engine Assets > Source model (.mdl)*, and choose the .MDL file you're after. The weapon should have a rig and textures set up.

Sound effects & Voice lines

- (Vanilla) Download [GCFSScape](#). Use it to open garrysmo~~d~~_dir.vpk and browse the Sound folder. Extract the files you need.
- (Workshop and Mods) Go through the files you need.