

P2PP V4.0

User Manual

DISCLAIMER

P2PP is a post processing tool that relies on patterns in the gcode generated by PrusaSlicer in order to insert the required GCODE for the Palette 2 operation. Due to the vast amount of settings in PrusaSlicer it CANNOT be guaranteed that all combinations of settings will produce output that print as intended. P2PP and its developer cannot be held responsible for any damage occurring from using this tool. Use at your own risk

Introduction

P2PP is a post processing script developed to transform PrusaSlicer MMU2S generated G-Code into G-Code that can be used directly with Palette 2 plugin and hardware. It is an addition to the existing tools provided by Mosaic (like Canvas ad Chroma). Besides the functionality available in the existing tools, it offers the advantage of MMU2S specific features that come with the PrusaSlicer. As a backside, it does not allow for self-painting existing monochrome prints like Canvas does.

In addition to the features available in PrusaSlicer:

- Wipe to object
- Wipe to infill
- Variable Layer heights multi-color prints

It also offers new addon features to further reduce the waste of printing time and material

- Support for tower delta (now also partially implemented in PS2.2)
- Support for full tower reduction with newly generated tower with minimal wipe
- Side wipe support, dumping only needed purge at the side of the bed
- Side Wipe support for Big Brain 3D side wipe mechanism

P2PP is an open source, privately driven initiative held up by a group of 3d printing enthusiasts who help resolve issues where they can.

Special thanks to: Tim Brookman , Khalil Nurallah, Casey Eberle, Paul Kramer, Klause Knute for their help in developing, supporting P2PP and maintaining the support on the P2PP FaceBook community page.

Before diving into P2PP

P2PP offers a lot of nice 3d printing support features to for your Palette 2(Pro)(S), but it is important to understand that no tool will ever produce good results when the hardware is not correctly setup and calibrated. So:

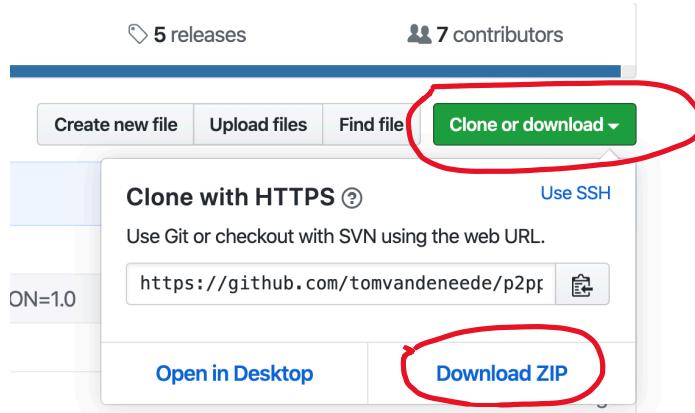
- Make sure your printer is capable of successfully printing single color 3d prints
- Make sure the extruder is properly calibrated... P2 relies heavily on the gcode defined filament consumption to be the same as the actual filament consumption... any deviation may result in color swaps, early or late transitions, color smear ... P2 is a clever device which will try to correct up to a certain level of error but the responsibility with keeping the error low is with the end user
- Follow the P2 calibration process using the Mosaic Mfg suggested prints using the Mosaic supplied slicing tools. Though they can be achieved using P2PP, when new it is a safer bet to use Canvas or Chroma

Start simple... P2PP and PrusaSlicer offer a large interesting features, but before swimming across the Atlantic, try getting on the other side of the pool... no kidding... Try to work from a profile you know works in single color on your printer, only when basic multi-color printing is working, turn to features like the tower delta, side wipe, wipe to infill etc.... it is tempting, but most of troubleshooting will link back to the success of simple things...

Setting up the P2PP Script

Windows

Download the latest version from the [p2pp github page](#) by clicking the **Clone or Download** button on the screen and selecting the **Download ZIP** option.



Unzip the downloaded zip-file to a location of your choice. The location of this file is important for the rest of the installation

Download and Install Python. P2PP is currently compatible with both Python 2.7 and 3.x. <https://www.python.org/downloads/windows/> When installing It is best to check the option to add the python directory to the path.

In the folder where you unpacked the P2PP zip file you will find a windows batch file with the content shown below. You have to edit the path and executable for the python interpreter for your installation of python. If you checked the option to add python to the path, you can just put `PYTHON.EXE %MYPATH%\p2pp.py -i %1`

```
@ECHO OFF
SET MYPATH=%~dp0
REM Under normal circumstances, you should not need to edit this file
unless the path for Python2.7 is incorrect.
REM Avoid editing the "MYPATH" variable as this will automatically
determine the path of the p2pp script.

REM remove "REM" from the line below if you wish to pause before p2pp
executes. This is useful for single extrusion prints.
REM pause

REM Edit the line below if you need to change the python path.
c:\python27\python.exe %MYPATH%\p2pp.py -i %1

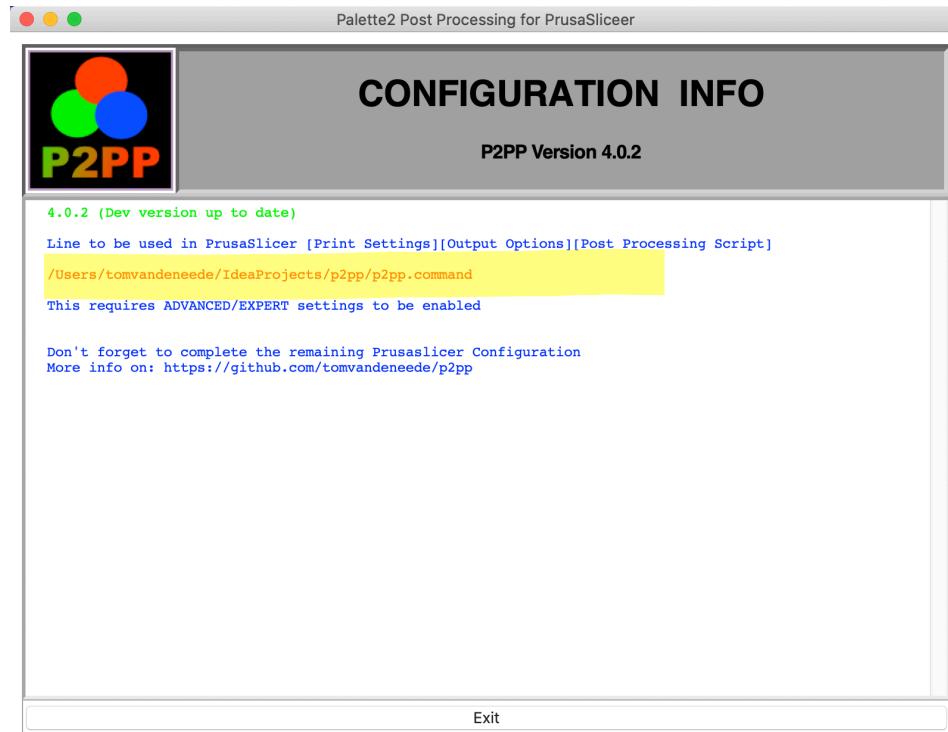
pause
```

If you now double click the P2PP.BAT file in windows explorer, a p2pp window will popup showing some basic information (screen shown here is for a MAC installation)

Leave this window open for now as you will need the line marked in yellow during a next step in the installation.

Congratulations... you have completed the first step of the installation... you can move on to the section on PrusaSlicer configuration

If you did not get the screen you probably mistyped the path to your Python executable... check the steps above before continuing



MAC OSX

Start by downloading the simplified installation zip file for Mac OSX from the [P2PP Github Page](#). Look for a file called p2pp_mac.zip, click the link and download the file to your system

p2pp.bat	- Fix Batch File.	9 months ago
p2pp.command	MAC Simplified installation	5 months ago
p2pp.sh	MAC Simplified installation	5 months ago
p2pp_mac.zip	Updated readme and MAC installation file	5 days ago
version.py	Added DEBUGCOMMAND for debugging purposes, DO NOT USE in regular prints	5 days ago

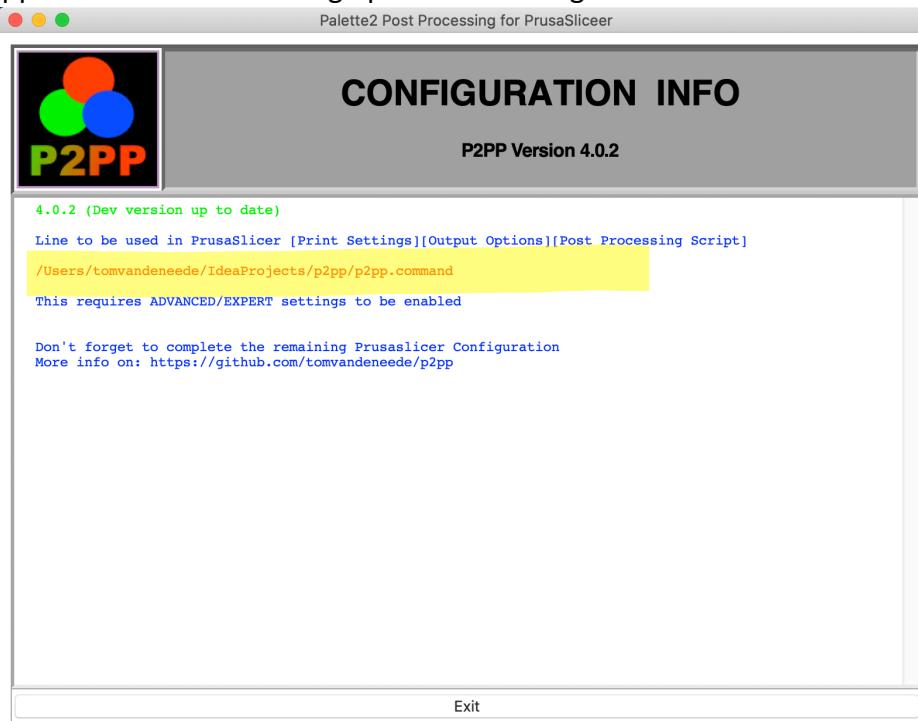
Unarchive the zip file and copy the folder to a location of your choice.

Using Finder, browse into the p2pp folder you just unzipped and you should find a program p2pp.command.

When you double click this command you might get a warning that you cannot execute downloaded internet content:

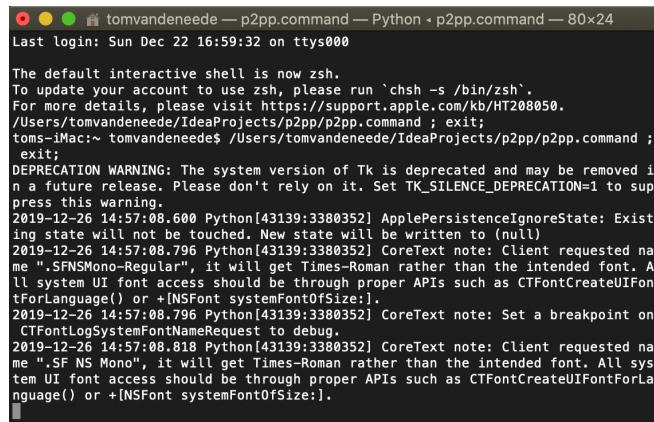
As this is a downloaded application, macOS may prevent you from executing it. You can also grant an exception for a blocked app by clicking the “Open Anyway” button in the General pane of Security & Privacy preferences. This button is available for about an hour after you try to open the app. (You will only need to do this one time after installation) To open this pane, choose **Apple menu > System Preferences, click Security & Privacy**, then click **General**.

Clicking p2pp.command should bring up a the following window.



Leave this window open for now as you will need the line marked in yellow during a next step in the installation.

A second window will also show (terminal window). You can close this window AFTER you have closed the P2PP window. This window will only appear when the p2pp script is called directly. Later when called from PrusaSlicer, there will be no terminal window.



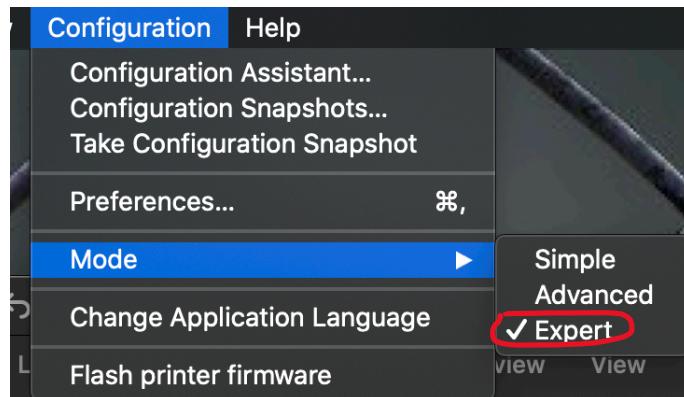
```
tomvandeneede — p2pp.command — Python - p2pp.command — 80x24
Last login: Sun Dec 22 16:59:32 on ttys000
The default interactive shell is now zsh.
To update your account to use zsh, please run `chsh -s /bin/zsh`.
For more details, please visit https://support.apple.com/kb/HT208050.
/Users/tomvandeneede/IdeaProjects/p2pp/p2pp.command ; exit;
toms-iMac:~ tomvandeneede$ /Users/tomvandeneede/IdeaProjects/p2pp/p2pp.command ;
exit;
DEPRECATION WARNING: The system version of Tk is deprecated and may be removed in a future release. Please don't rely on it. Set TK_SILENCE_DEPRECATION=1 to suppress this warning.
2019-12-26 14:57:08.600 Python[43139:3380352] ApplePersistenceIgnoreState: Existing state will not be touched. New state will be written to (null)
2019-12-26 14:57:08.796 Python[43139:3380352] CoreText note: Client requested name ".SFNSMono-Regular", it will get Times-Roman rather than the intended font. All system UI font access should be through proper APIs such as CTFFontCreateUIFontForLanguage() or +[NSFont systemFontOfSize:].
2019-12-26 14:57:08.796 Python[43139:3380352] CoreText note: Set a breakpoint on CTFFontLogSystemFontNameRequest to debug.
2019-12-26 14:57:08.818 Python[43139:3380352] CoreText note: Client requested name ".SF NS Mono", it will get Times-Roman rather than the intended font. All system UI font access should be through proper APIs such as CTFFontCreateUIFontForLanguage() or +[NSFont systemFontOfSize:].
```

Congratulations... you have completed the first step of the installation... you can move on to the section on PrusaSlicer configuration

PrusaSlicer configuration

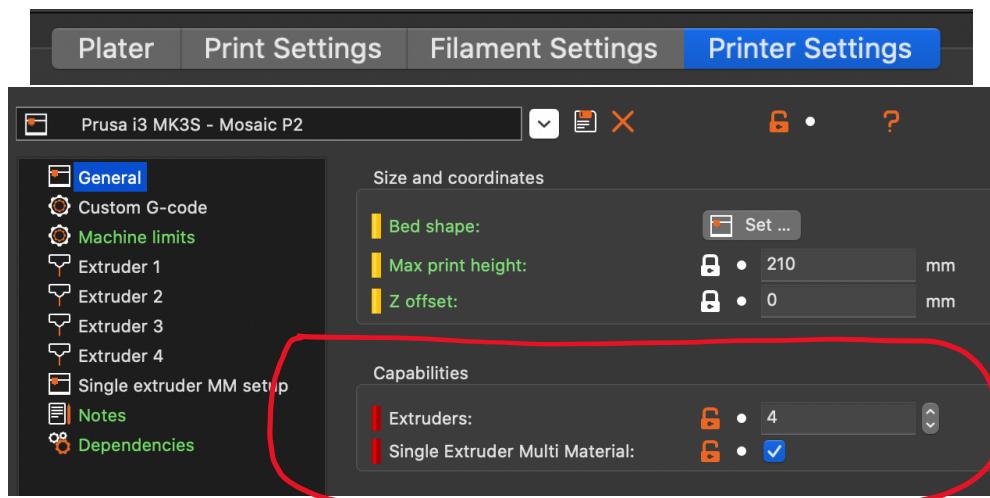
Basic configuration

Put PrusaSlicer in EXPERT mode

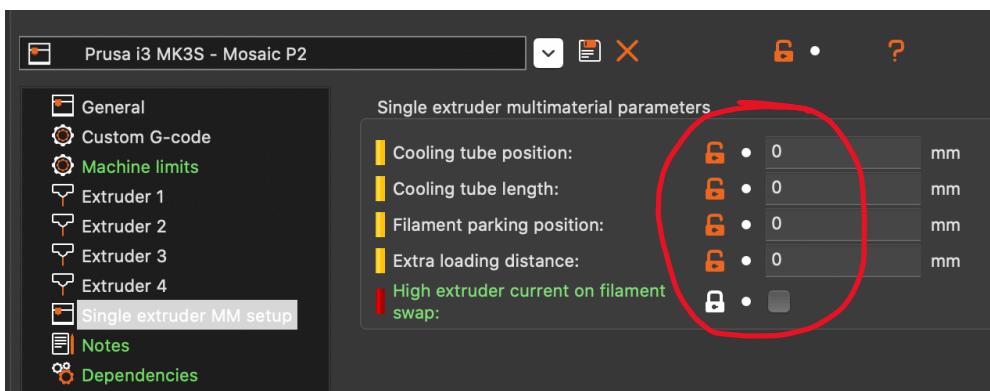


Define the number of extruders (you may have to repeat this for different profiles)

Set the number of extruders to 4 / enable "Single Extruder Multi Material" :



Set all parameters under Single extruder MM setup to zero

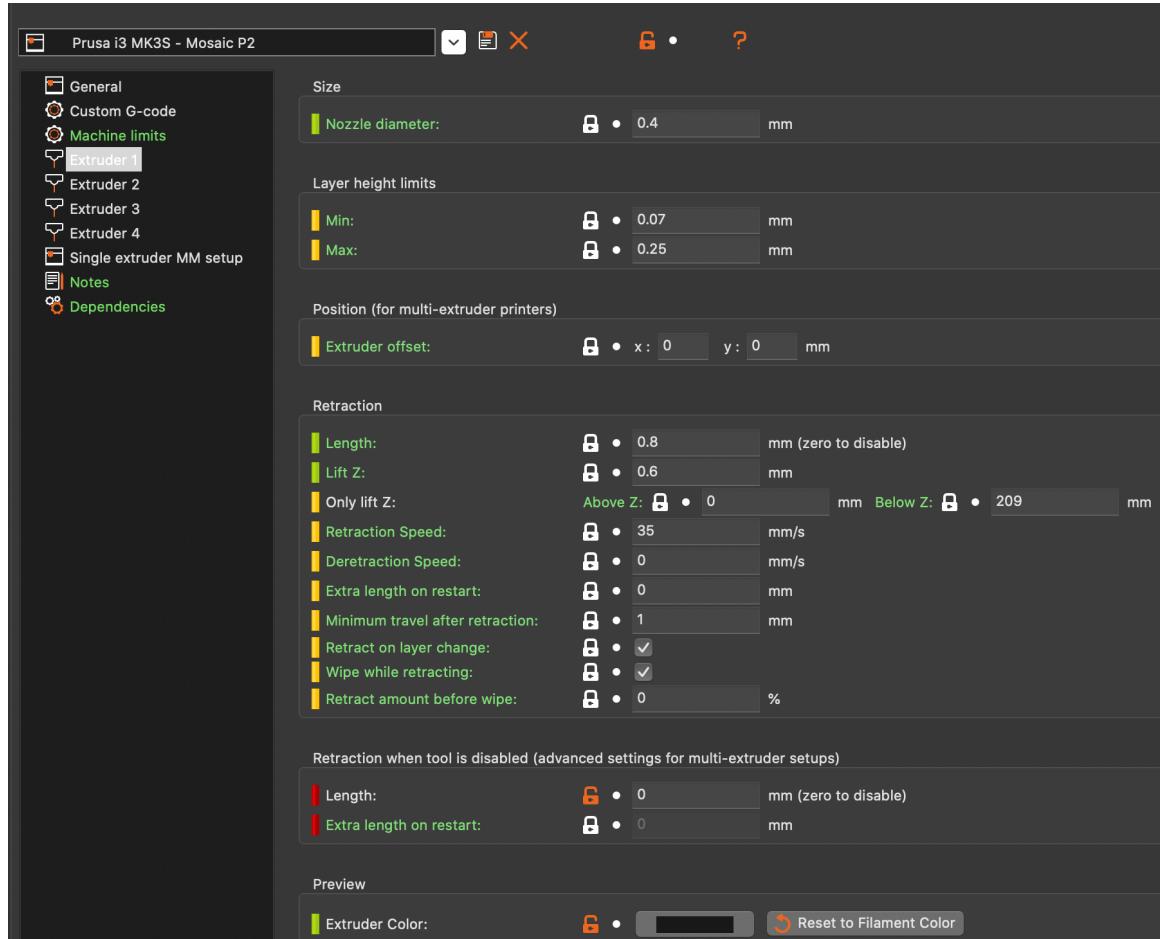


Extruder1/2/3/4 settings:

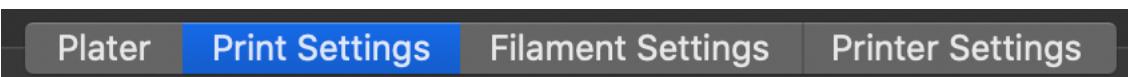
Make sure the settings of each of the 4 extruders are set correctly. For now the Lift Z cannot be 0. If you don't want a lift Z on retraction, put in a low value (like 0.05)

Settings for retraction when the tool is disabled are not used.

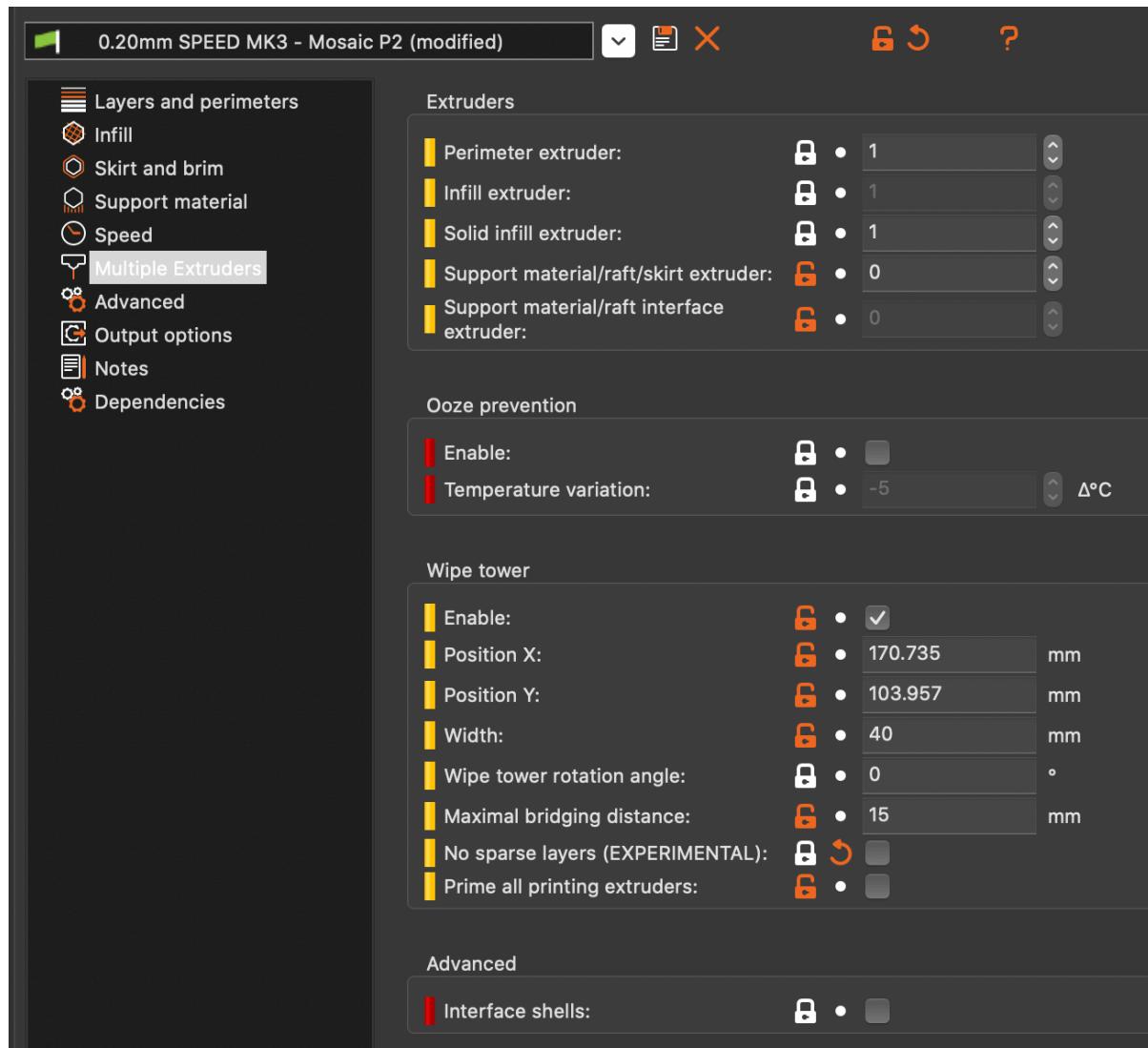
You can also set the extruder color here. This is the color that will be shown on the Palette screen during the loading process. The color can also be set from the plater interface.



Tower setup (you may have to repeat this for different profiles):



Keep the wipe tower rotation angle set to 0 for proper operation. Width can be specified here. The location can be set later by dragging the tower during the splicing process. Ooze prevention does NOT apply because we use a single extruder only.



Filament Setup:



This step should be repeated for EVERY filament type you want to use with your Palette 2

First step is to set all ramming parameters to 0 as well as all of the tool change and wipe tower related settings under the advances tab of your filament.

Note the filament type (here PLA) listed in the dropdown box. This is the reference to the filament that will later be used when creating splicing algorithms to tell Palette 2 how much heat/compression/cooling will be required to splice this filament together to other types of filament.

Prusa PLA Mosaic P2

Filament
Cooling
Advanced
Filament Overrides
Custom G-code
Notes
Dependencies

Filament properties

Filament type: PLA
Soluble material:

Print speed override

Max volumetric speed: 15 mm³/s
First layer volumetric flow rate is maximized when printing external perimeters with a volumetric rate 1.51 mm³/s at filament speed 0.63 mm/s.
Volumetric flow rate is maximized when printing infill with a volumetric rate 16.28 mm³/s at filament speed 6.77 mm/s.
Bridging volumetric flow rate is maximized when printing external perimeters with a volumetric rate 3.58 mm³/s at filament speed 1.49 mm/s.

Wipe tower parameters

Minimal purge on wipe tower: 0 mm³

Toolchange parameters with single extruder MM printers

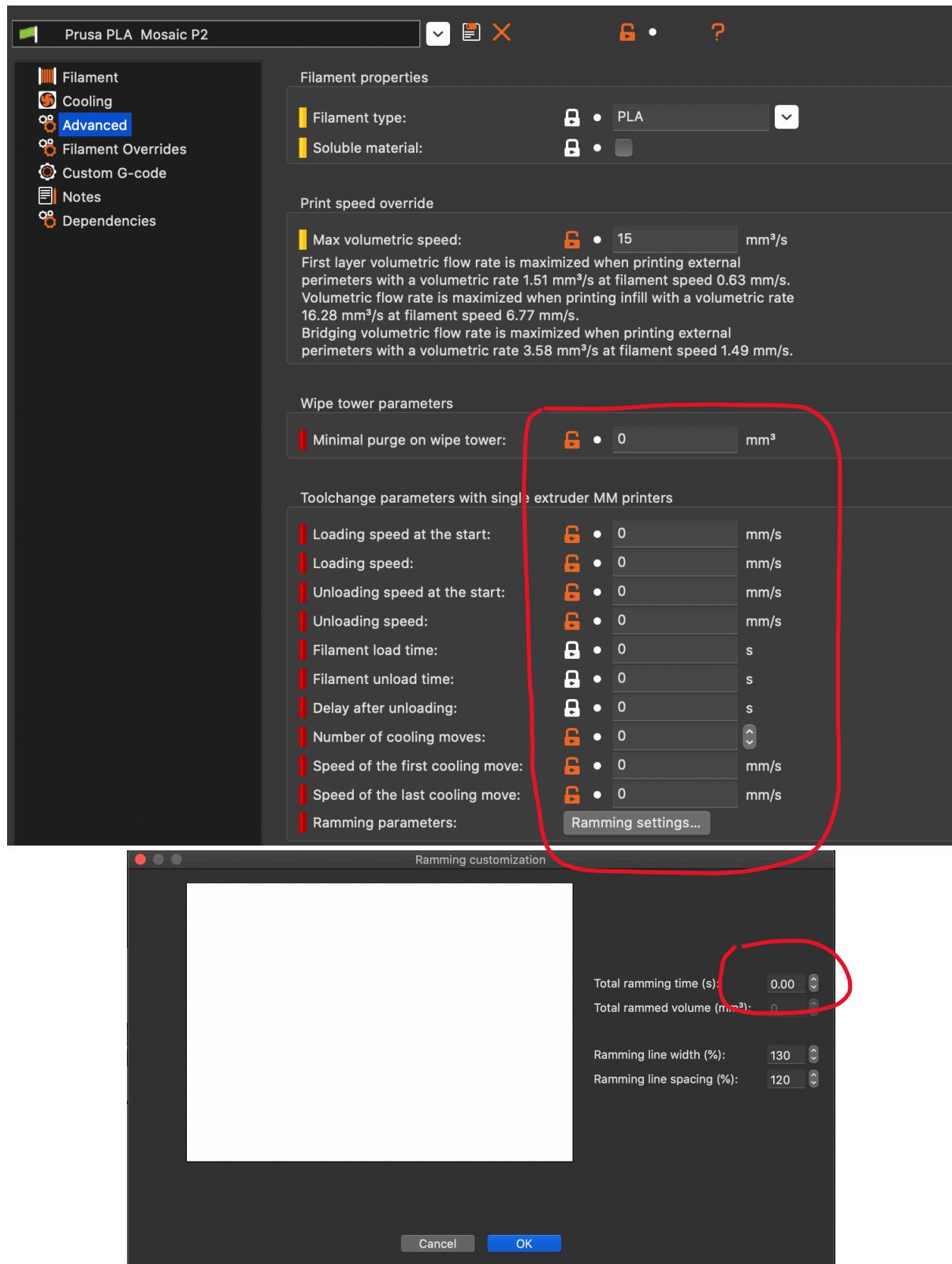
Loading speed at the start: 0 mm/s
Loading speed: 0 mm/s
Unloading speed at the start: 0 mm/s
Unloading speed: 0 mm/s
Filament load time: 0 s
Filament unload time: 0 s
Delay after unloading: 0 s
Number of cooling moves: 0
Speed of the first cooling move: 0 mm/s
Speed of the last cooling move: 0 mm/s
Rammimg parameters...
Rammimg settings...

Rammimg customization

Total rammimg time (s): 0.00
Total rammed volume (mm³): 0

Rammimg line width (%): 130
Rammimg line spacing (%): 120

Cancel OK

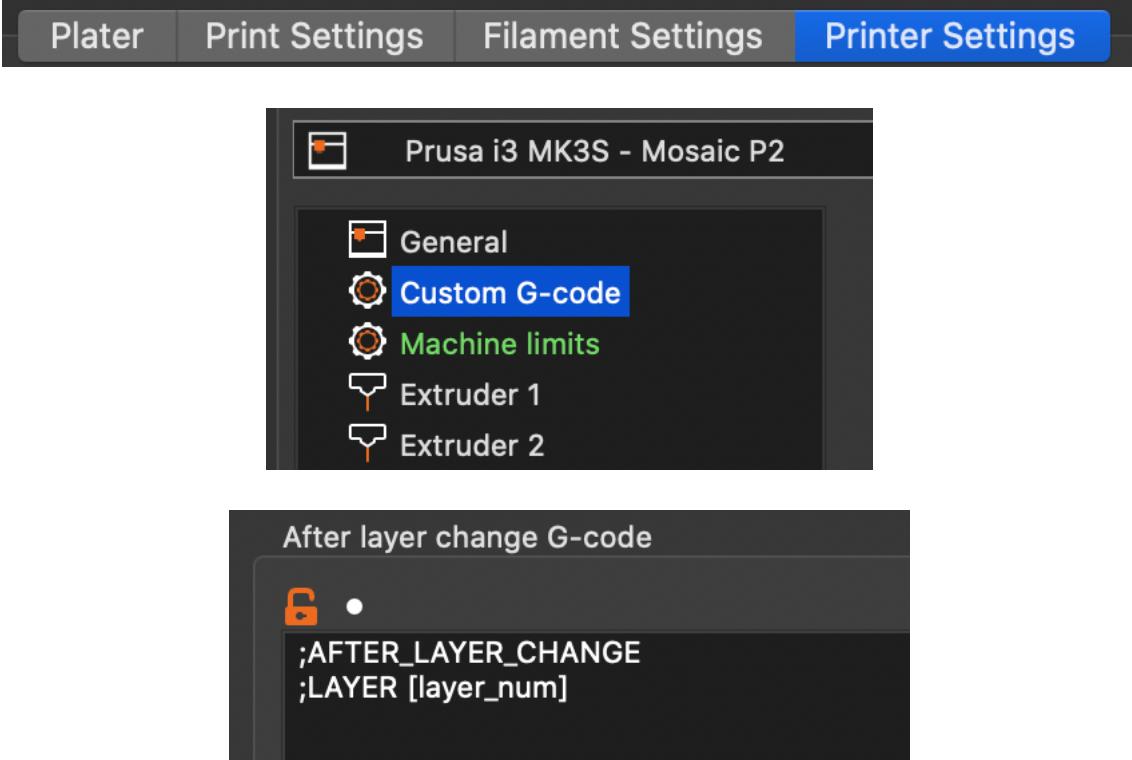


Layer Configuration

For some of the features P2PP will need to know which features are linked to a particular layer. Therefor a very specific layer setup is required to make sure this information is captures correctly

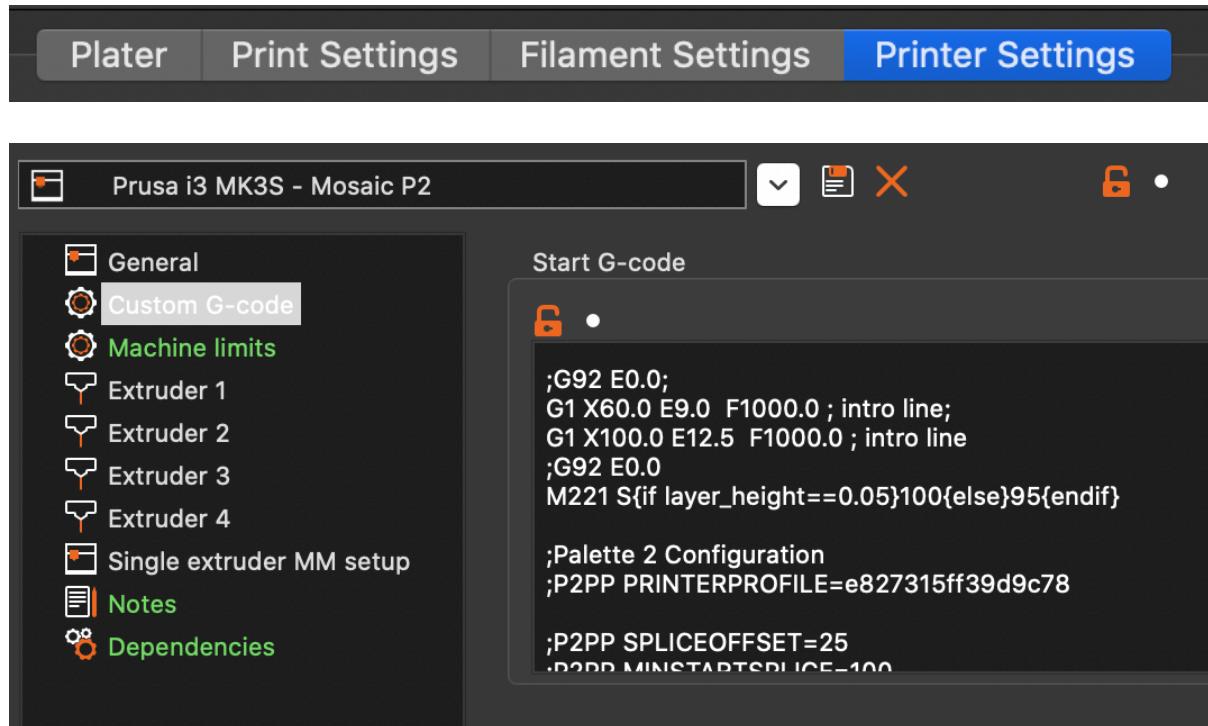
Make sure that the following code appears in the After layer change G-Code (copy this literally):

```
;AFTER_LAYER_CHANGE  
;LAYER [layer_num]
```



Printer Configuration

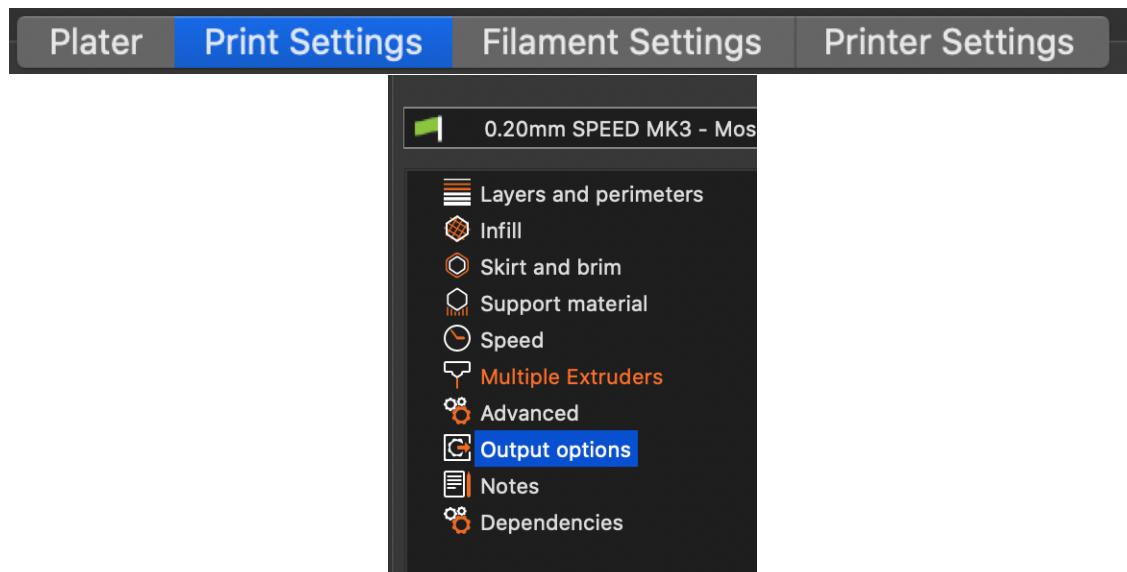
All basic configuration goes into the **Printer Settings → Custom G-Code → Start G-Code**
You can group all P2PP config settings at the end of your normal start g-code.

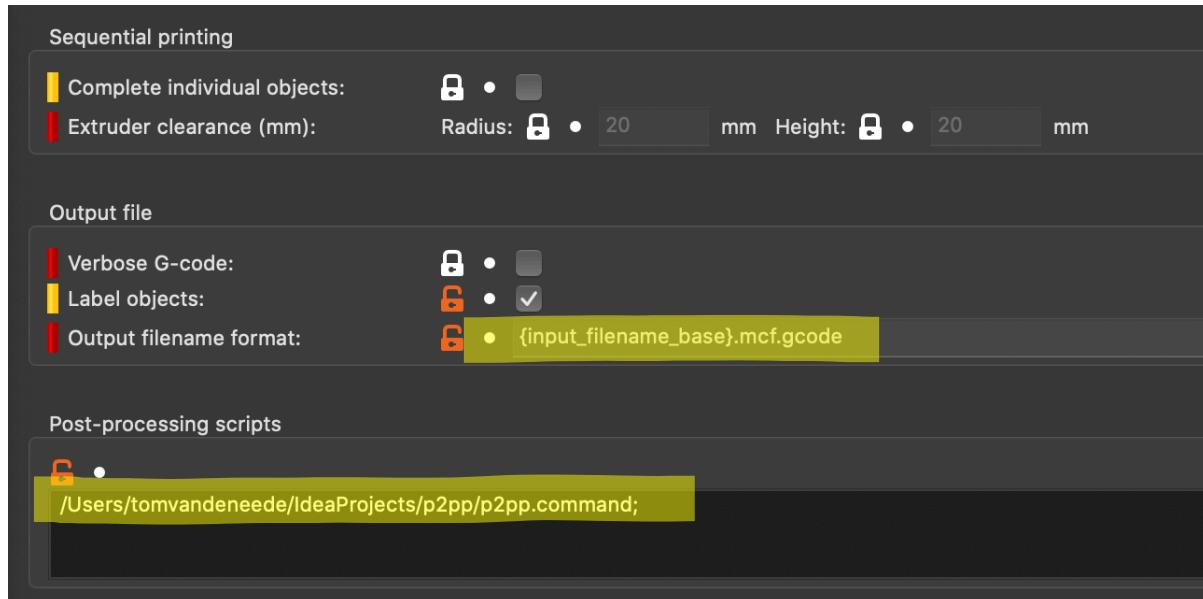


All settings will be prefixed by ;P2PP followed by a parameter name and a setting. Note that all P2PP configuration is considered **CaSe SeNsItIvE!!!**

Script Configuration in PrusaSlicer

Next thing to do is to configure the P2PP script in PrusaSlicer





You have to specify 2 pieces of information here

- File name template. You can use the current template but it may be interesting to add the .mcf.gcode instead of just gcode. This has no functional purpose but it will help you discriminate between P2 and non-P2 prints in the canvas hub overview.
- On the post processing script tab you will need to copy the full path of the script. This information you can copy from the window you ran right after you unzipped P2PP and clicked on the bat/command file

```
4.0.2 (Dev version up to date)
Line to be used in PrusaSlicer [Print Settings][Output Options][Post Processing Script]
/Users/tomvandeneede/IdeaProjects/p2pp/p2pp.command
This requires ADVANCED/EXPERT settings to be enabled

Don't forget to complete the remaining Prusaslicer Configuration
```

Printer Configuration

Printer Profile – PRINTERPROFILE

The printer profile is the unique ID that is linked to a calibration set in your P2. It will determine the loading length, palette 2 specific calibration as well as the long term historic learning factor associated to your printer/palette2 combination. You can make up your own 16-digit hex decimal value or if you are currently using Canvas or Chroma and you want to keep on using the same calibration information you can copy the information from an existing .mcf.gcode file.

In an mcf.gcode file look for a line starting with O22 followed by a D and 16 digits. These 16 digits are your printer ID. The first capital D is NOT part of the printer ID

1	021 D0014
2	022 De827315ff39d9c78
3	023 D0001
4	024 D0000

In the start g-code file should get exactly 1 line with a printer profile configuration

```
;P2PP PRINTERPROFILE=e827315ff39d9c78
```

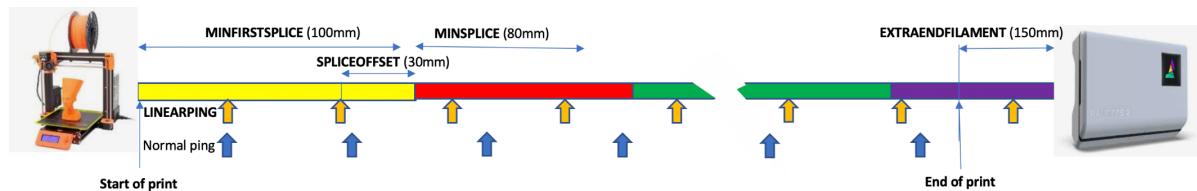
Printer Bed Size and position – BEDORIGINX/BEDORIGINY/BEDSIZEX/BEDSIZEY

The next part of the configuration is de definition of the bed size and origin of the printer. P2PP requires this information to confirm if the purge tower is located ON or OFF the bed as this is used as a trigger for the side wiping mechanism (see later).

P2PP only supports Rectangular beds for now. The default values are set for the Prusa i3 MK3 printers. Nevertheless it is good habit to include them in your setup:

```
;P2PP BEDORIGINX=0
;P2PP BEDORIGINY=0
;P2PP BEDSIZEX=250
;P2PP BEDSIZEY=210
```

Splice Configuration



- SPLICEOFSET
- MINFIRSTSPLICE
- MINSPLICE
- EXTRAENDFILAMENT
- LINEARPINGLENGTH

Splicing Offset - SPLICEOFFSET

Spliceoffset defines the amount of extra filament that is added to the first splice. It will force the color swap further into the purge. As a result, when a color change occurs, the printer will still continue to print in the old color for the amount defined in the spliceoffset parameter. This gives the print a small buffer in case more filament was consumed than foreseen. Making the offset too small may result in early transition. On the other hand but printing in the old color during the purge sequence, you remove that amount from the overall purge so if you make spliceoffset too large you will end up having color bleeding or late color transitions. 34-40 seems to be a good value. The value is specified in millimeter

```
;P2PP SPLICEOFFSET=40
```

Minimal Splice Lengths MINSPLICE/MINSTARTSPLICE

The length of splices that can be generated by the Palette 2 is limited to 100mm for the first splice and 80mm for consecutive splices. The values can be brought down slightly at your own risk. It is good practice to leave them to the default values.

```
;P2PP MINSTARTSPLICE=100  
;P2PP MINSPLICE=80
```

Extra filament at the end of the print - EXTRAENDFILAMENT

Depending on the setup of your printer you may need to add an extra length of filament to make sure the filament is driven by the filament motor until the very end of the print. On a direct drive a value of 100 to 150mm should probably be sufficient, on Bowden printers where the extruder motor is further away from the extruder, longer lengths are needed.

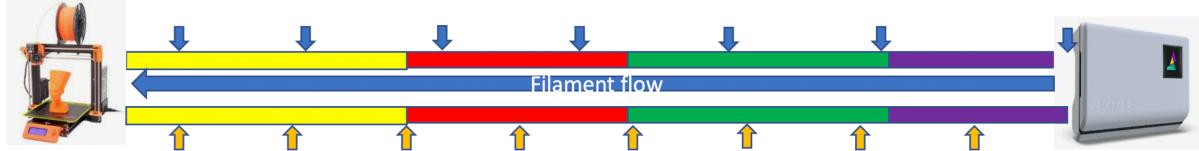
Note: the extra filaments. This parameter is specified in millimeter.

```
;P2PP EXTRAENDFILAMENT=150
```

Linear Ping setting – LINEARPINGLENGTH

Normal ping – distance between the pings increases throughout the print. Each next ping distance is approx. 3% longer than the previous. The first ping occurs after 350mm of printing. This can result in less strict filament consumption in later stages of the print.

Each ping is represented by  in the diagram below



Linear ping – distance between the pings stays constant throughout the entire print. The distance between pings can be defined by the user. The parameter should be specified in millimeter. The default distance is 350mm

Each ping is represented by  in the diagram below

```
; P2PP LINEARPINGLENGTH=350
```

Limiting the purge speed – PURGETOPSPEED

If you are using the PrusaSlicer generated tower code in unprocessed mode or in tower delta mode you may find the purge tower is printed too fast. You can prevent this add the PURGETOPSPEED (in mm/min) to limit the speed at which purge is printed.

Splicing Algorithms

One of the most important settings is the definition of how splices should be generated. Splices are defined by 3 numbers

- Heating
- Compression
- Cooling

A default splice has 3 numbers you can tune the settings for your material. The numbers can be obtained using the standard splicing tuning process as defined by Mosaic Mfg. They represent the heating, compressing and cooling factor.

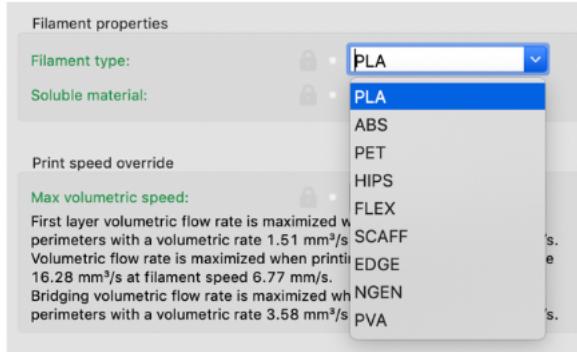
It is good practice to put all of your splicing algorithms into the g-code startup sequence. This will prevent surprises when mixing materials later on.

P2PP supports 2 ways to define your splices:

;P2PP MATERIAL_PLA_PLA_0_0_0

source material destination material heat factor compression factor cooling factor

First way is to describe the heat/compression/cooling factor for a splice between two materials. The name of the material (in this case PLA) is taken from the filament type defined in the filament descriptors within PrusaSlicer:



It is possible to define a default splice setting just in case no appropriate splice setting is defined.

;P2PP MATERIAL_DEFAULT_0_0_0

Used for all material combinations that are not explicitly defined heat factor compression factor cooling factor

This sets the default splice values in case no splicing algorithm is defined for 2 materials used in the print. P2PP will default to this process any time materials are fused that have no specific splice definition.

Special Features

Feature Compatibility Matrix

	ABSOLUTE	TOWER DELTA	FULL PURGE REDUCTION	SIDE WIPE	SIDE WIPE Big Brain
ABSOLUTE	X	X	X	X	X
TOWER DELTA	X	X	FULL PURGE	SIDE WIPE	SIDE WIPE
FULL PURGE REDUCTION	X	FULL PURGE	X	SIDE WIPE	SIDE WIPE
SIDE WIPE	X		SIDE WIPE	SIDE WIPE	SIDE WIPE
SIDE WIPE BIG BRAIN	X		SIDE WIPE	SIDE WIPE	SIDE WIPE

(*) SIDE WIPE is triggered by the position of the purge tower. This position is incompatible with all the other purging modes

Absolute Extrusion

During the gcode generation process in PrusaSlicer, only relative extrusion code is supported. That means that each extrusion command in the file is specifying the length relative to the current position of the extruder. The downside is that rounding error accumulate over time and this may have a negative effect on the power consumption in long prints. By converting the file to absolute extrusions, the error is still there to a lesser extent, but it is also taken into account for the pings so will be part of the Palette 2 correction mechanism.

To enable the extra cycle at the end of the processing to convert the the purge add the following line to your start g-code

```
;P2PP ABSOLUTEEXTRUDER
```

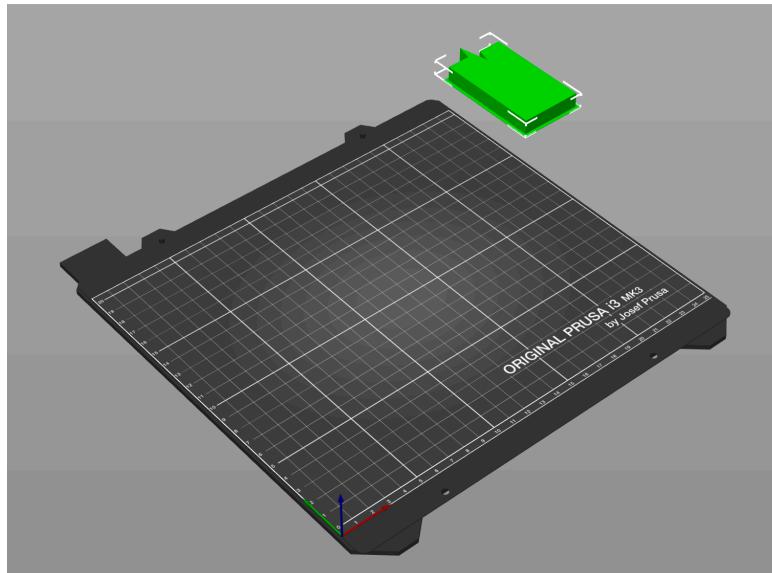
Side Wiping

Side wiping is a mechanism that fully replaces the purge tower but extrusions that happen off-bed and that will not make part of any physically printed structure. Depending on the method used you will either get an extrusion spaghetti or small blobs of plastic.

The advantage of side wipe is that no purge structure needs to be maintained so you just purge the exact amount you need, no more no less.

Downside is that you will end up with “non-structured” plastic blobs or spaghetti... when doing this make sure you have something that catches this purge so it cannot make it to your print. It could easily turn your extruder in a great mass of plastic..

In order to enable side wipe you need to move the purge block fully off the build plate.



Bed Side Purge

This is the first way of doing side purge. It will move the carriage to the side of the bed and extrude spaghetti of the side of the bed while moving back and forward on the Y-Axis.

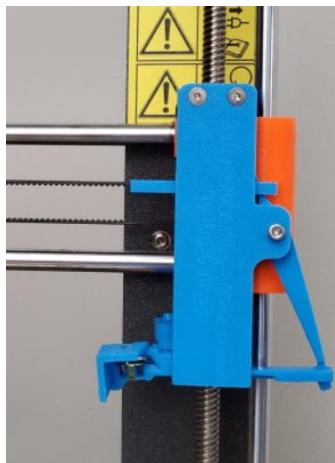
In order to setup this form of side wipe you need

- Setup the X-Position to which the printer should move
- Define the minimal and maximal Y-coordinate between which the printer will move
- Since there is no backpressure on the extruder SIDEWIPECORRECTION allows for a correction of the amount of purge made during this form of side wipe correction.

```
;P2PP SIDEWIPELOC=X254
;P2PP SIDEWIPEMINY=45
;P2PP SIDEWIPEMAXY=195
;P2PP SIDEWIPECORRECTION=1.0
```

Big Brian 3D Purge Mechanism

A second way of doing purge is collecting the purge in a Big Brain 3D mechanism



This device is currently only available for the Prusa MK3/MK3S printers and will require a stock extruder and x-idler. Step files are available when you purchase the device so you can try to adapt to your own printer.

The device comes with its own calibration guide. You will have to take the values from your calibration to complete the setup in P2PP

The following parameters need to be set to activate side wipe. Note that for side wipe to activate the purge block needs to be moved OUTSIDE of the heated bed. The Purge position needs to be taken from the calibration results for your printer.

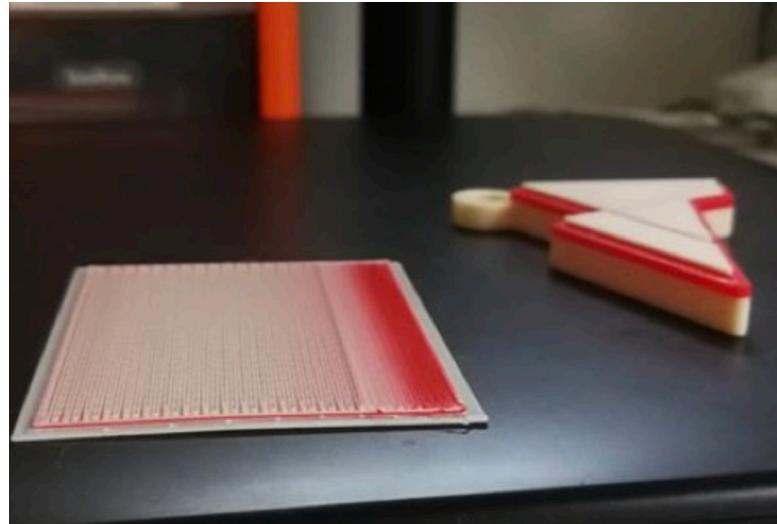
```
;P2PP BIGBRAIN3D_BLOBSIZE=40  
;P2PP BIGBRAIN3D_COOLINGTIME=20  
;P2PP BIGBRAIN3D_PURGEPOSITION=255  
;P2PP BIGBRAIN3D_MOTORPOWER_HIGH=850  
;P2PP BIGBRAIN3D_MOTORPOWER_NORMAL=550  
;P2PP BIGBRAIN3D_ENABLE
```

The other parameters are taken from the manufacturers website. The BLOBSIZE of 40 is the default value and all BLOBS will be 40mm of filament equivalent in size. This means that all purging sequences are rounded to the nearest greater multiple of 40mm

Cooling is set to 20sec, but could be reduced depending on the material used and the environmental situation. The blob needs to be hard enough to be pushed out of the way by the flicker arm

The motor power settings are defaulted to the manufacturers suggestions as well.

Tower Delta



Tower delta is the principle of not printing empty tower sequences that take up time and filament. Instead the tower grows at a slower pace than the print. This creates a risk that the print head or the beam may hit the actual print while printing the purge. In order to minimize the risk it is good practice to group the printed objects in front of the purge and to one side of the bed and put the tower on the diagonal opposite (back side) of the print.

PrusaSlicer has this feature built in as of version 2.2. The major difference between the PrusaSlicer implementation and the P2PP implementation is that in P2PP you can specify the maximum difference between the print and the tower to mitigate the risk of hitting the print.

There is one parameter that is used for both activating and setting the maximum delta of a tower delta print:

```
;P2PP PURGETOWERDELTA=10  
;P2PP WIPEFEEDRATE=3000
```

Adding this parameter allows for the tower to grow 10mm lower than the actual print. If you do not want to limit the difference, either put in a large number (999) or use fullpurgereduction feature described later

To disable the tower delta, put a value of 0

Make sure you have the correct after layer gcode inserted as described above or the feature will NOT work correctly

The Tower delta function is useful on prints where you have lengths of print in a single color. (i.e. layers with no color change). These layers in general cause empty layers to be created in the purge tower which can easily be ignored.

The wipe federate specifies the maximum federate during wiping

Full Purge Reduction

Full Purge Tower reduction is similar to Tower Delta but

- Has no difference in the number of layers
- Will generate a whole new purge tower (will be used for other features in the future)

```
;P2PP FULLPURGEREDUCTION  
;P2PP WIPEFEEDRATE=3000  
;P2PP AUTOADDPURGE
```

Full purge tower reduction will use the same amount of filament as Side Wiping but it will still create a physical purge tower that will need more time to create

The wipe federate specifies the maximum federate during wiping

There is an experimental setting AUTOADDPURGE automatically adds purge length in case a short splice is detected. This is only intended for occasional short splices, if there are many short splices in the print, the purge tower will grow over the print height. This will cause issues.

Accessory Mode [still under development]

Accessory mode prints generate Ping sequences in a separate MAF/MSF file that is put on an SD card and is fed to the Palette 2 in that way. The pings are inserted in the form of delays during the print. As such the print can be executed directly from SD card in the printer without the need of an Octoprint.

P2 Accessory Mode

Parameters need to be taken from calibration information

```
;P2PP ACCESSORYMODE_MAF
```

Palette + Accessory Mode

```
;P2PP ACCESSORYMODE_MAF  
;P2PP P+PPM=<ppm value>  
'P2PP P+LOADINGOFFSET=<loading offset>
```

Non-Print related settings

CHECKVERSION

Adding the checkversion keyword to your stag code will trigger P2PP to download the information about the latest master/dev online versions and warn you if updates are available

```
;P2PP CHECKVERSION
```

CONSOLEWAIT

Setting the CONSOLEWAIT parameter will trigger P2PP to wait for user response even if no warning were issued

```
;P2PP CONSOLEWAIT
```

IGNOREWARNINGS

Ignore warnings will close the window even if there were warnings during the processing

```
;P2PP IGNOREWARNINGS
```

DEBUGTCOMMAND

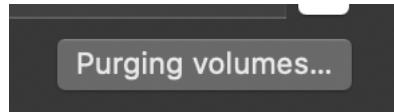
Leave the T-commands untouched in the print so it can be previewed in g-code previewers while maintaining the information on the tools used (the files generated this way will NOT print correctly with the P2)

```
;P2PP DEBUGTCOMMAND
```

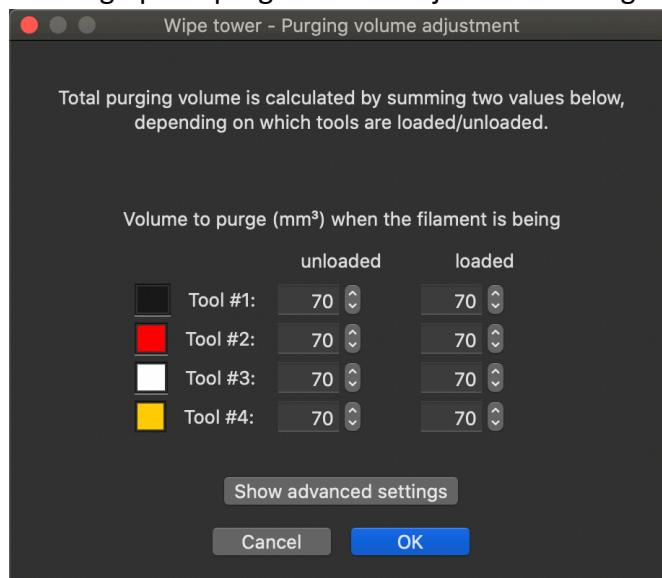
PURGE/WIPE LENGTH

The quality of your color transitions will greatly depend on the purge or wipe length defined in PrusaSlicer.

When in platter vies mode in a multicolor setup, you will notice a Purging Volumes button right above the object tree windows



Clicking this button will bring up the purge volume adjustment dialog.



The values in this diagram are defined in mm^3 (volume of filament), not in mm (length of filament). 70mm^3 of filament is about $70/2.4 \text{ mm}$ of filament.

The length of the purge will depend on the mature of the filament used. The effective length of purge generated is the sum of unload pf the old value and the load value for the new filament.

It is good practice to keep the minimal purge around 180mm so you avoid short splice specifically when using layers with only tiny details in a specific color.

As a rule of thumb:

- Strong colors (e.g. red/black) will need a low load value / high unload value
- Weak colors (e.g. white transparent) will need high load / low unload values

So going from black to white will be unload black + load white = high value

The other way around form white to black, will add low unload + low load = low value

Running P2PP

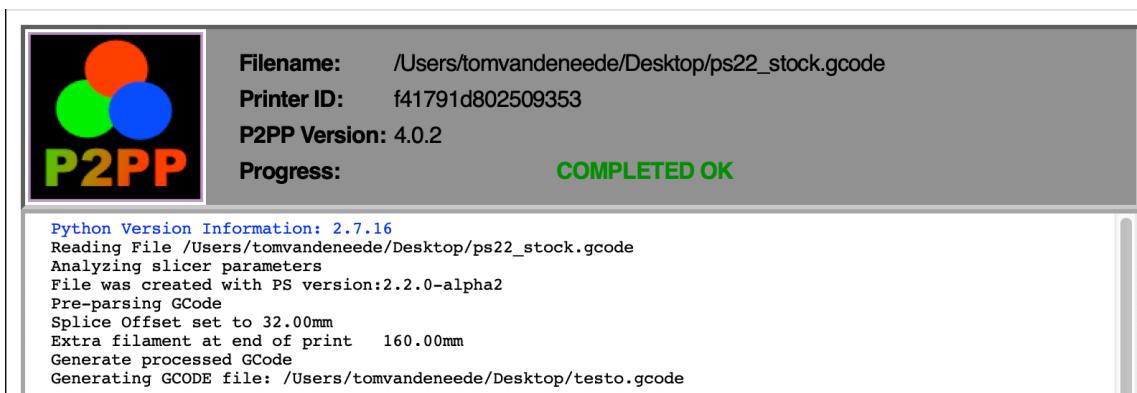
Once you have all parameters entered you can load a multi-color file in PrusaSlicer and click the export button/send to printer button... if all goes well you will get a pop-up window from P2PP

If there were warnings (like in the example below), the window will remain open (as long as the window is open PrusaSlicer will not function normally and should not be used)

You can review the errors and see what is needed to correct them. Clicking OK will return you to PrusaSlicer. If uploading was selected, the print will be uploaded to the printer.



If the processing went through without issue the window will close automatically without waiting and the PrusaSlicer will take control from there. If you have the consolewait parameter set you will get a similar window with a header confirming the correct processing



In either case the output file will contain the processed information from P2PP, even if errors have been detected.

Troubleshooting

Before reporting an issue:

- Make sure you have the latest version of P2PP running. We only support the latest master and latest development version of P2PP. Your bugs may have been solved in later versions of the tool.

If after reading all of the documentation, you should still have issue to make successful prints using P2PP, you can turn to the [P2PP Community Help](#) page on Facebook.

If you cannot find the information you are looking for in the existing topics, you can post your own question. Be thorough in your descriptions, people are there to help and the more you give them to work with the better chance you will have to get your specific issue resolved...

When reporting issues you will often be asked to provide supporting material:

- Processed GCODE file (output of P2PP script) so we can see what code is generated and if it shows any sign that could link the problem to the code
- Unprocessed GCODE file (output of PrusaSlicer without the P2PP processing). The easiest way to achieve this is to add an extra character in the output
- OctoPrint log file of the print that showed the problem

.