# Eagle2freerouter SPECCTRA DSN converter usage guide

Version 8 March 2014

## Eagle2Freerouter.ulp updates

8-3-2014
Ignore microscopic curved lines (replace by straight segments)
26-2-2014
Fixed a bug in reading dimensionless values from the BRD XML file.
20-2-2014
Added start-up message with the ULP version and a warning to save the BRD file if design rules were changed.
Changed box showing DSN output file location for better readability.
15-01-2014
Removed the cumbersome temp file generation and restarting of the converter, since in V6 the design rules can be read directly from the XML board file.

**As of this version, save the Board file in Eagle before running this ULP, if you changed any design rules after the last Board save.**

30-12-2013
Fix to generate a default via even for single layer boards, otherwise the autorouter does not start.
30-09-2013
Earlier versions of Eagle2Freerouter.ulp implicitly drew the contours of the library parts as individual line or curve segment outlines. These outlines show the part location while running Freerouter, but they do not contribute to the routing process. Some library components, in particular high pin count SMD's, generate so many outline vectors that they prevent completing the Board description in Freerouter and the start of the actual routing. Furthermore, a call in the ULP to either the tplace(21) or the tdocu(51) component description will yield outline data for both layers (?), further complicating the drawing.

The integer parameter *Outline*, defined in the heading of this version of Eagle2Freerouter, controls the amount of outline drawing in the ULP:

*Outline* = 0: (default) means no components outlines
*Outline* = 1: Creates a simplified component outline, where the ULP tries to link line and curve segments as a single outline wire. The resulting component outline often contains artifacts.
*Outline* = 2: draw all lines individually as in the earlier versions.
Since the components outlines have no routing function, the default should normally be all you need.

## Description

Eagle2Freerouter.ulp is a utility for the Eagle schematic capture and PCB design software. It captures the structure of a PCB design (an Eagle <board_name>.brd file), which includes board dimensions, components, wiring, signal definitions and other data, sufficient to exactly define the intended circuit board (PCB) design.

The purpose of the Eagle2Freerouter utility is to transform the all the Eagle circuit board information in a standard language (DSN descriptors). The DSN descriptors can be read by Freerouter, a high quality 'auto-router' (ie a tool that places the actual circuit wiring in an optimal structure on the board), which is available in the public domain at: www.freerouting.net. The English forum on that website has the latest copy of Eagle2Freerouter.ulp. The code is modular, and you are free to modify the code to include other features. Modified or unmodified, all use of this converter is at your own risk in any case.

Freerouter has both auto and manual modes (manual use requires some training) but the easily accessible autoroute mode gives excellent results, even for complex boards, and it has an elegant manual mode to make wiring changes as well. The use of Freerouter is free (as in free beer), and there are no software limitations in board size, system complexity or the number of layers in a circuit board. Freerouter is network based, and you need a network connection to run the software.

The autoroute mode will first attempt to turn all signals (airwires) into wiring on the circuit board. It will then start the wiring optimization phase. No need to wait until the end of the process; as soon as you see an acceptable layout you can generate a SCRIPT file which updates Eagle with the circuit wiring generated by Freerouter. You can then make further design changes in Eagle, use Eagle auto routing (within the limits of your license) or generate the board manufacturing output.

**Moving components:** When enabled in the Freerouter menu, components and other objects can be moved around to facilitate routing. Select the device, hit delete to remove any wiring, and select move from the context menu (or 'm' on the keyboard). The context menu allows rotation or moving the component to the other side of the board. The moves are stored in the script file and written back to the Eagle board window together with the routing data.

**Getting started**

Make sure you have the Eagle2Freerouting.ulp file (can be downloaded from the Forum section of www.freerouting.net or from www.cadsoft.de ¦ downloads ¦ User language programs), and preferably store it in the directory where Eagle stores its ULP's.

Before you run the ULP from the Eagle board window make sure you have your design done and the Eagle Design rules (Eagle Design Rule Check button on the vertical tool bar) loaded with the DRC definitions that your board manufacturer provides. The default value of the DRC is accepted by most manufacturers, but their DRC settings allow often for a denser board wiring. The default value is suitable for home production of a board, but you might like to make the circuit wiring a bit wider (Sizes ¦ Minimum width in the DRC menu).

In the Eagle board window, run the ULP (File ¦ Run and then pick the Eagle2Freerouting.ulp), it will generate in the Eagle project directory a 'DRU' file called <board name>.dru. Go to www.freerouting.net, select 'Eagle user', click 'Start' the router, and your PC will load the Java application that is used to start the autorouter (you need to have a current version of Java installed).

You might get a security warning depending on the security settings on your system; accept to run the application, which is then downloaded and opens the Freerouter window. Click on 'Open your own design' and navigate to the directory where you have your – just produced – dru file (probably something like My Documents ¦ eagle ¦ <some project>).

When you run the dru file, a Freerouter board window opens and creates your board design that was captured in the DSN file. You can now click the 'autorouter' button in the command bar, and autorouting will start. You can stop and execute other commands by clicking with the mouse in the board window. The help facility gives more details, but for many items you have to be familiar with the aspects of advanced routing. One item that make routing a bit more flexible is to un-click the 'restrict pin exit directions (the 'Parameter ¦ Route' menu).

After laying all wires, Freerouter starts a (potentially long) process to reduce the wiring length and reduce the number of vias. For simple boards this is a matter of seconds or minutes, but for a complex multi layer board a run overnight can mean a substantial improvement. Rarely the autoroute process 'hangs' trying to re-route the last few wires to be laid over an over again. In that case stop the autorouter, delete a few wires around the critical place and start the router again. Or manually route a few circuits to get over the dilemma.

To transport the results to Eagle, stop the auto-router and and select 'Export Eagle session script' with the 'file' command in the board window. The file window defaults to the project file, where you can store the script file, that has an extension .scr. In the same board window, run this file with 'File ¦ script', navigate to your Eagle project directory (the default is the directory of Eagle example scripts). On the (optional) question from Eagle to rip up all existing wiring, click in general 'yes' as the script file contains a complete copy of the board.

After generating a script file you can restart autorouting and leave FreeRouting running in the background to see if you can get further improvements.

**Advanced use of Eagle2Freerouter**

Eagle2Freerouter provides advanced features, some of which are not available in Eagle:

- It optimizes the conversion of curves to line segments, even for curved board outlines, allowing the routing to accurately observe complex curved designs. You can create round board outlines with a circle, and modify polygons in any layer to have curved segments (select a Polygon wire with the INFO button and give the wire a non-zero Curve value).
- Selective locking to prevent the autorouter in Freerouting to rip-up or wires or vias:

  : Lock individual routed wires by setting their style in Eagle to WIRE_STYLE_SHORTDASH. After routing, running the script from FreeRouter to write the results in Eagle, their style will be back to continuous.

  : Lock wires, polygons and via's based on their signal name, by setting (multiple) signal names in the *wire_restrict, poly_restrict* and *via_restrict* parameters in the header of Eagle2Freerouter.ulp. This prevents for instance the autorouter to remove a group of vias to GND in an HF board design. Locking polygons protect them from moving, but not from breaking up with signal wires.

  : Polygons can be can be protected from breaking continuity by removing them before routing. Specify the relevant signal names (e.g. GND) in the *drop_poly* parameter in the ulp. This will for instance route the GND wiring as if the polygon was not there. When writing the script file back to Eagle, the still present polygon will 'absorb' the GND wiring it covers. It might still break in parts, but will not compromise continuity.
- For boards with more than two layers, Eagle2Freerouter can add single layer via or wire restrict (keepout) areas. Restrict patterns in one or more named user layers are inserted in the specified internal layer. This can prevent the rip-up of via's or breaking polygons in GND or power areas, or keep wiring away in the layer under an HF component. The details are described in the header of the ULP.
- As a byproduct of the design, the converter can route multiple board contours (circles or any line and curve based shape) on a single .brd file, as long as the individual contours do not overlap. This allows distribution and alignment of components between several boards. It makes it easy to add and align connectors between the boards (add two copies of a connector with the same signals, for each board one). The airwires between the boards are simply not routed (they will be replaced by the interconnects) and can be ignored when generating the final layouts.

**Current limitations**

The following items are not yet implemented:

- Via's with a shape difference between top and bottom footprints. For all internal layers vias have a single round shape.
- Via's with custom shapes formed by polygons (at least that has not been tested).
- Differential signals and meandering must be first routed in Eagle. To preserve them in Freerouter, lock them by setting their line style in Eagle to WIRE_STYLE_SHORTDASH.

--