

Table of Contents

1 Overview	1
2 Installation/Setup	2
3 Usage	3
4 Limitations/Known Issues	9
5 More Information	10

1 OVERVIEW

In order to fabricate a PCB, most manufacturers require Gerber and Excellon files to be submitted for the PCB design. These files are standard within the PCB manufacturing industry and can be generated by most PCB design software. However, there is often a fairly large learning curve for the design software, and some of the fancier design packages can be too expensive for casual hobbyist usage. Pdf2Gerb tries to address these issues by allowing simpler, general purpose software to be used for the PCB design step.

Pdf2Gerb generates Gerber 274X photoplotting and Excellon drill files from PDFs of a PCB. Up to three PDFs are used: the top copper layer, the bottom copper layer (for 2-sided PCBs), and an optional silk screen layer. The PDFs can be created directly from any PDF drawing software, or a PDF print driver can be used to capture the Print output if the drawing software does not directly support output to PDF.

The general workflow is as follows:

1. Design the PCB using your favorite CAD or drawing software.
2. Print the top and bottom copper and top silk screen layers to a PDF file.
3. Run Pdf2Gerb on the PDFs to create Gerber and Excellon files.
4. Use a Gerber viewer to double-check the output against the original PCB design.
5. Make adjustments as needed.
6. Submit the files to a PCB manufacturer.

Please note that Pdf2Gerb does **NOT** perform DRC (Design Rule Checks), as these will vary according to individual PCB manufacturer conventions and capabilities. Also note that Pdf2Gerb is not perfect, so the output files must always be checked before submitting them. As of version 1.6, Pdf2Gerb supports most PCB elements, such as round and square pads, round holes, traces, SMD pads, ground planes, no-fill areas, and panelization. However, because it interprets the graphical output of a Print function, there are limitations in what it can recognize (or there may be bugs).

There are many excellent sources of documentation for the Gerber format itself as well as other PCB concepts, so that information will not be repeated here. The remainder of this document will briefly describe how to use Pdf2Gerb and what it does. For more detailed information, contact the authors or just look at the source code.

2 INSTALLATION/SETUP

You will need one or more of the following in order to use Pdf2Gerb:

- Pdf2Gerb – get the latest released version from <http://swannman.github.com/pdf2gerb/>
- Design software – use your favorite CAD or drawing software to create the PCB design
- PDF print driver – allows Print output to be saved as a PDF file
- Perl – any standard Perl engine should work
- Gerber viewer – a Gerber viewer should be used to check Pdf2Gerb output

These items will be summarized below. Specific installation instructions are not included here because that is typically included with each individual component.

Pdf2Gerb

Pdf2Gerb consists of a single text file named **Pdf2Gerb.pl**. This is a Perl script, so the text file = the source code. There is no “installation” procedure per se - just download it and place it into a folder where you would like to run it from. There are also several test/sample PDF files included in the TestFiles folder, but these are optional and Pdf2Gerb can be used without those.

Pdf2Gerb is released under GPL 3, but in the spirit with which Pdf2Gerb was developed please share any improvements or bug fixes with the original authors so that others may also benefit.

Design software

Anything can be used. Some drawing tools allow output directly to PDF, and those will work as long as they generate vector-based PDF graphics. The current version of Pdf2Gerb does not support raster (scanned) images within the PDF.

If your design software does not have a PDF output function that generates vector graphics, then one can be added simply by installing a PDF print driver (see below).

PDF print driver

A PDF print driver creates a virtual printer device that behaves like a regular printer. However, instead of printing on paper, it creates a PDF file of the printed output. This allows the Print output from any software to be “captured” and saved as a PDF.

There are various PDF print drivers available, and any of them should work as long as they can create PDFs with vector graphics. However, some testing and experimentation may be required. PDFCreator from sourceforge.net is known to work okay with Pdf2Gerb (see the links section). Note that Pdf2Gerb version 1.6 or later supports compressed PDFs (PDF 1.4 format).

Pdf2Gerb 1.6

Perl

A Perl engine is required in order to run Pdf2Gerb. Any standard Perl should work, although you may need to install some optional packages. Strawberry Perl for Windows (see the links section) is known to work with Pdf2Gerb without any additional packages to install.

Gerber viewer

To help validate the output from Pdf2Gerb, one or more Gerber viewers should be used. Any Gerber viewer can be used for this. Two that seem to work reasonably well are Gerbv and Viewplot (see the links section).

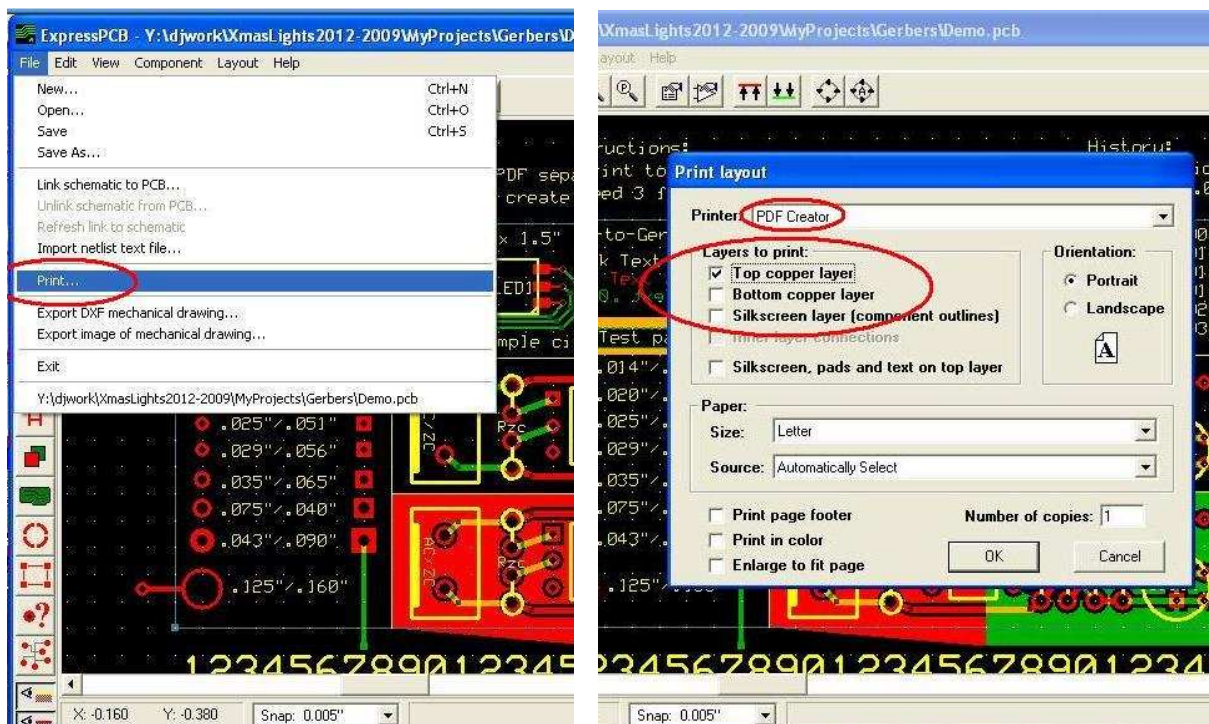
Note that even though a PCB might look okay in a Gerber viewer, it may not pass the PCB manufacturer's Design Rules, or vice versa. Therefore Gerber viewers are no substitute for a Design Rules check – both forms of validation should be used.

3 USAGE

Creating PDFs

You will need 1 - 3 PDFs of the PCB: one PDF layer for the top and bottom copper layers, as well as an optional silk screen layer. These can be defined as separate layers within one PDF or as separate PDFs, whichever is easier using the software tools you have available.

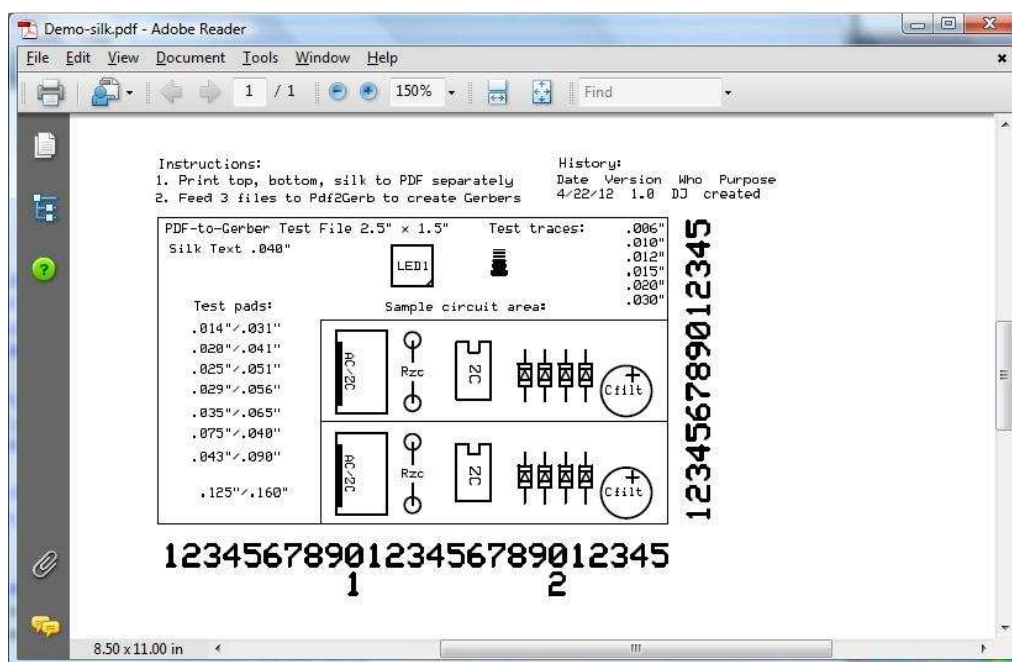
For example, if your design software has a Print function and you have a PDF printer driver such as PDF Creator installed, you could use the Print function to print the top, bottom, and silk screen layers one at a time as follows:



Pdf2Gerb 1.6

The layers can be printed in color or black-and-white, as long as drilled holes end up being white circles in the final output. Each layer must be printed separately because Pdf2Gerb does not separate layers by color; any non-white items will be treated as copper or silk areas.

The layers must be printed actual size in order to generate the correct PCB dimensions - do not use options such as "Enlarge to fit page". Leave other page decorations such as footers turned off – any non-white areas in the output will be treated as copper or silk areas. Look at the PDFs to verify that they have been correctly formatted. For example, a silk screen layer might look as follows:



If the PDFs look correct, the next step is to run them through Pdf2Gerb.

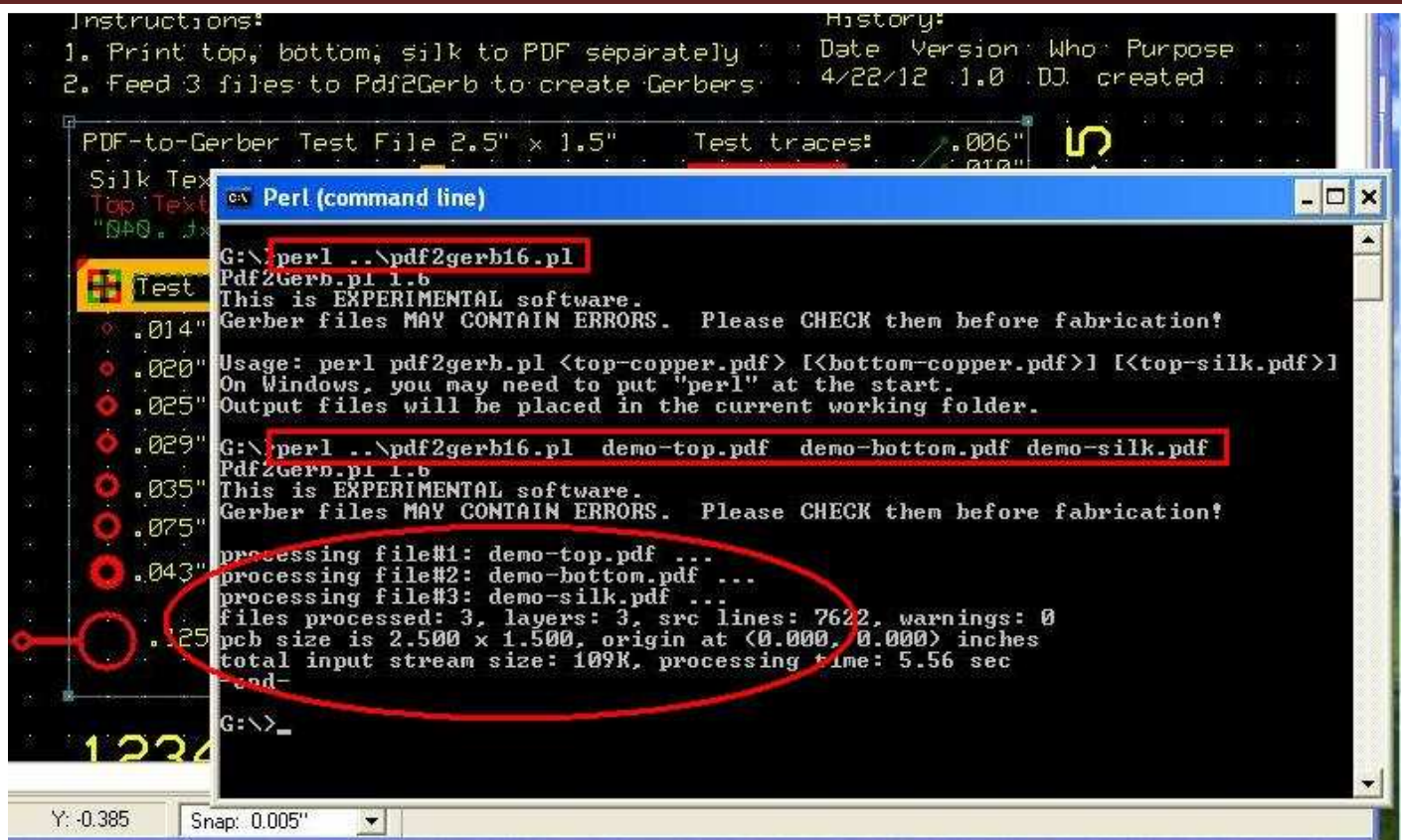
Running Pdf2Gerb

Open a console window and run Pdf2Gerb from the command line. On a Windows machine, you will probably need to start the command line with "perl" unless you are using a Unix-like interface such as CygWin.

With no command line arguments, Pdf2Gerb will just display a usage message and then exit.

Pdf2Gerb expects 1 – 3 PDF files as command line arguments, representing the top, bottom and silk screen layers. The layers can all be in the same PDF or they can each be in separate PDFs, but the order must be top copper, bottom copper, then silk screen. The file(s) will be read, layers parsed, and Gerber files created in the current working folder. Summary information will then be displayed, as shown below:

Pdf2Gerb 1.6



```
Instructions:
1. Print top, bottom, silk to PDF separately
2. Feed 3 files to Pdf2Gerb to create Gerbers

History:
Date: 4/22/12
Version: 1.0
Who: DJ
Purpose: created

PDF-to-Gerber Test File 2.5" x 1.5" Test traces: .006"

Silk Text
Top Text
"040. Jx"

Test
.014"
.020"
.025"
.029"
.035"
.075"
.043"
.125"

G:\>perl ..\pdf2gerb16.pl
Pdf2Gerb.pl 1.6
This is EXPERIMENTAL software.
Gerber files MAY CONTAIN ERRORS. Please CHECK them before fabrication!

Usage: perl pdf2gerb.pl <top-copper.pdf> [<bottom-copper.pdf>] [<top-silk.pdf>]
On Windows, you may need to put "perl" at the start.
Output files will be placed in the current working folder.

G:\>perl ..\pdf2gerb16.pl demo-top.pdf demo-bottom.pdf demo-silk.pdf
Pdf2Gerb.pl 1.6
This is EXPERIMENTAL software.
Gerber files MAY CONTAIN ERRORS. Please CHECK them before fabrication!

processing file#1: demo-top.pdf ...
processing file#2: demo-bottom.pdf ...
processing file#3: demo-silk.pdf ...
files processed: 3, layers: 3, src lines: 7622, warnings: 0
pcb size is 2.500 x 1.500, origin at (0.000, 0.000) inches
total input stream size: 109K, processing time: 5.56 sec
--end--
G:\>
```

Pdf2Gerb will generate up to 7 output files. If the names of the input layers or files are "ttd", "bbb" and "sss" (representing the top and bottom copper layers and the silk screen layer of the PCB), then the following output files will be created:

- ttd-top-copper(GTL).grb
- ttd-top-mask(GTS).grb
- ttd-top-outline(OLN).grb
- ttd-top-drill(DRD).txt
- bbb-bottom-copper(GBL).grb
- bbb-bottom-mask(BGS).grb
- sss-silk(GTO).grb

Depending on your chosen PCB manufacturer or the PCB itself, some of the output files may not be needed. You can rename them as needed. Pdf2Gerb suggests some names by using the source layer/file name as the first part (to indicate the source layer or file), and then appends additional words to indicate the purpose of the output files.

The "top-copper" and "bottom-copper" files contain traces and pads, and the "top-mask" and "bottom-mask" files contain the solder masks (if any pads were detected). Since the outline and drill holes are the same for the entire PCB, they are only created once (for the first layer), and are named "top-outline" and "top-drill". Since the silk screen layer does not have traces or pads, there is only a "silk" file generated for it.

Pdf2Gerb 1.6

Since the file naming conventions vary between PCB manufacturers, Pdf2Gerb uses a generic Gerber file extension for most of the files (".grb"), but includes an alternate

There are many tunable parameters within Pdf2Gerb, defined as constants up near the top of the source code. Depending on the PCB design and complexity, it may be necessary to adjust some of these values, but for simpler PCBs you can probably just leave them as-is. Here is a list of the tunable parameters:

Name	Purpose	Default value
APERTURE_LIMIT	Max #apertures to use	0 (no limit)
BEZIER_PRECISION	Controls speed and accuracy of Bezier curve rendering (used for circles, silk screen and thermal pads). Higher = more precise, slower, and generates larger files.	36
CIRCLE_ADJUST_MINX, CIRCLE_ADJUST_MINY, CIRCLE_ADJUST_MAXX, CIRCLE_ADJUST_MAXY	Used to adjust X/Y coordinates of circles; helps compensate for PDF rendering or arithmetic errors.	0, -1 pt (~ -.002"), +1 pt (~ .002"), 0
FILL_WIDTH	Line width to use when filling larger areas. Smaller = smoother, slower, and generates larger files.	.01"
GERBER_DEBUG	Controls amount of debug info included within Gerbers; higher values = more details	0 (none)
HOLE_ADJUST	Used to adjust hole sizes; helps compensate for PDF rendering or arithmetic errors.	+2.6 pts (~ +.004")
MAX_APERTURE	Largest aperture size to use for pads and traces. Larger areas will be filled using parallel lines.	Largest value from TOOL_SIZES, currently 0.220"
MAX_BYTES	Max size to read for each PDF file (in bytes).	10 MB
MAX_DRILL	Largest drill size to use. Larger areas will be unfilled copper areas (but not drilled).	Largest value from TOOL_SIZES, currently 0.125"
METRIC	Select metric vs. US units	FALSE (US)
PANELIZE	Number of times to repeat in X and Y directions, and whether to allow silk layer to overhang first/last panel.	1 x 1, TRUE (allow overhangs)
RECTPAD_ADJUST	Used to adjust rectangular pad sizes; helps compensate for PDF rendering or arithmetic errors.	0

Pdf2Gerb 1.6

Name	Purpose	Default value
RNDPAD_ADJUST	Used to adjust round pad sizes; helps compensate for PDF rendering or arithmetic errors.	-2 pts (~ -.003")
SOLDER_MARGIN	Clearance around pads for solder mask.	.012"
SQRPAD_ADJUST	Used to adjust square pad sizes; helps compensate for PDF rendering or arithmetic errors.	+5 pts (~ .001")
TOOL_SIZES	List of standard pad, hole and trace sizes; used to avoid irregular values (especially for drill sizes)	Various; add or remove values as desired
TRACE_ADJUST	Used to adjust trace sizes; helps compensate for PDF rendering or arithmetic errors.	0
WANT_DEBUG	Controls amount of debug info displayed to the console; higher values = more detail	0 (none)
WANT_STREAMS	Allows decompressed PDF streams to be saved to a file (for debug)	FALSE (do not save)

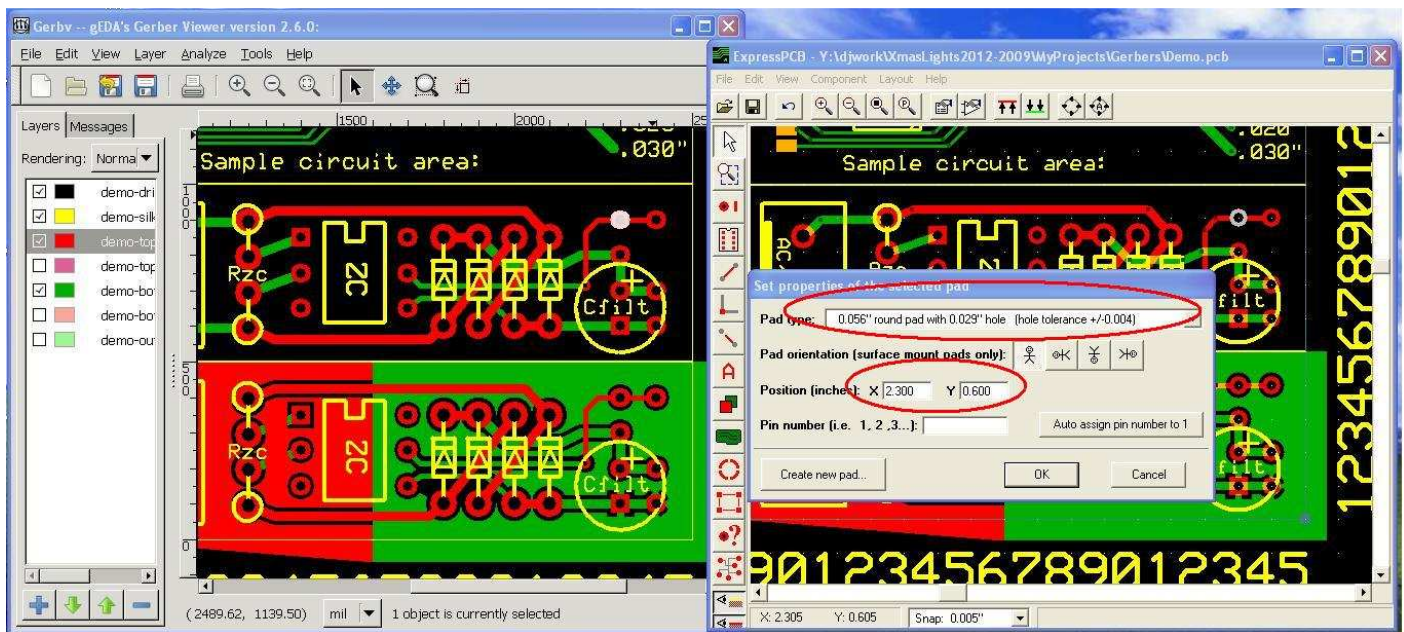
Since Pdf2Gerb.pl is just a text file, any text editor can be used to change it. When adjusting any values, make a backup copy first in case you need to roll back the changes.

Gerber validation

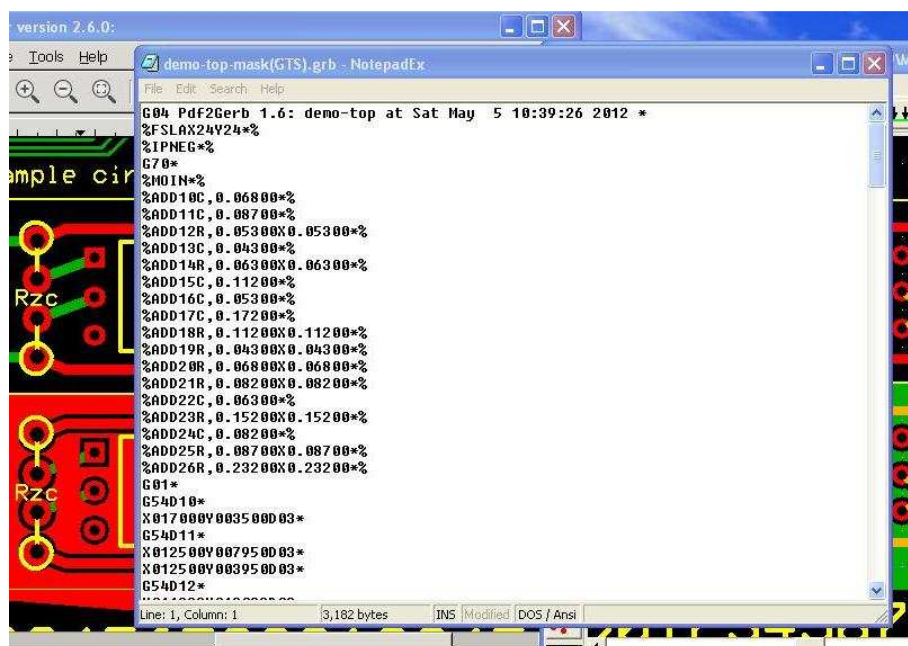
There are a few ways to validate that the generated Gerbers are correct. The first way is to visually inspect the files. For an initial assessment, display the PCB using your design software, side-by-side with the Pdf2Gerb output files displayed in a Gerber viewer. For this type of general comparison, it is helpful to use the same zoom settings in each software tool, and also to adjust layer colors and order so they match.

For areas that look questionable, zoom in closer and examine the properties of individual PCB elements in more detail. For example, you may need to verify that Pdf2Gerb selected the correct pad or hole sizes:

Pdf2Gerb 1.6



After a general visual inspection of the PCB layout, you can also open up the generated Gerber files and examine their contents directly, since they are just text files. For example, on a Windows box use Notepad to display the Gerber file contents:



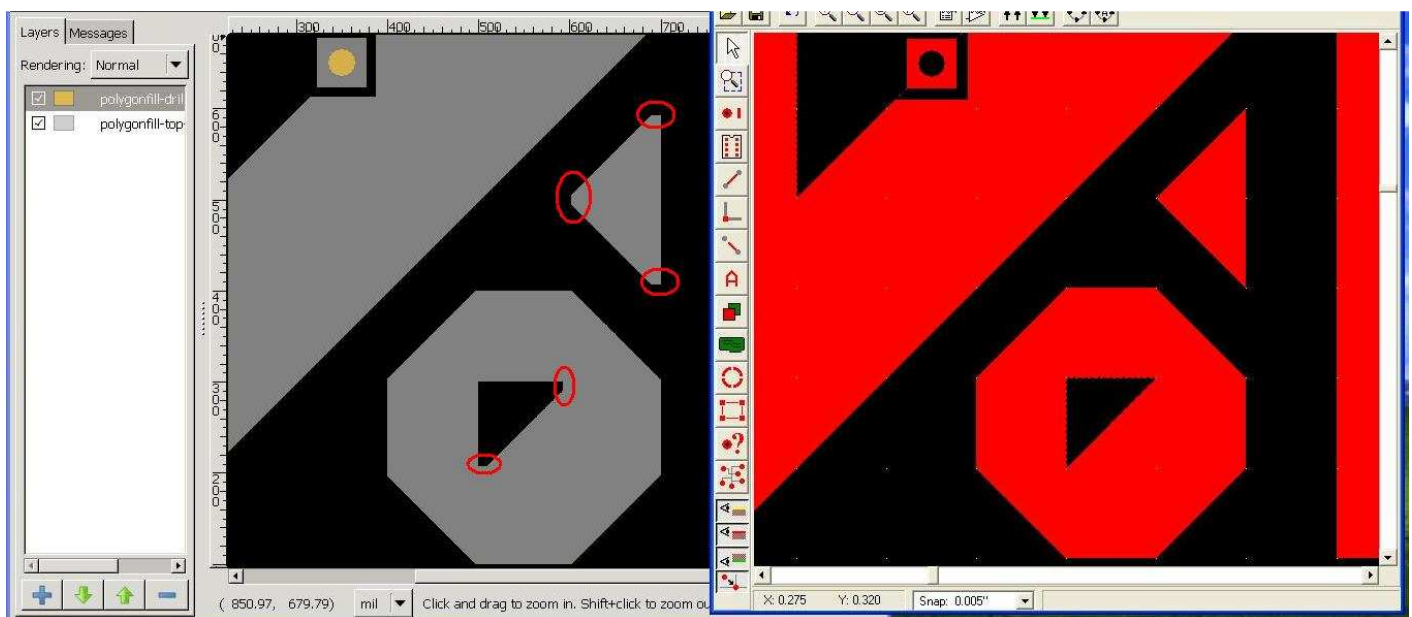
If all this looks good, then a final way to check is to use the PCB manufacturer's Design Rules checker. This will vary from one manufacturer to another, so check their web site for more details.

4 LIMITATIONS/KNOWN ISSUES

Although Pdf2Gerb 1.6 handles many PCB structures well, it has some limitations as noted below:

- PCB outline is assumed to be rectangular
- Holes in PDFs must be white circles; copper areas any color except white
- Some CAD packages have origin in top left, but PDF is bottom left
- Polygons and larger pads are filled with .001" lines; for non-rectangular ground planes, any points and intersections will be at least this wide (even if source CAD software shows them as points).
- Polygons (ground planes) where the edges define internal "cut-out" areas will be treated as such, even if the CAD software fills them.
- Larger pads that are filled will not have a solder mask opening (we don't want a solder mask opening on ground planes, for example).
- Panelization will squash text or other display elements outside the PCB border to avoid interference with adjacent panels (by design).

Below is an illustration of how aperture size limitations affect the fidelity of the corners of irregular shaped ground plane areas:



Although the source CAD files show sharp corners (to the right, above), the aperture size used to draw the fill lines leaves a slight flat edge at the corners (to the left, above).

The example files in the TestFiles folder can be used to test or experiment with the various features of Pdf2Gerb. Each test contains 4 files: top, bottom and silk PDFs that can be fed to

Pdf2Gerb 1.6

Pdf2Gerb, and an additional PDF that shows all layers together (for visual comparison; not for use directly with Pdf2Gerb).

The test files are listed below, as well as any known problems with them:

Test name	Comments/status
Demo	PASS, upper right ground plane, middle trace: margin a little large, bottom lower pad is .030 instead of .031; upper left ground plane: lower embedded rect too large, lower area slightly off
Groundplane	PASS, points/widths as noted above, irregular X/Y points off by $\sim .001 - .005$ "
HelloWorld	PASS
Pads+Holes	PASS, with exception of .043/.090 (600 dpi not high enough resolution to distinguish)
SMD	PASS; horizontal line at (.355 - .525, .805) looks $\sim .001$ lower than original CAD file
Text	PASS
Traces	PASS
Panelization of any of the above	PASS, with exceptions as noted above

Other

Some other future features might be as follows:

- elliptical pads? (draw short line seg using round aperture)
- use G02/G03/G75 circular commands instead of drawing circles with line segments?
- use hollow apertures? (pads are currently solid circles and hole is in center; this seems okay)
- make it run faster? (not too bad now)
- add command-line parameters instead of editing config constants?

5 MORE INFORMATION

Pdf2Gerb 1.4 and earlier

(c) 2010 Matthew M. Swann, swannman@mac.com

Pdf2Gerb 1.5*/1.6

(c) 2012 djulien17@thejuliens.net

More information about this work can be found at the following URL:

<http://swannman.github.com/pdf2gerb/>

This work is released under the terms and conditions set forth under the GNU General Public License 3.0. For more details, see the following:

Pdf2Gerb 1.6

<http://www.gnu.org/licenses/gpl-3.0.txt>

Below are some additional links to helpful related information:

Gerber files

- Gerber intro: http://www.apcircuits.com/resources/information/gerber_data.html
- G-codes + D-codes: <http://www.artwork.com/gerber/appl2.htm>
- 274X format: <http://www.artwork.com/gerber/274x/rs274x.htm>
- KiCAD Gerbers:
http://www.kxcad.net/visualcam/visualcam/tutorials/gerber_for_beginners.htm
- Excellon (drill file): <http://www.excellon.com/manuals/program.htm>
- Creating Gerbers: <http://www.sparkfun.com/tutorials/109>
- Gerbv (viewer): <http://gerbv.gpleda.org/index.html>
- Viewplot (viewer): <http://www.viewplot.com>
- Pdf2Gerb: <http://swannman.github.com/pdf2gerb/>

Other

- Cubic Bezier curves for circles: <http://www.tinaja.com/glib/ellipse4.pdf>
- Polygon fill algorithm: http://alienryderflex.com/polygon_fill/
- Point-in-polygon algorithm: <http://alienryderflex.com/polygon/>
- Perl help: <http://www.perlmonks.org>
- PDFCreator 1.3.2 (CAREFUL: TURN OFF SPYWARE CHECKMARKS DURING INSTALL AND CHOOSE DECLINE): <http://sourceforge.net/projects/pdfcreator/>
- Strawberry Perl (for Windows): <http://www.strawberryperl.com>

Revision history

Version	Date	Who	What
1.4	7/2011	MS	Initial versions
1.6	5/5/12	DJ	Misc enhancements, released for testing
1.6a	5/7/12	DJ	Updated links and expanded PDFCreator warning

-end-