slic3r.org »

Slic3r Manual

Command Line Usage

Slic3r can be used as a command line tool. It provides you with great flexibility so that you can perform operations in batch or as part of more complex workflows.

Actions

The general syntax is:

```
slic3r [ ACTION ] [ OPTIONS ] [ model1.stl
```

where ACTION can be one (or more) of the following:

- `--help`: displays the inline help
- `--export-gcode` (shortcut: `--gcode` or `-g`): slices the given model(s) and exports G-code toolpaths
- `--export-stl`: exports the given model(s) as STL
- `--export-amf`: exports the given model(s) as AMF
- `--export-3mf`: exports the given model(s) as 3MF
- `--export-obj`: exports the given model(s) as OBJ
- `--export-pov`: exports the given model(s) as a POV-Ray definition
- `--export-svg`: slices the given model(s) and exports solid slices as SVG files
- `--export-sla-svg` (shortcut: `--sla`): slices the given model(s) and exports slices for SLA printers as SVG files (with infill patterns, raft support...)
- `--info`: outputs information about the model(s)
- `--save FILE`: saves configuration to the given file

If called with no ACTION, the graphical interface will be launched and the supplied models (if any) will be loaded in the plater. If multiple actions are be specified, they will be

executed in the given order. If multiple models are supplied, the requested action will be performed separately for each one.

Model transform

The following options will affect how the model(s) are transformed before performing the requested action:

- `--merge` (shortcut: `-m`): the given models will be arranged and then merged into a single one, thus the action will be performed once
  - if the `--dont-arrange` option is supplied, models will be merged at their original coordinates
- `--center X,Y`: centers the given model(s) around the given point and also aligns them to *z = 0*
  - as an alternative, `--align-xy X,Y` aligns the model(s) to the given point
  - if none of those are specified, `--export-gcode` will center the model around the centroid point of the configured print bed (unless the `--dont-arrange` option is supplied: original coordinates will be used in that case)
  - `--export-svg` and `--export-sla-svg` will align to *z = 0* anyway
- `--scale FACTOR`: the given model(s) will be scaled by the given factor or percentage (the % character determines how the number is parsed)
- `--scale-to-fit X,Y,Z`: the given model(s) will be scaled to fit the given volume expressed in millimeters
- `--rotate ANGLE`, `--rotate-x ANGLE`, `--rotate-y ANGLE`: rotates the given model around the Z, X, Y axes by the given degrees
- `--duplicate NUM`: multiplies the copies of the given model(s)
- `--duplicate-grid X,Y`: multiplies the copies of the given model(s) by creating a grid (spacing can be customized with `--duplicate-distance`)
- `--cut Z`: cuts the given model(s) in two halves at the given Z and exports both the resulting models (note that the given coordinate should be relative to the object's bottom and not to its absolute position)

- `--cut-x X`, `--cut-y Y`: cuts the given model(s) in two halves along the X or Y axis at the given absolute coordinate and exports both resulting models
- `--cut-grid X,Y`: cuts the given model(s) according to a grid having the given pattern and exports ann the resulting models
- `--split`: detects unconnected parts in the given model(s) and splits them into separate objects
- `--repair`: this tries to repair any non-manifold meshes (this option is implicitly added whenever we need to slice the model, i.e. for `--cut*`, `--export-gcode`, `--export-svg`, `--export-sla-svg`)

These transform options are applied in the given order.

For convenience, if any of `--cut`, `--cut-x`, `--cut-z`, `--split`, `--repair` is supplied and none of the above actions is specified, `--export-stl` is implicitly assumed.

Examples:

```
slic3r --export-obj --scale 200% cube.stl
slic3r --cut 20 cube.stl
slic3r --cut 20 --export-3mf cube.stl
slic3r --gcode --merge model1.stl model2.s
slic3r --split cubes.stl
```

Export options

All the `--export-*` actions are affected by the following options:

- `--output FILENAME` (shortcut: `-o`): write output to the given file instead of the default one
  - if an existing directory is supplied instead of a file, the file will be created inside that directory using the automatically generated file name
- `--output-filename-format FORMAT`: this is the string pattern used to define the output file if `--output` is not specified. It defaults to `[input_filename_base].EXT` where EXT is the extension of the output format (`gcode`, `stl` etc.). See the documentation about placeholders.
  - when using `--export-svg`, the default format is `[input_filename_base]_[layer_num].svg`

Configuration

Configuration options affect toolpath generation (thus `--export-gcode` and `--export-sla-svg`). All the options available in the graphical interface of Slic3r can be used from command line as individual switches.

For example:

```
slic3r -g my_model.stl --layer-height 0.2
```

This will generate a file named *my_model.gcode* in the same directory as the input STL file. You may want to specify a custom output path:

```
slic3r -g my_model.stl --layer-height 0.2
```

To get the full listing, reference and defaults of available command line switches, just run:

```
slic3r --help-options
```

Most of the options accept an argument, like `--layer-height 0.2` or `--perimeters 3`. Howevere there are some boolean options that work as simple flags, like `--wipe` or `--avoid-crossing-perimeters`. To negate those options you just need to prepend `--no-` to them (as in `--no-wipe` or `--no-avoid-crossing-perimeters`).

Some options, including the ones related to multiple extruders, accept multiple values. You can just append them multiple times:

```
slic3r --infill-extruder 2 --nozzle-diamet
```

Note: the print/filament/printer presets defined in the graphical interface are completely ignored when running in command line mode. Slic3r will always default to its factory default settings.

In order to use your presets you'll need to export them with the *Export Config...* command, which is located in the *File* menu. It will prompt you to save a `.ini` file that you can load from command line this way:

```
slic3r -g my_model.stl --load my_config.in
```

You can override single options by appending them as command line switches:

```
slic3r -g my_model.stl --load my_config.in
```

You can also create a config file from command line:

```
slic3r --nozzle-diameter 0.35 --filament-d
    --temperature 185 --first-layer-temper
    --save my_config.ini
```

If you're an advanced user you can split your configuration into multiple *.ini* files and load them by appending multiple `--load` switches.

The `--ignore-nonexistent-config` will prevent Slic3r from throwing an error in case a non-existent file is supplied to `--load`.

One more way to use the print/filament/printer presets on command line is launching Slic3r with the `--autosave` option:

```
slic3r --autosave my_config.ini
```

This will launch the graphical interface of Slic3r but will automatically export the current configuration to the specified file. Thus, the last used presets will be remembered whenever you `--load` that file.

Positioning objects in the G-code coordinates

While in the graphical interface you can freely position your objects in the bed, command line provides two ways for telling position(s) to Slic3r:

1. Use the `--center X,Y` option for defining a point in G-code coordinates and Slic3r will center the print around that point:

   slic3r -g my_model.stl --center 40,40

(By default, the center point is automatically calculated as the centroid of the configured print bed.)

2. If you trust the coordinates of your STL file(s) and they are compatible with your machine's print bed, you can use `--dont-arrange`: Slic3r will leave the input locations untouched.

   slic3r -g my_model.stl --dont-arrange

Graphical interface options

- `--autosave` will automatically export the last selected config to a specified file (see the above paragraph about Configuration for its usage);
- `--datadir` followed by the path to a directory will tell Slic3r to use that directory for storing and

reading its configuration instead of the default system preferences directory. (See the page about configuration organization for more details.)