

QCAD SVG to modern DXF

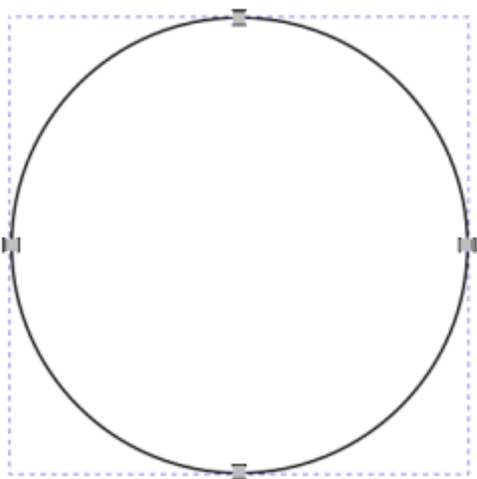
This extension uses [QCAD Pro](#) 3.X to convert an Inkscape SVG file to a nice DXF file, which has continuous polylines. It was created because the DXF12/DXF14 default export plugins of Inkscape do not allow to create proper DXF files in some manner, when it comes to more special use cases like part nesting with third party software. The rewrite of the Inkscape exporter plugins would be possible but is a complex thing to do so. As we support open source Software, let's use QCAD to solve it using it's powerful and easy to use libraries.

While Inkscape is not able to properly solve the polyline problem in exporting DXFs, it handles it correctly when importing DXF files, even the modern ones like DXF 2018 (R32). This was thoroughly tested though.

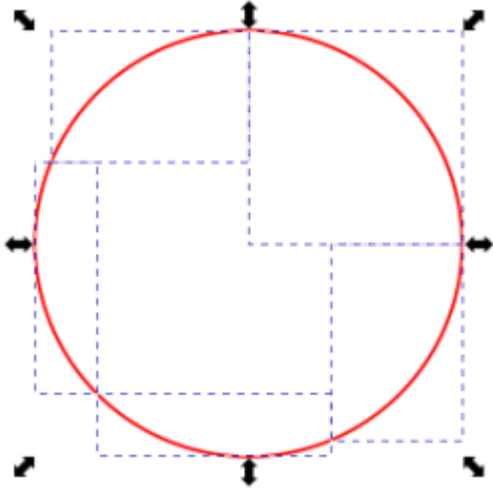
Warning: This extension does **not work with QCAD CE** (Community Edition), because it makes use of the pro's version (~40 €) advanced library functionality. So please do **not make use** of default installations like `apt install qcad` or `dnf install qcad`.

Short explanation of the issue

A default circle, converted to a path, looks like this in Inkscape:



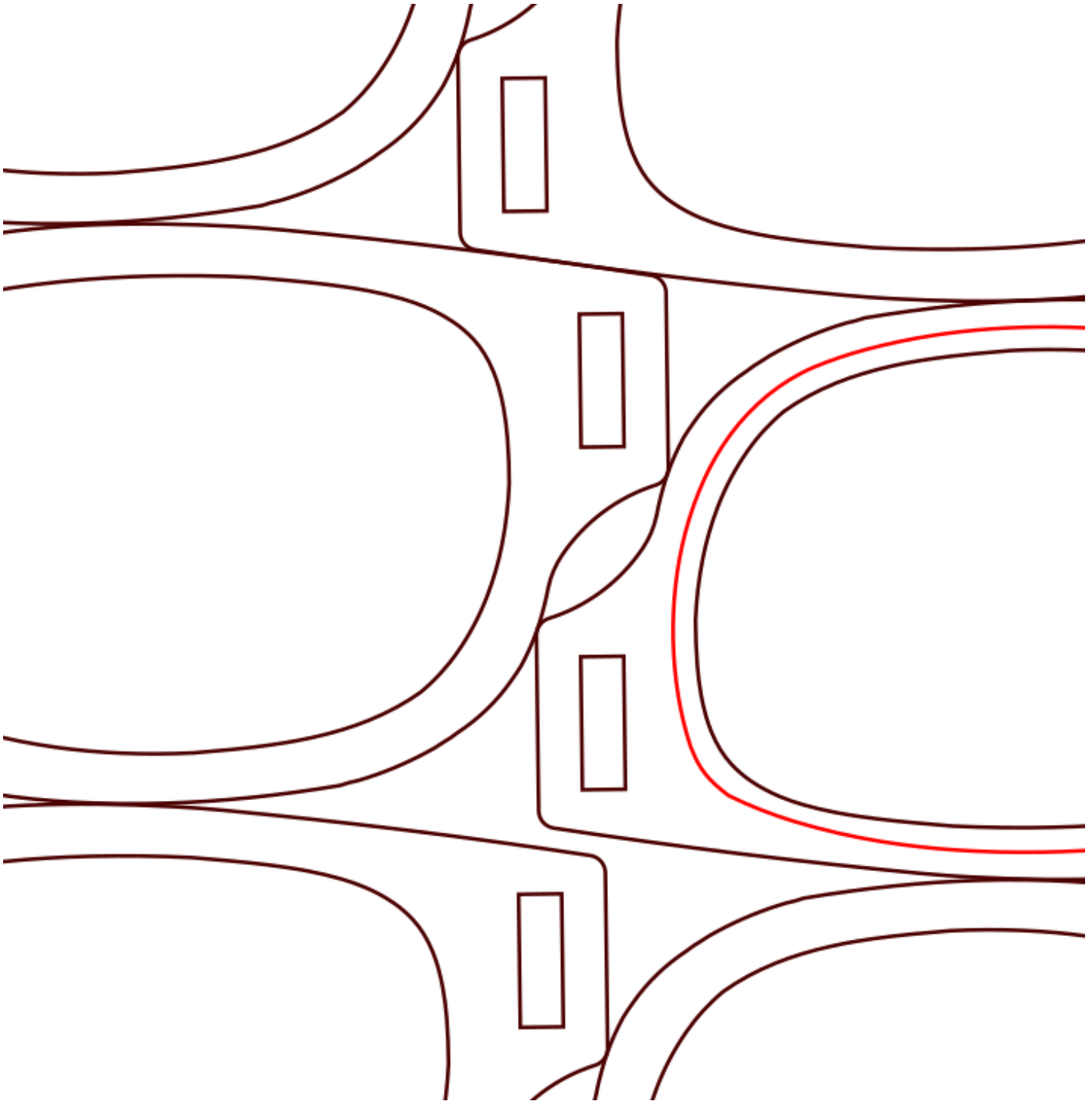
Exported to DXF 14 and reimported to Inkscape looks like this:



It comes as a group, but not as a single path.

Color misinterpretation

Using OpenDesign DXF library from QCAD, we get wrong colors. Black gets brown or blue for example. The reason is not known. So we use the builtin DXF R15 by dxflib by default. This supports correct color set.

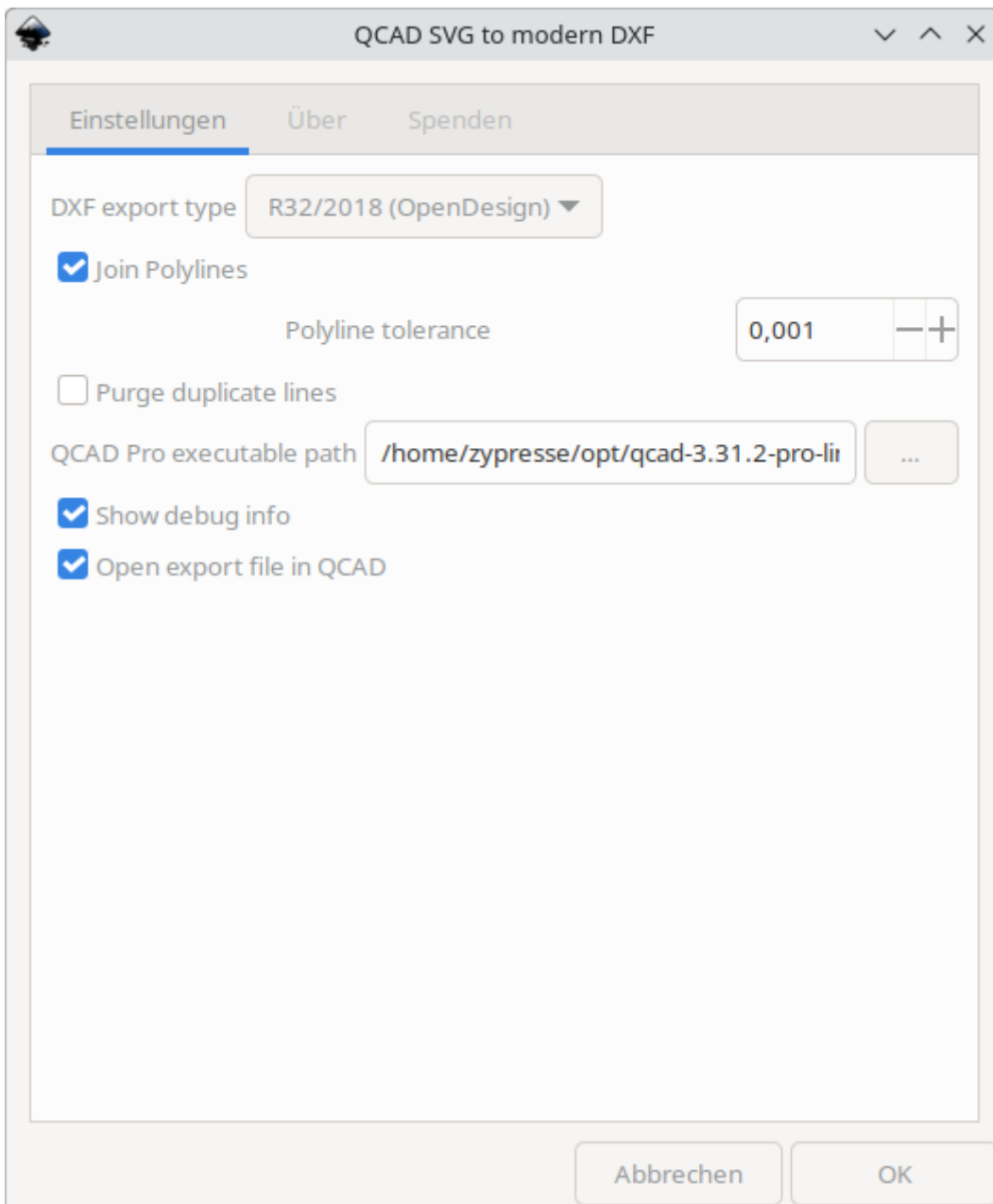


How to use the extension

Buy and install QCAD Pro by downloading it from the provided installation source URL (this info is received after payment).

```
#install qcad as regular user  
chmod +x *.run  
./qcad-3.28.2-pro-linux-x86_64.run
```

Then you can use the extension from GUI to export SVG to QCAD/DXF



Manual process

The following script is used to convert SVG to proper DXF:

To make this work, ensure QCAD is not already running.

```
vim polify-dxf.js
```

```
include("scripts/ImportExport/SvgImporter/SvgImporter.js")
include("scripts/Pro/Draw/Polyline/PolylineFromSelection/PolylineFromSelection.js");
include("scripts/ImportExport/SvgImporter/SvgImporterInit.js");
RFileImporterRegistry.registerFileImporter(new SvgImporterFactory());
```

```
qApp.applicationName = "MyApplication";
var storage = new RMemoryStorage();
var spatialIndex = new RSpatialIndexSimple();
var document = new RDocument(storage, spatialIndex);
var di = new RDocumentInterface(document);
const importer = new SvgImporter(document);
di.importFile("/home/tomate/Downloads/DXF-Test.svg");
di.selectAll();
var tolerance = 0.001;
PolylineFromSelection.autoJoinSegments(di, tolerance);

//use legacy dxflib to have correct colors. using the export of "OpenDesign" modify original
colors, like black #000000 is getting #00004C or sth. like this
//di.exportFile("/home/tomate/Downloads/DXF-Test.svg.dxf", "R32 DXF");
di.exportFile("/home/tomate/Downloads/DXF-Test.svg.dxf", "dxflib");
```

```
/home/himbeere/opt/qcad-3.28.2-trial-linux-qt5.14-x86_64/qcad -autostart polify-dxf.js >
/dev/null 2>&1
```

See also <https://qcad.org/rsforum/viewtopic.php?t=8471>

Version #1

Erstellt: 2025-05-24 14:07:32 CEST von Mario Voigt

Zuletzt aktualisiert: 2025-05-24 14:21:36 CEST von Mario Voigt