

TXL Wizard: Reference Manual

Esteban Marín

2016-05-13

Contents

1	Introduction	3
1.1	What does it do?	3
1.2	Technical Information	3
2	TXLWizard Example	4
3	Technische Dokumentation	8
3.1	Module	8
3.1.1	Methoden	8
3.1.2	Eigenschaften	8
3.2	Strukturen	8
3.2.1	DataStructure	8
3.2.2	FieldStructure	11
3.2.3	PageStructure	11
3.2.4	FormFieldStructure	11
4	TXLConverter	12
	Appendices	13
A	TXLWizard: Advanced Example	13

1 Introduction

This document describes the usage and technical reference of the python program “TXL-Wizard” written by Esteban Marin (estebanmarin@gmx.ch).

1.1 What does it do?

The “TXLWizard” provides routines for generating TXL files (.txl) for the preparation of E-Beam lithography masks using python code. The TXL files can be processed with BEAMER. See the following links:

- <http://genisys-gmbh.com/web/products/beamer.html>
- http://cad035.psi.ch/LB_index.html
- http://cad035.psi.ch/LBDoc/BEAMER_Manual.pdf

The generated TXL files are also converted to HTML / SVG for presentation in any modern browser or vector graphics application.

Moreover, a command line interface “TXLConverter” provides conversion of existing TXL files to HTML / SVG (See Section 4).

1.2 Technical Information

The “TXLWizard” is written in python and will run in Python version 2.7+ and 3.1+. In order to use the “TXLWizard”, package must be available as a python package, i.e. either it must be copied to `Path.to.my.python.installation/site-packages/` or to the path where your script is located.

Alternatively, you can also prepend the following command to your python script:

- Windows: `sys.path.append('path to the TXLWizard')`
- Linux: `sys.path.append('path to the TXLWizard')`

2 TXLWizard Example

The following code demonstrates a simple example usage of the “TXLWizard” for generating TXL files with python code.

The code can be found in the file `Content/Example.Simple.py`.

The resulting image is shown in Figure 1. A more advanced example is shown in Section A

```

1 #####
2 # Import Libraries #
3 #####
4
5 # Import TXLWriter, the main class for generating TXL Output
6 import TXLWizard.TXLWriter
7
8 # Import Pre-Defined Shapes / Structures wrapped in functions
9 import TXLWizard.ShapeLibrary.EndpointDetectionWindows
10 import TXLWizard.ShapeLibrary.Label
11
12 # Import math module for calculations
13 import math
14
15
16 #####
17 # Sample / Structure Parameters #
18 #####
19
20 # Define all sample parameters
21 SampleParameters = {
22     'Width': 8e3,
23     'Height': 8e3,
24     'Label': 'Simple Demo',
25 }
26
27 # Define all structure parameters
28 StructureParameters = {
29     'Circle': {
30         'Radius': 50,
31         'Layer': 3
32     },
33     'CircleArray': {
34         'Columns': 6,
35         'Rows': 5,
36         'ArrayXOffset': 500,
37         'ArrayYOffset': -500,
38         'ArrayOrigin': [0.75e3, 3e3],
39         'Label': 'R{:d}C{:d}',
40     }
41 }
```

```

42 |
43 |
44 | #####
45 | # Initialize TXLWriter #
46 | #####
47 | TXLWriter = TXLWizard.TXLWriter.TXLWriter(
48 |     Width=SampleParameters[ 'Width' ],
49 |     Height=SampleParameters[ 'Height' ]
50 | )
51 |
52 | #####
53 | # Define Structures #
54 | #####
55 |
56 | ## Sample Label ##
57 |
58 | # Give the sample a nice label
59 | SampleLabelObject = TXLWizard.ShapeLibrary.Label.GetLabel(
60 |     TXLWriter,
61 |     SampleParameters[ 'Label' ],
62 |     OriginPoint=[0.5e3, 1. * SampleParameters[ 'Height' ] / 2. -
63 |         500],
64 |     FontSize=150,
65 |     StrokeWidth=20,
66 |     Layer=1
67 | )
68 |
69 | ## Endpoint Detection ##
70 |
71 | # Use Pre-Defined Endpoint Detection Windows
72 | TXLWizard.ShapeLibrary.EndpointDetectionWindows.
73 |     GetEndpointDetectionWindows(
74 |         TXLWriter, Layer=1)
75 | ## User Structure: Circle ##
76 |
77 | # Create Definition Structure for Circle that will be reused
78 | CircleStructure = TXLWriter.AddDefinitionStructure( 'Circle' )
79 | CircleStructure.AddPattern( 'Circle',
80 |     Center=[0, 0],
81 |     Radius=StructureParameters[ 'Circle' ][ 'Radius' ],
82 |     Layer=StructureParameters[ 'Circle' ][ 'Layer' ]
83 | )
84 | )
85 |
86 | # Create array of the definition structure above
87 | CircleArray = TXLWriter.AddContentStructure( 'CircleArray' )

```

```

88 CircleArray.AddPattern( 'Array',
89     ReferencedStructureID=CircleStructure.ID,
90     OriginPoint=StructureParameters[ 'CircleArray' ][ 'ArrayOrigin'
91         ],
92     PositionDelta1=[
93         StructureParameters[ 'CircleArray' ][ 'ArrayXOffset' ], 0
94     ],
95     PositionDelta2=[
96         0, StructureParameters[ 'CircleArray' ][ 'ArrayYOffset' ]
97     ],
98     Repetitions1=StructureParameters[ 'CircleArray' ][ 'Columns' ],
99     Repetitions2=StructureParameters[ 'CircleArray' ][ 'Rows' ]
100 )
101
102
103 #####
104 # Generate Output Files #
105 #####
106
107 # Note: The suffix (.txl, .html, .svg) will be appended
108     automatically
109 TXLWriter.GenerateFiles( 'Masks/EM160509_Demo_Simple' )

```

Content/Example_Simple.py

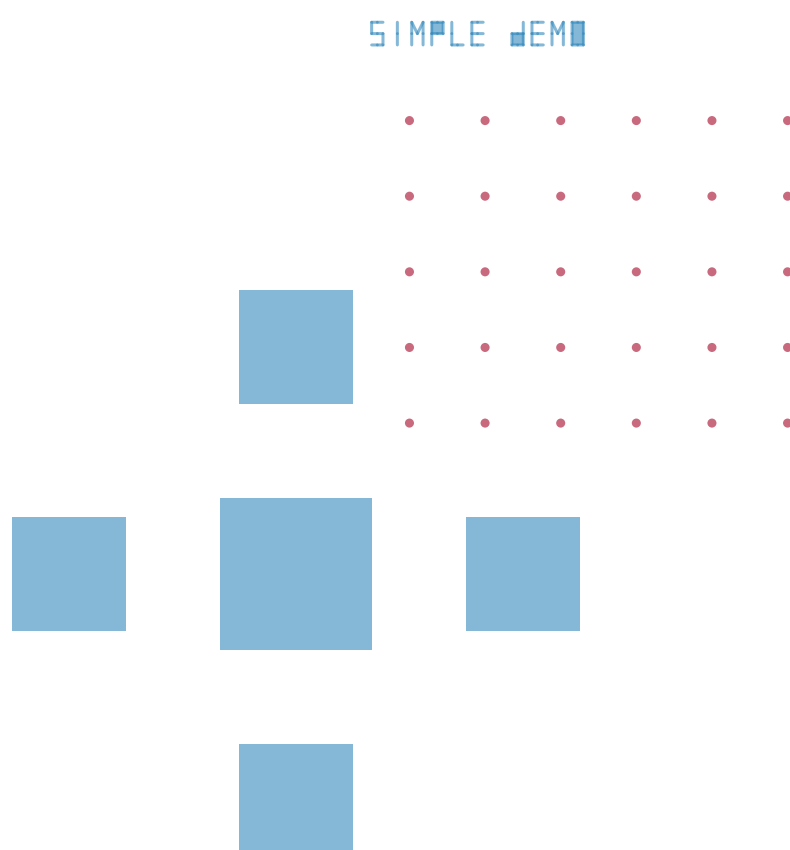


Figure 1: Simple Example: Generated Mask

3 Class Reference

3.1 TXLWriter

3.1.1 Methods

as

3.1.2 Properties

3.2 Strukturen

3.2.1 DataStructure

Die DataStructure ist wie folgt definiert und gilt jeweils für eine Datenbank-Tabelle:

- **Control** Array mit Informationen zur Datentabellen-Kontrolle.
 - **Table*** **String** Tabellen-Name
 - **IndexColumn*** **String** Name der Index-Spalte. Standardwert: "ID"
 - **ModifiedTSColumn*** **String** Name der Spalte mit Änderungs-Timestamp. Standardwert: "ModifiedTS"
 - **CategoryColumn*** **String** Name der Kategorien-Spalte. Falls angegeben (muss vom **Type Module** mit **DataCardinality** "1:n" sein), kann nach dieser Spalte kategorisiert werden. (z.B. select-Feld, etc.) Standardwert: "ID"
 - **NoUserInput*** **Bool** Ob kein User-Input in die Daten gelangen darf. Standardwert: **false**
 - **ReadOnly*** **Bool** Ob Datenbank-Tabelle schreibgeschützt ist. Standardwert: **false**
 - **NoPurge*** **Bool** Ob gelöschte Einträge in Datenbank-Tabelle nicht gelöscht werden sollen. Standardwert: **false**
 - **DBConnectionID*** **String** Name der Datenbankverbindung. Standardwert: "ClusterNetAppDB"
- **Columns** Array mit Definitionen der einzelnen Daten-Spalten. **Key** ist der Spalten-Name und **Value** ein **Array** mit der Definition:
 - **Type** **String** Typ der Spalte. Kann folgende Werte annehmen:
 - * **Integer** Ganzzahl
 - * **String** Text
 - * **Float** Gleitkommazahl
 - * **Checkbox** Checkbox
 - * **Date** Datum
 - * **Time** Zeit
 - * **DateTime** Datum und Uhrzeit
 - * **Array** Array, wird als JSON gespeichert.
 - * **Module** Fremdschlüssel (ID, Referenz) zu einem Eintrag in einem anderen Modul.

- * **FileResource** Dateien
- * **Option** Mehrfachauswahl

Wird **Type** nicht angegeben, so wird in `$GLOBALS['DefaultFieldStructures']` nach einem Eintrag gesucht. In diesem Fall kann lediglich der **DefaultValue** erhalten bleiben.

- **Validation*** **Array** mit Validierungs-Optionen.
 - * **Required*** **Bool** Ob Pflichtfeld. Standardwert: **false**
- **LabelKey*** **String** Key für Language Label. Standardwert: ""
- **TypeSpecificConfiguration*** **Array** spezifische Konfiguration (abhängig von **Type**)
 - Type* "Module"
 - * **ModulePath** **String** Pfad des Moduls
 - * **MModulePath*** **String** Pfad des M-Moduls der n:m-Relation (benötigt für **DataCardinality** "n:m")
 - * **DataCardinality*** **String** Kardinalität. Erlaubt: "1:n", "n:m", Standardwert: "1:n"
 - * **NToMTable*** **String** Tabelle für n:m-Relation. (benötigt für **DataCardinality** "n:m")
 - * **NForeignKeyColumn*** **String** Foreign Key Spalte n:m-Relation (n-Tabelle). (benötigt für **DataCardinality** "n:m")
 - * **MForeignKeyColumn*** **String** Foreign Key Spalte für n:m-Relation. (m-Tabelle) (benötigt für **DataCardinality** "n:m")
 - * **ModuleInstance*** **Reference** Referenz zu einer Modul-Instanz. Falls leer, wird diese anhand des **ModulePath** erzeugt.
 - * **MModuleInstance*** **Reference** Referenz zu einer M-Modul-Instanz. Falls leer, wird diese anhand des **ModulePath** erzeugt. (für **DataCardinality** "n:m")
 - * **ReplicateToJS*** **Bool** ob Javascript-Modul-Instanz erstellt werden soll. Standardwert: **false**
 - * **OverrideInstanceProperties*** **Array** Eigenschaften, die bei Instanziierung des Objekts gesetzt werden sollen. Standardwert: leeres **Array**
 - * **OverrideForeignInstanceProperties*** **Array** Eigenschaften, die bei Instanziierung des M-Objekts gesetzt werden sollen. (für **DataCardinality** "n:m") Standardwert: leeres **Array**
 - * **DetailViewType*** **String** Anzeigemodus in der Detailansicht. (für **DataCardinality** "n:m"). Erlaubte Werte: "Full", "Minimal" Standardwert: "Minimal"
 - * **OptionListAdditionalColumns*** **Array** Spalten, die beim Select-option zusätzlich im Frontend vorhanden sein sollen (für **DataCardinality** "1:n"). Standardwert: leeres **Array**
 - * **RequirePermission** **Bool** ob relationaler Eintrag gelesen werden können muss, um Datensatz zu lesen. Standardwert: **false**

Type "Date"

- * **AddCurrentTimeToInputDate*** Bool ob beim Parsen von User-Input die aktuelle Uhrzeit mitgespeichert werden soll. Standardwert: **false**

Type “String”

- * **EncryptDBContent*** Bool Ob Wert in der Datenbank verschlüsselt werden soll. Standardwert: **false**
- * **RichText*** Bool Ob Rich Text editing & anzeige aktiviert ist. Standardwert: **false**
- * **AdditionalAllowedHTMLTags*** String Weitere erlaubte Tags. Standardwert: “”

Type “FileResource”

- * **IsCollection*** Bool Ob mehrere Dateien. Standardwert: **false**

Type “Integer”

- * **RoundingFactor*** Float Rundungsfaktor. Standardwert: 1

Type “Float”

- * **RoundingFactor*** Float Rundungsfaktor. Standardwert: 1

Type “Checkbox”

- * **YesNoLabelKeys*** Array spezielle Yes/No Label Keys. Standardwert: leeres Array

Type “Option”

- * **Options** Array Optionen, jedes Element ein Array (ID =_i 'ID', LabelKey =_i 'LabelKey')
- **Visible*** Bool sichtbar? Standardwert: **true**
- **DefaultValue*** String Initialwert. Standardwert: **Type**-abhängigen Standardwert gesetzt.
- **Update*** Bool ob Feld bei Update in DB geschrieben wird. Standardwert: **true**
- **Insert*** Bool ob Feld bei Insert in DB geschrieben wird. Standardwert: **true**
- **NoUserInput*** Bool Ob Feld keine Benutzer-Eingaben erhalten darf. Standardwert: **false**
- **DisplayOptions*** Anzeigeoptionen. Wird an Form übergeben. Standardwert: leeres Array
 - * **IsPassword*** Bool ob Textfeld als Password dargestellt werden soll. Standardwert: **false**
 - * **IsCurrency*** Bool ob Textfeld als Währung dargestellt werden soll. Bsp: 2'700.35 Standardwert: **false**
 - * **IsLanguageLabelKey*** Bool ob es ein Language Label Key ist. (für **type String**) Standardwert: **false**
 - * **Multiline*** Bool ob mehrzeiliges Textfeld (nur für **String**) Standardwert: **false**
 - * **FormRowCSSClasses*** String zusätzliche Klassen für Formular-Zeile. Standardwert: “”

- * **CSSClasses*** **String** zusätzliche Klassen für Formular-Feld. Standardwert: ""
- * **LinkURLs*** **Bool** ob URLs im Inhalt gesucht und verlinkt werden sollen (nur für **String**) Standardwert: **false**
- * **DateFormat*** **String** Datumsformat für "Date"-Feld. Standardwert: "d.m.Y"
- **FormFieldParameters*** **Array** Parameter, die an das Formular-Objekt übergeben werden. Standardwert: leeres **Array**
 - * **IsTabFocusField*** **Bool** ob Textfeld im aktuellen Tab standardmässig fokussiert sein soll. Standardwert: **false**
- **SessionPersistent*** **Bool** Ob Wert der Variable Session-persistent gespeichert werden soll. Standardwert: **false**
- **UserPersistent*** **Bool** Ob Wert der Variable User-persistent gespeichert werden soll. Standardwert: **false**

3.2.2 FieldStructure

3.2.3 PageStructure

3.2.4 FormFieldStructure

4 **TXLConverter**

Appendices

A TXLWizard: Advanced Example

The following code demonstrates the usage of “TXLWizard” in a more advanced way, including labelling of array objects, nested referencing, etc. The generated mask is shown in Figure 2.

```

1 #####
2 # Import Libraries #
3 #####
4
5 # Import TXLWriter, the main class for generating TXL Output
6 import TXLWizard.TXLWriter
7
8 # Import Pre-Defined Shapes / Structures wrapped in functions
9 import TXLWizard.ShapeLibrary.EndpointDetectionWindows
10 import TXLWizard.ShapeLibrary.Markers
11 import TXLWizard.ShapeLibrary.Label
12 import TXLWizard.ShapeLibrary.CornerCube
13
14 # Import math module for calculations
15 import math
16
17
18 #####
19 # Sample / Structure Parameters #
20 #####
21
22 # Define all sample parameters
23 SampleParameters = {
24     'Width': 8e3,
25     'Height': 8e3,
26     'Label': 'GOI Demo CornerCube',
27 }
28
29 # Define all structure parameters
30 StructureParameters = {
31     'CornerCube': {
32         'BridgeLength': 8,
33         'ParabolaFocus': 9,
34         'XCutoff': 9,
35         'AirGapX': 3,
36         'AirGapY': 1,
37         'LabelXOffset': 0,
38         'LabelYOffset': 50,
39         'Label': 'R{:d}C{:d}',
40         'Layer': 2

```

```

41     },
42     'Circle': {
43         'Radius': 5,
44         'Layer': 3
45     },
46     'CornerCubeArray': {
47         'Columns': 6,
48         'Rows': 5,
49         'ArrayXOffset': 500,
50         'ArrayYOffset': -500,
51         'ArrayOrigin': [0.75e3, 3e3]
52     }
53 }
54
55
56 #####
57 # Initialize TXLWriter #
58 #####
59 TXLWriter = TXLWizard.TXLWriter.TXLWriter(
60     Width=SampleParameters['Width'],
61     Height=SampleParameters['Height']
62 )
63
64 #####
65 # Define Structures #
66 #####
67
68 ## Sample Label ##
69
70 # Give the sample a nice label...
71 SampleLabelObject = TXLWizard.ShapeLibrary.Label.GetLabel(
72     TXLWriter,
73     SampleParameters['Label'],
74     OriginPoint=[0.5e3, 1. * SampleParameters['Height'] / 2. -
75                 500],
76     FontSize=150,
77     StrokeWidth=20,
78     Layer=1
79 )
80 # ...and some other information
81 Alphabet = TXLWizard.ShapeLibrary.Label.GetLabel(
82     TXLWriter,
83     'abcdefghijklmnopqrstuvwxyz0123456789 megamega ggg ah
84     extraaaa rischaaar',
85     OriginPoint=[0.5e3, 1. * SampleParameters['Height'] / 2. -
86                 600],
87     FontSize=50,
88     ShowBar=False,

```

```

86     StrokeWidth=3,
87     Layer=1
88 )
89
90 ## Endpoint Detection ##
91
92 # Use Pre-Defined Endpoint Detection Windows
93 TXLWizard.ShapeLibrary.EndpointDetectionWindows.
    GetEndpointDetectionWindows(
94     TXLWriter, Layer=1)
95
96 ## Alignment Markers ##
97
98 # Use Pre-Defined Alignment Markers
99 TXLWizard.ShapeLibrary.Markers.GetMarkers(
100     TXLWriter, Layer=1)
101
102
103 ## User Structure: Corner Cube ##
104
105 # Create Definition Structure for Corner Cube that will be
reused
106 CornerCubeDefinition = TXLWizard.ShapeLibrary.CornerCube.
    GetCornerCube(
107     TXLWriter,
108     ParabolaFocus=StructureParameters[ 'CornerCube' ][ '
        ParabolaFocus' ],
109     XCutoff=StructureParameters[ 'CornerCube' ][ 'XCutoff' ],
110     AirGapX=StructureParameters[ 'CornerCube' ][ 'AirGapX' ],
111     AirGapY=StructureParameters[ 'CornerCube' ][ 'AirGapY' ],
112     Layer=StructureParameters[ 'CornerCube' ][ 'Layer' ]
113 )
114
115 # Create Definition Structure for combination of cornercube and
additional circle
116 FullCornerCubeNoRotation = TXLWriter.AddDefinitionStructure( '
    FullCornerCubeNoRotation' )
117 FullCornerCubeNoRotation.AddPattern( 'Reference',
118     ReferencedStructureID=CornerCubeDefinition.ID,
119     OriginPoint=[1. * StructureParameters[ 'CornerCube' ][ '
        BridgeLength' ] / 2., 0]
120 )
121 FullCornerCubeNoRotation.AddPattern( 'Circle',
122     Center=[0, 0],
123     Radius=StructureParameters[ 'Circle' ][ 'Radius' ],
124     Layer=StructureParameters[ 'Circle' ][ 'Layer' ]
125 )
126

```

```

127 # Create definition structure with rotation of entire referenced
    structure
128 FullCornerCube = TXLWriter.AddDefinitionStructure( '
    FullCornerCube',
129
    RotationAngle
    =45)
130 FullCornerCube.AddPattern( 'Reference',
131     ReferencedStructureID=FullCornerCubeNoRotation.ID,
132     OriginPoint=[0, 0]
133 )
134
135 # Create array of the definition structure above
136 CornerCubeArrayFine = TXLWriter.AddContentStructure( '
    CornerCubeArrayFine')
137 CornerCubeArrayFine.AddPattern( 'Array',
138     ReferencedStructureID=FullCornerCube.ID,
139     OriginPoint=StructureParameters[ 'CornerCubeArray' ][ '
    ArrayOrigin' ],
140     PositionDelta1=[
141         StructureParameters[ 'CornerCubeArray' ][ 'ArrayXOffset' ],
142         0
143     ],
144     PositionDelta2=[
145         0, StructureParameters[ 'CornerCubeArray' ][ 'ArrayYOffset' ]
146     ],
147     Repetitions1=StructureParameters[ 'CornerCubeArray' ][ 'Columns' ],
148     Repetitions2=StructureParameters[ 'CornerCubeArray' ][ 'Rows' ]
149 )
150
151 # Add Labels to each array element
152 for Row in range(1, StructureParameters[ 'CornerCubeArray' ][ 'Rows' ] + 1):
153     for Column in range(1, StructureParameters[ 'CornerCubeArray' ]
        ][ 'Columns' ] + 1):
154         RowColumnCountLabel = TXLWizard.ShapeLibrary.Label.
            GetLabel(
155             TXLWriter,
156             StructureParameters[ 'CornerCube' ][ 'Label' ].format(
                Row, Column),
157             OriginPoint=[
158                 StructureParameters[ 'CornerCubeArray' ][ '
                ArrayOrigin' ][0]
159                 + StructureParameters[ 'CornerCubeArray' ][ '
                ArrayXOffset' ]
160                 * (Column - 1) + StructureParameters[ 'CornerCube

```



```

161         '][ 'LabelXOffset' ],
162         StructureParameters[ 'CornerCubeArray' ][ '
163         ArrayOrigin' ][1]
164     + StructureParameters[ 'CornerCubeArray' ][ '
165         ArrayYOffset' ]
166     * (Row - 1) + StructureParameters[ 'CornerCube' ][
167         'LabelYOffset' ]],
168     FontSize=16,
169     StrokeWidth=3,
170     Layer=1,
171     RotationAngle=45
172 )
173 #####
174 # Generate Output Files #
175 #####
176 # Note: The suffix (.txl, .html, .svg) will be appended
177         automatically
178 TXLWriter.GenerateFiles( 'Masks/EM160509_GOI_Demo_CornerCube' )

```

Content/Example_GenerateMask.py

GO I DEMO CORNER
 ABCDEFGHIJKLMNOPQRSTUVWXYZ 123456789 MEGAMEGA GI

R1C1

R1C2

R1C3

R2C1

R2C2

R2C3

Figure 2: Advanced Example: Part of the Generated Mask