

Symbolize in QGIS with python

The qlyrx plugin development
Building symbologies from legacy definitions



<https://github.com/arc2qgis/lyrxtoqml>

Netta Beninson & Dror Bogin, June 2019, Open Source GIS

The pain



Yaara Herling ▸ **qgis** שו"ת

August 23, 2018 · 🌐



איך מסדרים סימבולוגיה של מבא"ת?

היו פה כמה שאלות לגבי זה וניסיתי את הקובץ ש **Reuven Kost Machado** העלה פעם והוא עובד חלקית. מה הפתרון שלכם? איך אפשר לעזור ליצור פתרון?



2



Roav Farago ▸ **qgis** שו"ת

December 4, 2018 · 🌐



ושב בענייני סימבולוגיה והמרות מ ARC ל QGIS

למישהו יש פתרון לסימבולוגיה של שכבות ייעודי קרקע בתכניות ישנות?

(אני מודע לקובץ סימבולוגיה תואם מבא"ת שעלה כאן. תודה לך

Netta Ben

ובכלל לכול מי שתמך ועמל על הדבר, זה מדהים...)

בתכניות ישנות, שדה הסימבולוגיה הוא מספר, וישנה עמודת ייעוד - (YEUD), ולא על פי הקוד החדש של מבא"ת.

בהרבה מקרים יש שכבת LYR אבל היא אינה שמישה ב QGIS (אולי אני לא מעודכן

וטועה...? ויש סוג של plugin או ממיר כלשהו, בענן או איזשהו כלי)

לאחר שיטוט בפורומים, הפתרון המוצע הוא המרה של קובץ ה LYR לקובץ SLD. העניין

הוא שאת ההמרה נדרש לעשות ב ARC (שאינ לי) ולייצא כבר שם את ה LYR ל - SLD.

עד כה לא הצלחתי למצוא כלי ממיר שעושה את העבודה.

אם למישהו יש פתרון, אנא שתפו.



Dror Bogin

June 11, 2018



בהשראת הכנס היום,

אלו מכם שעוסקים בתחום התכנון (סרטי או ציבורי), אני יודע שסימבולוגיות קבועות הן חסם מרכזי במעבר שלכם ל QGIS.

מה עוד מונע את המעבר הזה?

אם מחר מנהל התכנון משחרר את הקבצים במקום בlyr בפורמט שקריא בשתי התוכנות, אתם מסוגלים לבצע את המעבר מערקמאפ?



Reuven Kost Machado uploaded a file.

Admin · September 26, 2017



קובץ סימבולוגיה - יעודי הקרקע של המבא"ת מותאם (גם) ל QGIS.

מבא"ת הוא הפורמט שבו מוגשות תכניות מתאר בשנים האחרונות והקובץ יכול מאד לעזור למתכננים ולכל מי שצורך תבע"ות / תכניות מתאר.

הסימבולוגיה עדין לא טובה בכל מה שקשור לשימושים מעורבים, אבל הפיתרון לזה מורכב יותר (מקוה שיבוא מזור עם הגעתה של גרסה 3 ל QGIS) אבל תהנו ממה ממה שיש.

איך להשתמש?

שימו לב, זה לא פורמט של LAYER DEFINITION FILE !

1. כנסו למאפיינים של השכבה.
2. כפתור הSTYLE שלמטה!
3. לחצו על LOAD STYLE
4. שנו את הפורמט ל SLD ובחרו את הקובץ.
5. הוא יופיע לכם בבסייל כRULE BASE.

זהו.

אם מישהו/י יתנדב לתקן את הסימבולוגיות הבעייתיות תבוא עליי הברכה ויוכפרו לו כל עוונותיו גם בכיפור זה וגם בכיפור הבא.



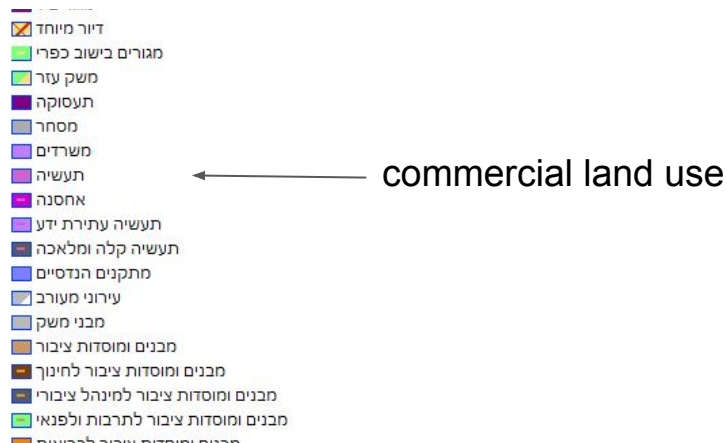
Heli Zecharya ▸ **qgis** שו"ת

תודה רבה. SLD לקובץ LYR יודעת כיצד ניתן להמיר קובץ... · 🌐 · Dec 28, 2017
(: למסייעים)

The pain

Users and Organizations who have used ArcGIS products for years, will usually have working habits and regulations rooted in specific symbology.

- For example



The symbology rules are stored in .LYR files

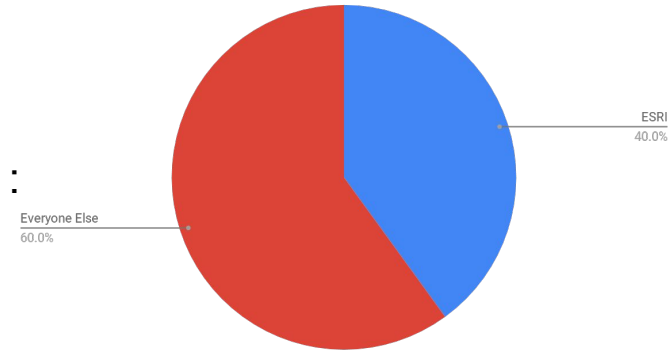
With the more elaborate .MXD which also stores the order of layers

The pain

The .LYR files contains symbology definitions and are :

- Binary format (non human readable)
- Usable only with ESRI software
- Some data of their data can be extracted with ESRI development libraries.

The GIS Market



ESRI holds more than 40% of the GIS software market, the `lyr file dilemma` represents a situation in which ESRI's users are blocked from switching to other options, due to non-existing conversion processes

Things have changed a bit with the release of ArcGIS Pro

The future

ArcMap .lyr file

```
D1 11 AA 82 00 C0 4F A3 3A 15 02 00 00 00 01 00 .....:.....  
44 00 00 00 43 00 3A 00 5C 00 4D 00 41 00 56 00 D...C.:.\.M.A.V.  
41 00 54 00 5F 00 41 00 50 00 50 00 4C 00 5C 00 A.T. .A.P.P.L.\.  
4D 00 6F 00 69 00 6E 00 53 00 65 00 72 00 76 00 M.o.i.n.S.e.r.v.  
65 00 72 00 5C 00 74 00 65 00 6D 00 70 00 5C 00 e.r.\.t.e.m.p.\.  
73 00 68 00 70 00 00 00 90 00 00 00 41 00 52 00 s.h.p.....A.R.  
43 00 49 00 4E 00 46 00 4F 00 3A 00 20 00 57 00 C.I.N.F.O.:.W.  
6F 00 72 00 6B 00 73 00 70 00 61 00 63 00 65 00 o.r.k.s.p.a.c.e.  
20 00 3D 00 20 00 5C 00 5C 00 4D 00 41 00 56 00 .=..\.M.A.V.  
41 00 54 00 2D 00 51 00 55 00 45 00 55 00 45 00 A.T.-.Q.U.E.U.E.  
31 00 31 00 5C 00 43 00 24 00 5C 00 4D 00 41 00 1.1.\.C.$.\.M.A.  
56 00 41 00 54 00 5F 00 41 00 50 00 50 00 4C 00 V.A.T._.A.P.P.L.  
5C 00 4D 00 6F 00 69 00 6E 00 53 00 65 00 72 00 \.M.o.i.n.S.e.r.  
76 00 65 00 72 00 5C 00 74 00 65 00 6D 00 70 00 v.e.r.\.t.e.m.p.  
5C 00 73 00 68 00 70 00 3B 00 00 00 1E 00 00 00 \.s.h.p.;.....  
53 00 68 00 61 00 70 00 65 00 66 00 69 00 6C 00 S.h.a.p.e.f.i.l.  
65 00 20 00 44 00 61 00 74 00 61 00 00 00 11 5A e..D.a.t.a...Z  
8E 58 8B D0 D1 11 AA 7C 00 C0 4E A3 3A 15 03 00 y | .
```

Translated to HEX with IDA - The Interactive Disassembler

ArcGIS Pro .lyrx file default symbology

```
"label" : "דיוור מיוחד",  
"patch" : "Default",  
"symbol" : {  
  "type" : "CIMSymbolReference",  
  "symbol" : {  
    "type" : "CIMPolygonSymbol",  
    "symbolLayers" : [  
      {  
        "type" : "CIMSolidStroke",  
        "enable" : true,  
        "capStyle" : "Round",  
        "joinStyle" : "Round",  
        "lineStyle3D" : "Strip",  
        "miterLimit" : 10,  
        "width" : 0.280000000000000003,  
        "color" : {  
          "red" : 255,  
          "green" : 0,  
          "blue" : 0,  
          "alpha" : 1  
        }  
      },  
      {  
        "type" : "CIMHatchFill",  
        "enable" : true,  
        "colorLocked" : true,  
        "lineSymbol" : {  
          "type" : "CIMLineSymbol",  
          "symbolLayers" : [  
            {  
              "type" : "CIMSolidStroke",  
              "enable" : true,  
              "capStyle" : "Round",  
              "joinStyle" : "Round",  
              "lineStyle3D" : "Strip",  
              "miterLimit" : 10,  
              "width" : 1,  
              "color" : {  
                "red" : 255,  
                "green" : 0,  
                "blue" : 0,  
                "alpha" : 1  
              }  
            }  
          ]  
        }  
      }  
    ]  
  }  
}
```

The future

The ArcGIS Pro symbology files are based on JavaScript Object Notation (JSON) which is human readable, meaning we can extract values from them.

```
{
  "type" : "CIMCharacterMarker",
  "enable" : true,
  "colorLocked" : true,
  "anchorPointUnits" : "Relative",
  "dominantSizeAxis3D" : "Y",
  "size" : 25.5,
  "billboardMode3D" : "FaceNearPlane",
  "characterIndex" : 34,
  "fontFamilyName" : "ESRI Default Marker",
  "fontStyleName" : "Regular",
  "fontType" : "Unspecified",
  "scaleX" : 1,
  "symbol" : {
    "type" : "CIMPolygonSymbol",
    "symbolLayers" : [
      {
        "type" : "CIMSolidFill",
        "enable" : true,
        "color" : {
          "type" : "CIMRGBColor",
          "values" : [
            230,
            0,
            169,
            100
          ]
        }
      }
    ]
  }
}
```

rgb(230, 0, 169) OK

Or Use HTML5:



And If we can get the values we can use them to define their matching parameters in QGIS.



qlyrx: use arcPro symbology in QGIS

Step 1

- Create *.lyrx from *.lyr

ArcGIS Pro => layer => right click

=> Sharing => Save as Layer file => lyrx

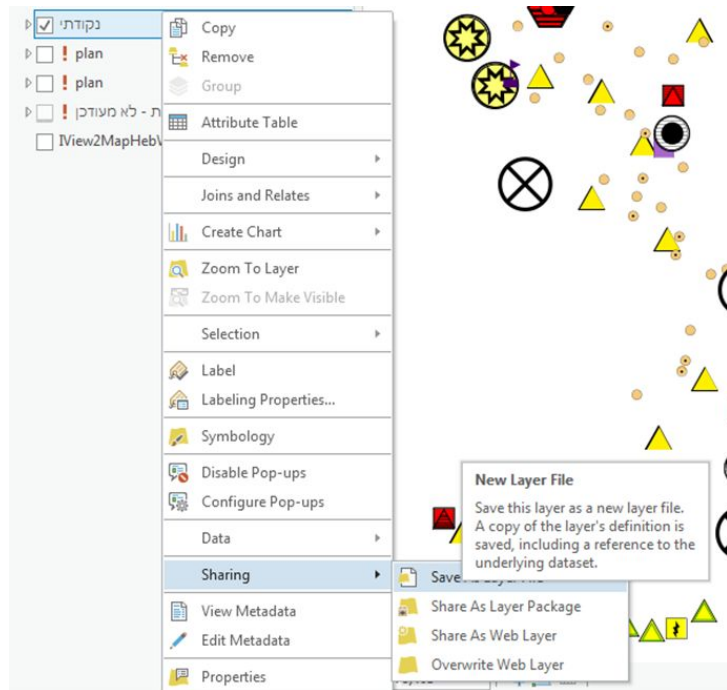
Step 2

- Install the qlyrx plugin

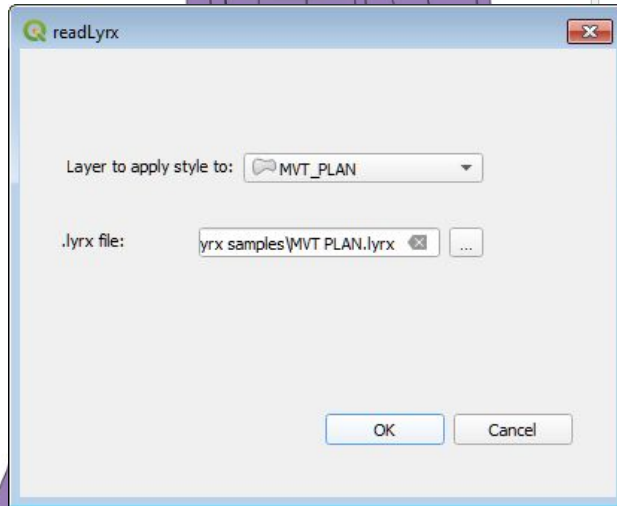


Step 3

- Run the plugin, choose a layer and a lyrx file



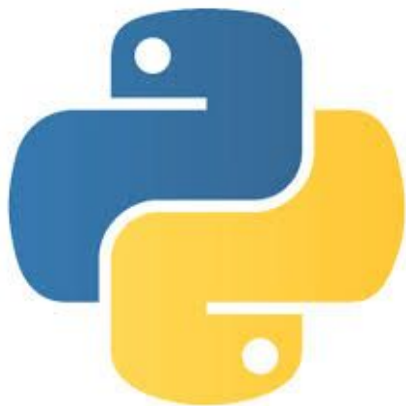
	yMin	xMax	yMax
1	654252.956524	176886.534713	654838.7709





Implementation

How It's Done

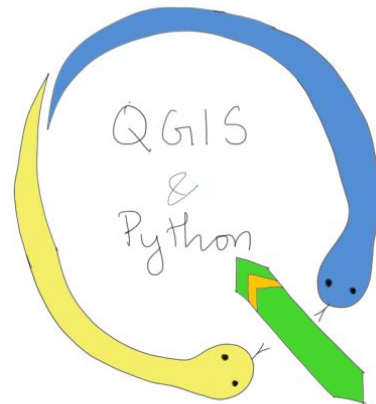


Basic definitions - [QgsSymbol](#)

Symbol Creation

[QgsSymbol](#).defaultSymbol(*QgsWkbTypes.GeometryType*)

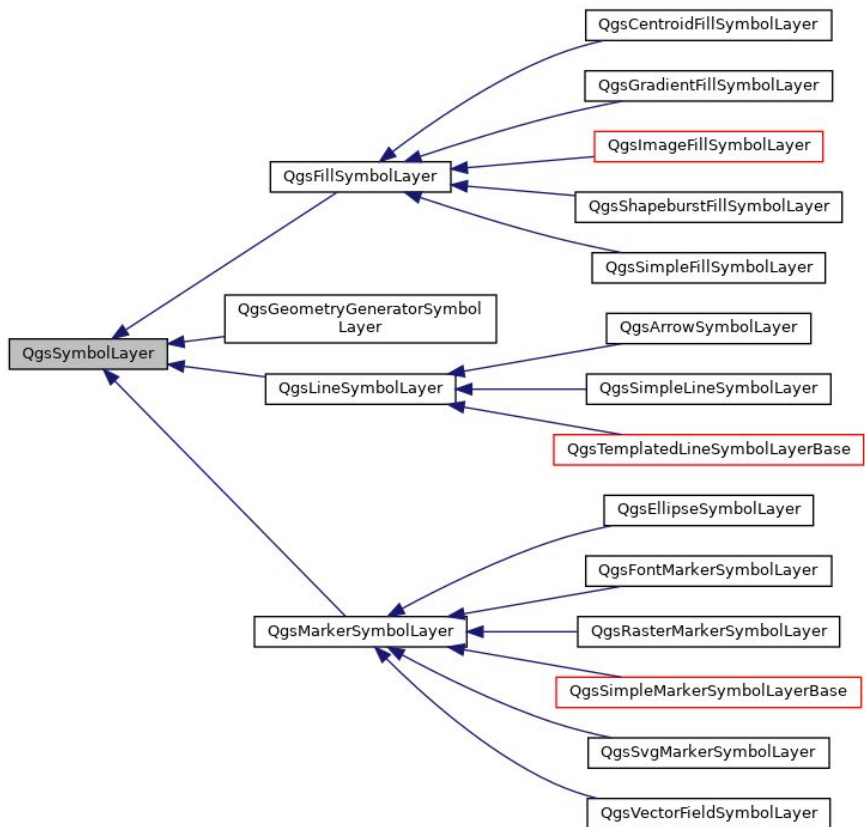
- Create a default symbol
- Customize color/stroke/opacity
- Symbol can have multiple symbolLayers
 - get/count
 - clone/move/delete



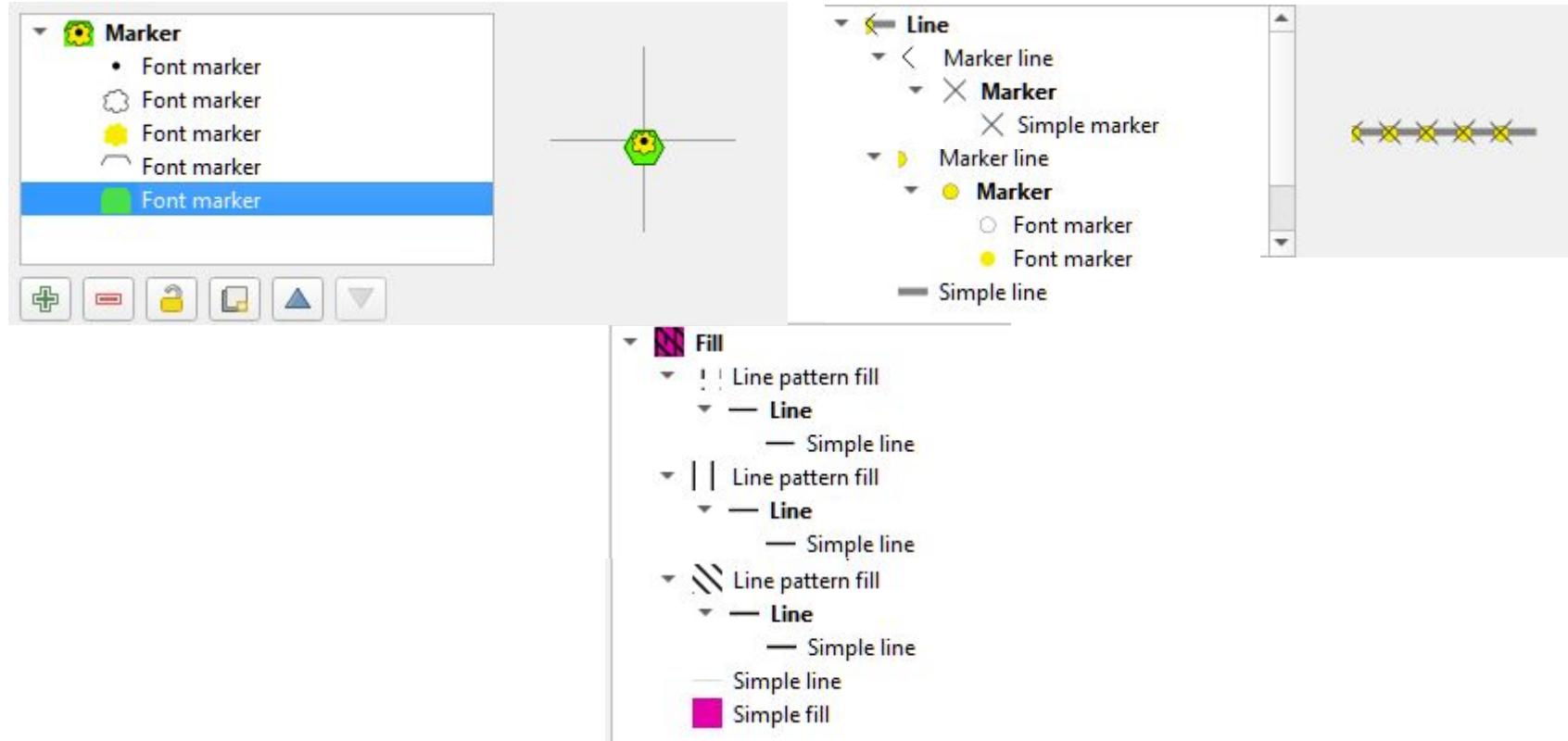
Basic definitions - [QgsSymbolLayer](#)

[QgsSymbolLayer](#)

abstract and utilities

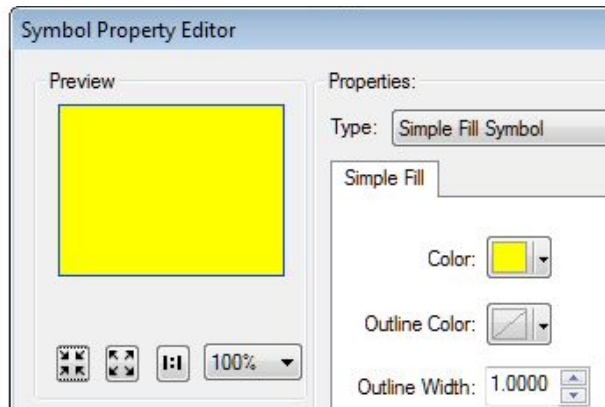


Complex symbols examples



Simple symbol *.lyrx

- ☒ מגורים ג
- ☒ מגורים ב-יעוד שהתבטל
- ☒ מגורים ב-2-יעוד שהתבטל
- ☒ מגורים ב-1 - יעוד שהתבטל
- ☒ מגורים ב
- ☒ מגורים א-3- יעוד שהתבטל
- ☒ מגורים א-2-יעוד שהתבטל
- ☒ מגורים א-1-יעוד שהתבטל
- ☒ מגורים א
- ☒ מגורים
- ☐ MVT_PLAN
- ☐ יעודי קרקע
- ☐ יעודי קרקע



```
{
  "type" : "CIMUniqueValueClass",
  "label" : "מגורים א",
  "patch" : "Default",
  "symbol" : {
    "type" : "CIMSymbolReference",
    "symbol" : {
      "type" : "CIMPolygonSymbol",
      "symbolLayers" : [
        {
          "type" : "CIMSolidStroke",
          "enable" : true,
          "colorLocked" : true,
          "capStyle" : "Butt",
          "joinStyle" : "Miter",
          "lineStyle3D" : "Strip",
          "miterLimit" : 10,
          "width" : 0.28000000000000003,
          "color" : {
            "type" : "CIMRGBColor",
            "values" : [
              255, 255, 0
            ]
          },
          "dash" : [
            1, 0
          ],
          "type" : "CIMSolidFill",
          "enable" : true,
          "colorLocked" : true,
          "color" : {
            "type" : "CIMCMYKColor",
            "values" : [
              0, 0, 0, 0
            ]
          }
        }
      ],
      "values" : [
        {
          "type" : "CIMUniqueValue",
          "fieldValues" : [
            "20"
          ]
        }
      ]
    },
    "visible" : true
  },
}
```



Simple symbol QGIS

lyrx def: CIMSolidFill, CIMSolidStroke

Create basic fill (**parseSolidFill**)

```
symbol = QgsSymbol.defaultSymbol(layer.geometryType())
```

```
symbol.setColor(new_color)
```

Create stroke (**parseStroke**)

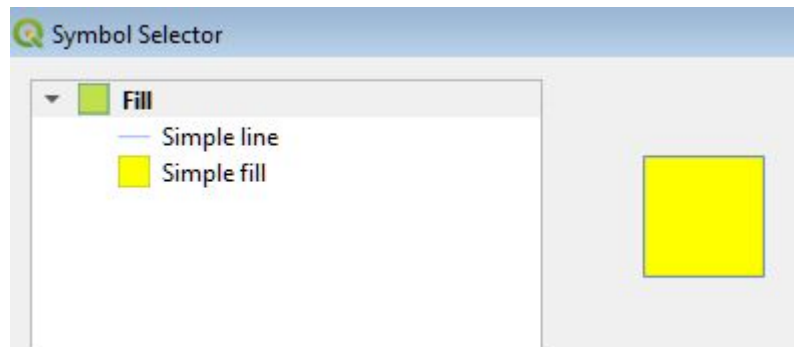
```
symbol_layer = QgsSimpleLineSymbolLayer()
```

```
symbol_layer.setColor(stroke_color)
```

```
symbol.appendSymbolLayer(symbol_layer)
```

Notes:

- As a default 2 symbols are created for simple fill with stroke





Plugin conversion functions

Categories & Simple Renderer functions

parseSolidFill - CIMSolidFill

parseStroke - CIMSolidStroke

parseLineFill - CIMHatchFill

parseStrokeEffects - effects :{"type" : "CIMGeometricEffectDashes","dashTemplate" : 'x,y'}

parseVectorSymbolLine* - CIMVectorMarker

parseCharacterFill - CIMCharacterMarker

parsePictureFill - CIMPictureFill

Utils

parseLineCap

parseLineJoin

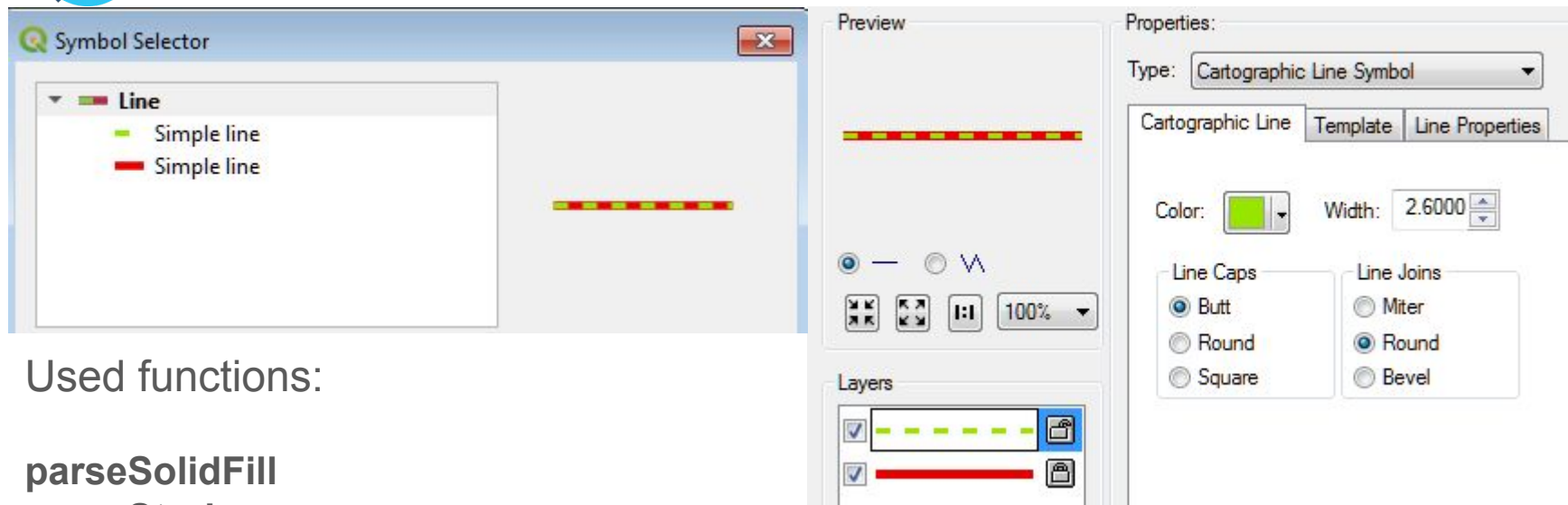
tweakHaloSymbol

colorToRgbArray

***base_symbol.subSymbol().changeSymbolLayer(0, subSymL), QgsSimpleMarkerSymbolLayerBase.decodeShape..**



Advanced *.lyrx conversion - Example 1



Used functions:

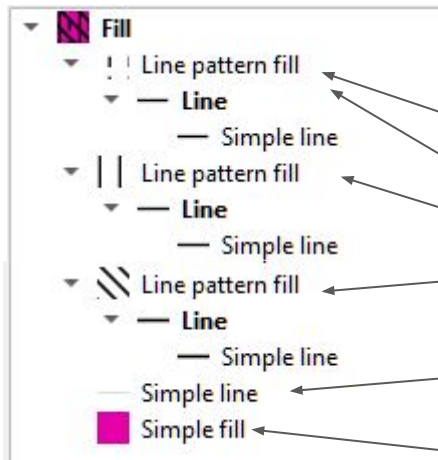
`parseSolidFill`

`parseStroke`

`parseStrokeEffects`



Advanced *.lyrx conversion- Example 2

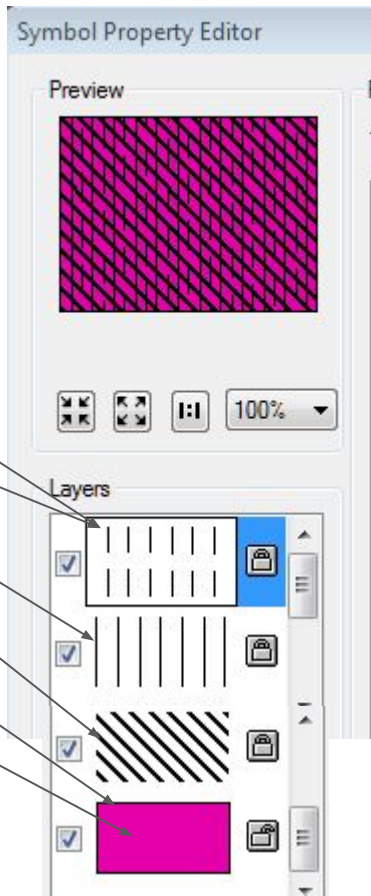


parseStrokeEffects

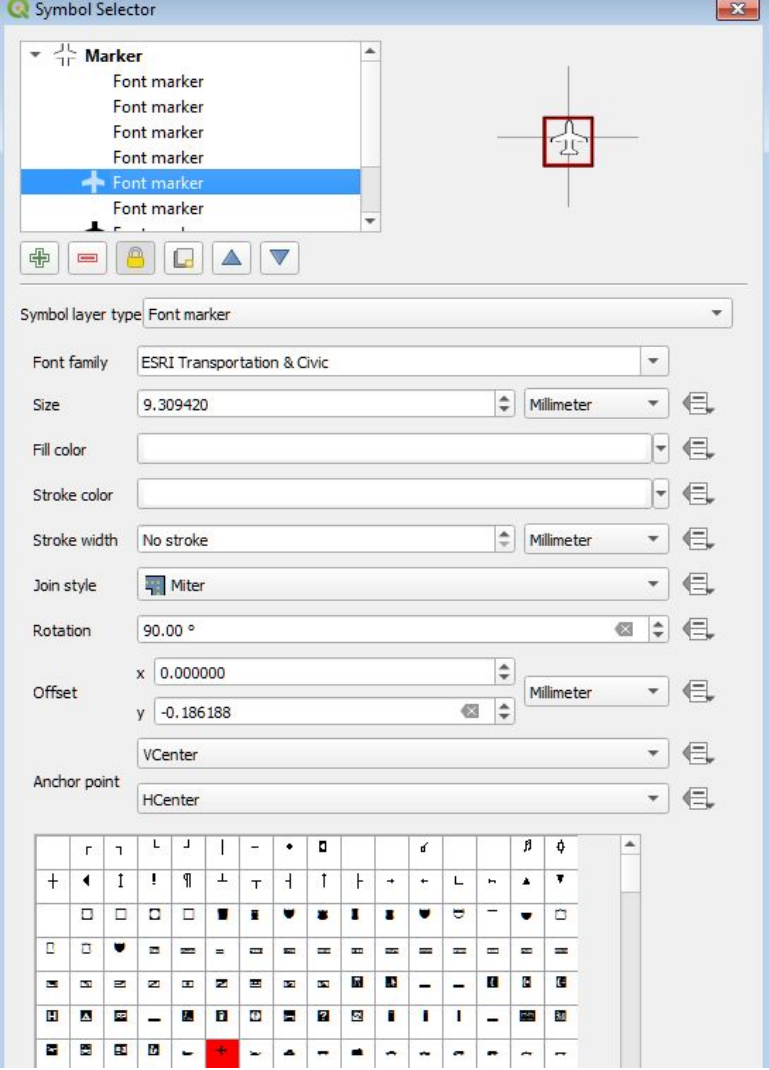
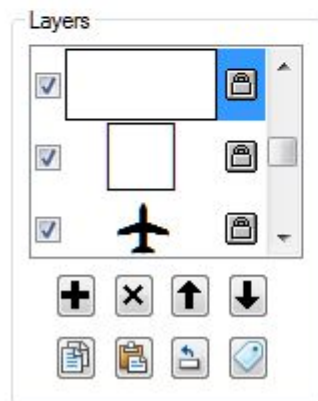
parseLineFill

parseStroke

parseSolidFill

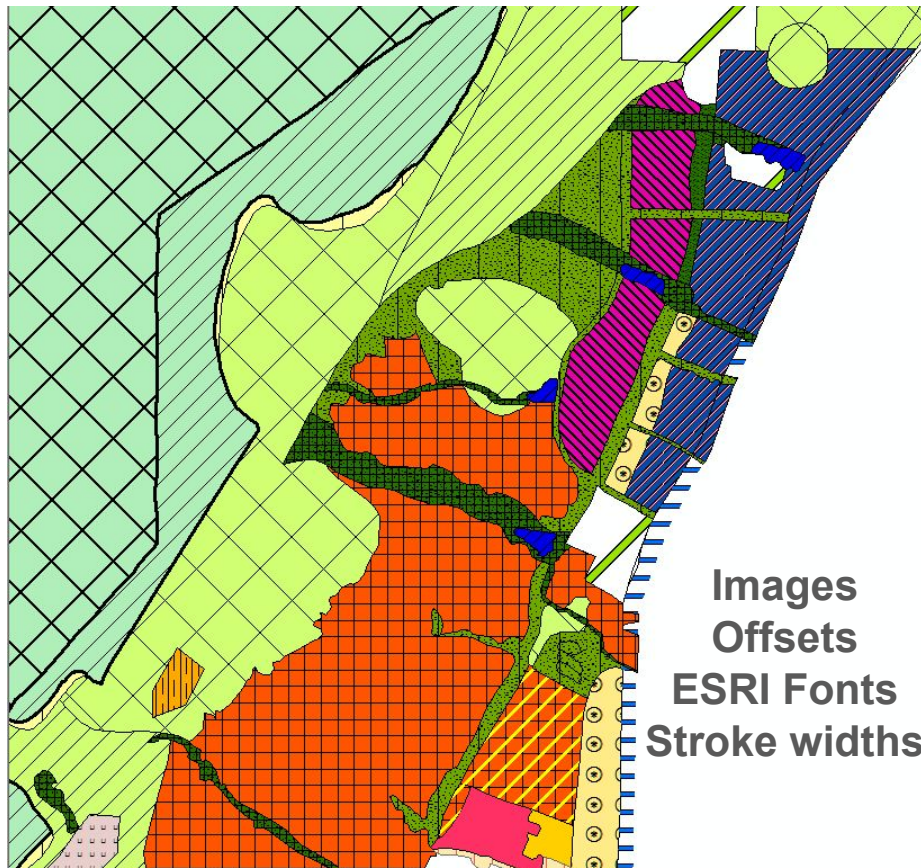


Symbol Property Editor

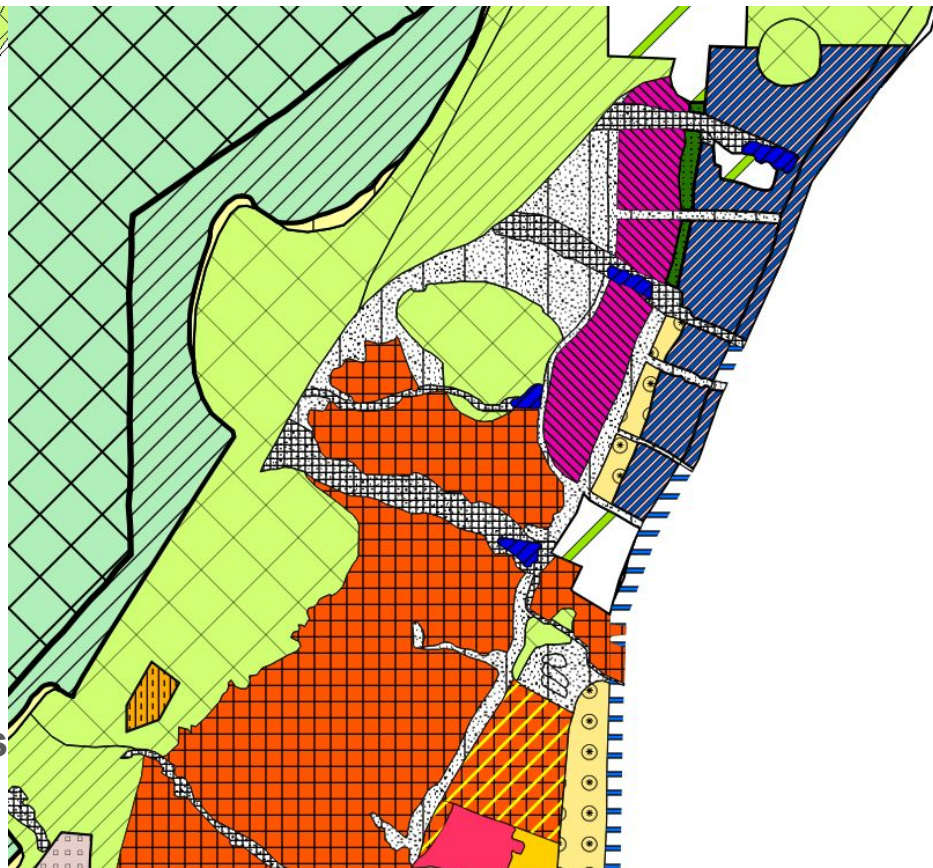




Challenges in *.lyrx conversion

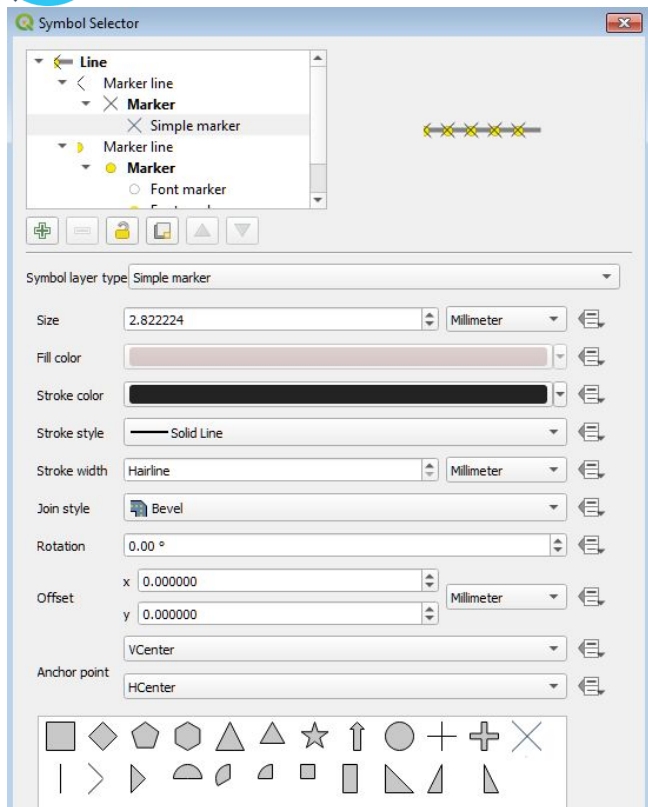


Images
Offsets
ESRI Fonts
Stroke widths

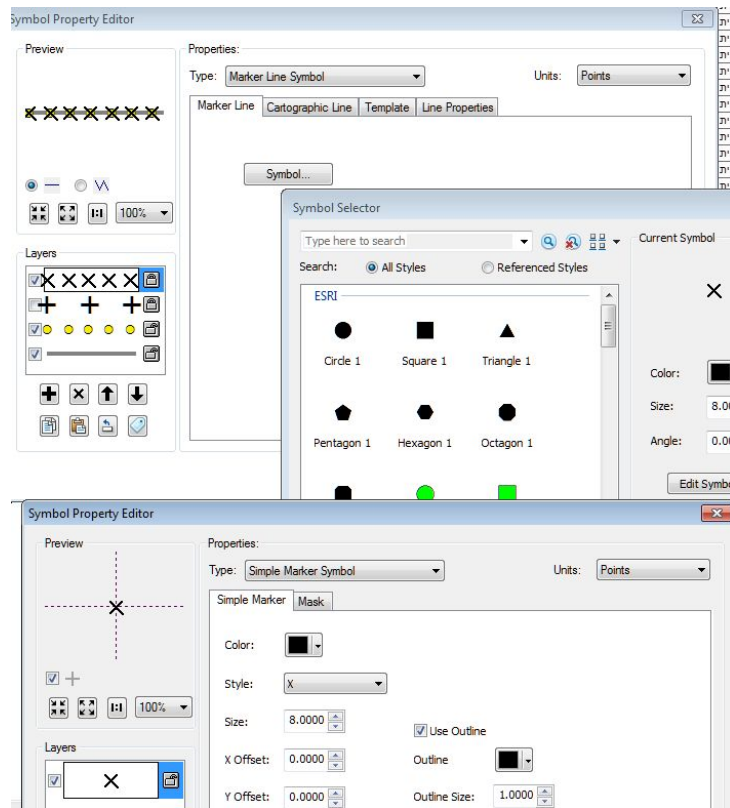




Challenges in *.lyrx conversion

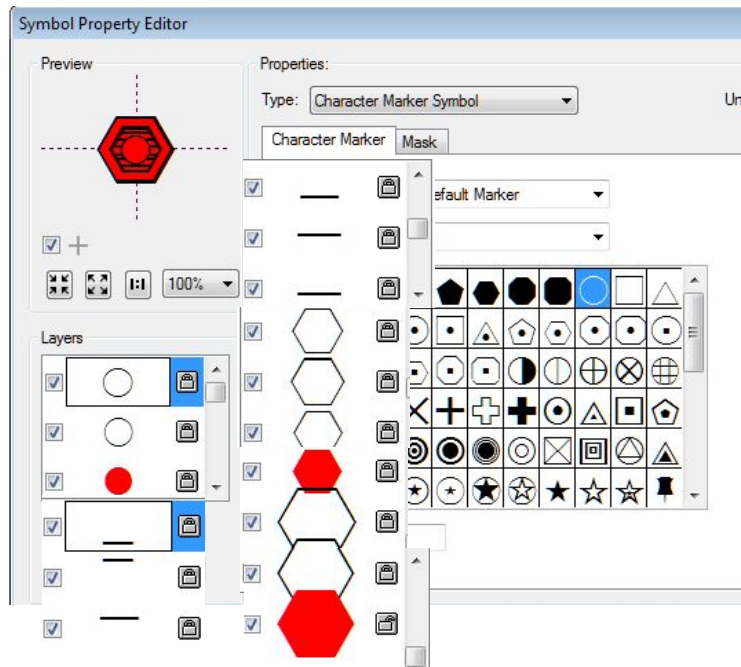


Marker symbols

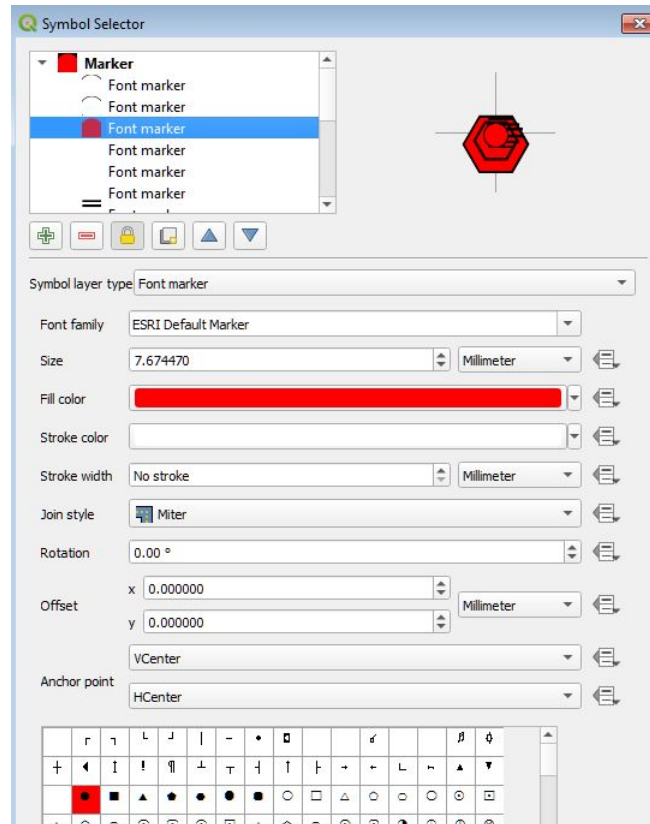




Problems in *.lyrx conversion



Multi shapes
Offset
Rotation
ESRI Fonts





Known issues

Missing in QGIS:

- Scale symbols by reference scale
- Halo
- ESRI fonts

Dev Stack:

- Graduated symbols

Bugs:

- Marker rotation offset problem
- The plugin does not report what went wrong when something fails





We need YOU!

- User interface
 - Defaults
 - Previews
 - On the fly edits
- Symbols definitions
 - Special shapes and definitions
 - Labels
 - Fallbacks
 - Svg specialist :)
- Refactoring
 - Automation of shape selection
 - symbolLayers drilldown
- Documentation and tests

