Research center on alpine environment
Centre de Recherche sur l'Environnement Alpin
Industrie 45 - CH 1951 SION +4127 324 03 80 crealp@crealp.vs.ch

Tutorial

How to create a ToolMap datamodel using TmDmCreator

Revised: March 4, 2013
Lucien Schreiber
lucien.schreiber@crealp.vs.ch

Contents

1 Introduction

This tutorial explains how to create a ToolMap project manually. This approach has the following advantages:

1. It ensures that the ID remain consistent
2. It could generate a multilingual model
3. It allows better monitoring of model changes

The main disadvantage of this approach is the lack of user interface as well as the need for the user to have some knowledge of SQL. Finally, this approach has been developed to meet the need for rigor in the management of the Swiss geological data model.

2 Conceptual Workflow

The diagram shown in figure 1 illustrates the proposed workflow. User edits the user_structure.sql and user_content.txt files. These files as well as base_structure.sql are used by the software TmDmCreator to produces either:

1. a SQL file defining the project (output 1)
2. a ToolMap project (output 2)

3 Data needed

For proper operation, TmDmCreator requires the following files:

base_structure.sql
: contains the necessary SQL code base for all ToolMap projects. This file should normally not be edited by users

user_structure.sql
: contains the SQL structure describing the layers attributes

user_content.txt
: Is a tabular file (editable in Excel for example) containing the definition of layers, objects, and attribute values.

The recommended way to work with user_structure.sql and user_content.txt is described below

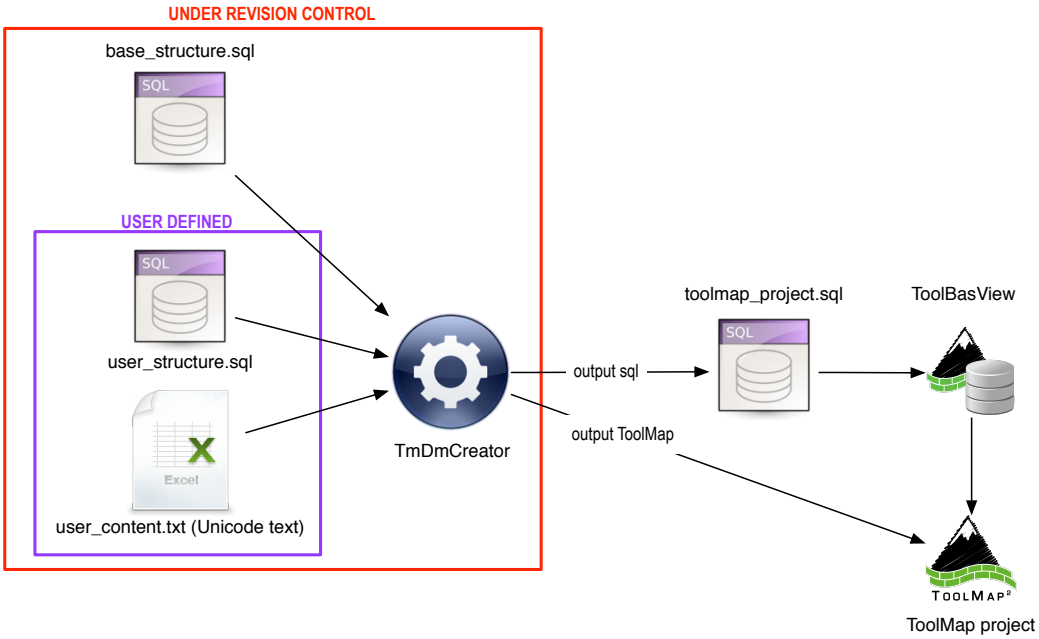

Figure 1: Conceptual workflow

4 Preparing user data

4.1 Layers

Open user_content.txt using a spreadsheet and edit the thematic_layers part. Each of the layers that we want to export should appear here. The structure is as follows (see figure 2):

LAYER_INDEX unique identifier of the layer
TYPE_CD layer spatial type as follow

- **0** = Line
- **1** = Point
- **2** = Polygon

LAYER_NAME the layer name. This name will be given to the SHP file when exporting

Make sure you choose an Unicode format (Unicode Text (*.txt) or UTF-16 Unicode Text (*.txt)) when saving from the spreadsheet.

	A	B	C
1	thematic_layers		
2	LAYER_INDEX	TYPE_CD	LAYER_NAME
3	1	1	Boreholes_PT
4	2	1	Water_Outflow_PT
5	3	1	Planar_Structures_PT
6	4	1	Folds_PT
7	5	0	Erosional_Structures_L
8	6	0	Instability_Structures_L
9	7	0	Tectonic_Boundaries_L
10	8	2	Unconsolidated_Deposits_PLG
11	9	2	Instability_Structures_PLG
12	10	2	Bedrock_PLG

Figure 2: List of layers as shown in user_content.txt

4.2 Objects

Edit the file user_content.txt to add objects. They must have the following structure (See figure: 3):

OBJECT_ID object unique ID.
OBJECT_CD object code, should not necessarily be unique
OBJECT_TYPE_CD object spatial type, uses same values as those described above for TYPE_CD in thematic_layers

THEMATIC_LAYERS_LAYER_INDEX the index of the layer that the object refers to. The value 1 shown in the example (Figure 3) therefore relates to the theme Boreholes_PT.

OBJECT_DESC_0,1,2,3,4,5 object description in up to 5 languages.

OBJECT_ISFREQ Set to 1 for frequent objects and 0 otherwise. This parameter is only taken into account for line type objects. Set to 0 for all point or polygon objects.

SYMBOL_CD leave empty

RANK leave empty

REMARK leave empty

4.3 Attributes structure

Edit the file user_structure.sql with Notepad (or even better with Notepad + +). For each layer containing attributes, there must be a SQL code of the type:

```
1 -- layer_at1 --
2 CREATE TABLE `layer_at1` (
3 `OBJECT_ID` int(10) unsigned NOT NULL,
4 -- add user attributes here --
5 PRIMARY KEY (`OBJECT_ID`),
6 KEY `LAYER_ATX_FKIndex1` (`OBJECT_ID`)
7 ) ENGINE=MyISAM DEFAULT CHARSET=utf8;
```

This code is the basic template for creating an attribute table. The number after layer_at (see line 2) indicates the layer index and refers to the LAYER_INDEX column in user_content.txt. In our example layer_at1 describe the attributes for the layer Boreholes_PT. User attributes can then be added on line 4 of this template. Below are described the five attributes that can be used in a ToolMap data model as well as the corresponding SQL code

4.3.1 Enumeration

```
1 `D_C_UNDERG` int(11) DEFAULT NULL COMMENT 'ENUMERATION',
```

If you add such fields, then you must also fill the list of supported values (see section 4.4).

4.3.2 Text

```
1 `DESCRIPT` varchar(255) DEFAULT NULL,
```

14	dmn_layer_object														
15	OBJECT_ID	OBJECT_CD	OBJECT_TYPE_CD		THEMATIC_LAYERS_LAYER_INDEX	OBJECT_DESC_0	OBJECT_DESC_1	OBJECT_DESC_2	OBJECT_DESC_3	OBJECT_DESC_4	OBJECT_ISFREQ	SYMBOL_CD	RANK	REMARK	
16	1	774		1	1	forage	drill	bohrung			0				

Figure 3: Objects structure as described in user_content.txt

The number next to the keyword varchar indicates the maximum text length.

4.3.3 Integer

```
1 `NUM_REF` int(11) DEFAULT NULL,
```

There is no special option for integer fields

4.3.4 Float

```
1 `TEMP` decimal(5,2) DEFAULT NULL,
```

The two digits next to the keyword decimal indicate the field precision and scale. In this example, 5 is the precision and 2 is the scale. The precision represents the number of significant digits that are stored for values, and the scale represents the number of digits that can be stored following the decimal point. In this case, values that can be stored range from -999.99 to 999.99.

4.3.5 Date

```
1 `REF_DATE` date DEFAULT NULL,
```

There is no special option for date fields

4.4 Attributes values

For each enumeration field previously added in the user_structure.sql file, it is necessary to define the allowed values. Therefore it is necessary to edit the attributes section of user_content.txt. The structure of the attributes section is shown in Figure 4. This table is divided into two parts, the first three columns describe the attribute fields, the remaining columns describe the values supported by these fields. Below is a description of each column.

ATTRIBUT_ID attribute unique ID.

LAYER_INDEX the index of the layer that the attribute refers to. The value 1 shown in the example (Figure 4, row 42 and 43) therefore relates to the theme Boreholes_PT.

ATTRIBUT_NAME attribute name. This name will be used as the column name in the exported SHP. Some limitations apply to SHP format for column names, for more information you can refer to `http://en.wikipedia.org/wiki/Shapefile#Shapefile_attribute_format_.28.dbf.29` or `http://www.gdal.org/ogr/drv_shapefile.html`

CATALOG_ID attribute value unique ID
CODE attribute value code, should not necessarily be unique
DESCRIPTION_0,1,2,3,4,5 attribute value description in up to 5 languages. The order of language is not important, but it must be identical to the one chosen for the objects (see 4.2).

40	attributs									
41	ATTRIBUT_ID	LAYER_INDEX	ATTRIBUT_NAME	CATALOG_ID	CODE	DESCRIPTION_0	DESCRIPTION_1	DESCRIPTION_2	DESCRIPTION_3	DESCRIPTION_4
42	1	1	D_C_UNDERG	1	1	oui	yes	ja		
43	1	1	D_C_UNDERG	2	2	non	no	nein		
44	2	1	MAIN_TARG	3	1	-	...	...		
45	2	1	MAIN_TARG	4	2	geothermique	...	...		
46	3	2	STATUS	5	1	captée				
47	3	2	STATUS	6	2	non captée				
48	4	2	TYPE	7	1	-				
49	4	2	TYPE	8	2	thermale				
50	5	3	PHASE	9	1	1ère phase				
51	5	3	PHASE	10	2	2ème phase				
52	6	4	PHASE	11	1	3ème phase				
53	7	7	LIM_TECT_U	12	1	non				
54	7	7	LIM_TECT_U	13	2	oui				
55	8	7	STATUS	14	1	probable				
56	8	7	STATUS	15	2	certain				
57	9	7	ACTIVITY	16	1	indefini				
58	9	7	ACTIVITY	17	2	inactif				
59	9	7	ACTIVITY	18	3	actif				
60	10	9	STATUS	19	1	actif				
61	10	9	STATUS	20	2	inactif				

Figure 4: Attributes section structure

5 Running TmDmCreator

TmDmCreator is a command-line utility. As an input, it takes the 3 files described in details above (base structure.sql, user_structure.sql, user_content.txt) and produces a resulting SQL file. Its behavior may be controlled using different parameters described bellow (see figure 5).

5.1 Optional parameters

–**verbose** Be more verbose, specifically when an error occur.
–**toolmap** Write output directly into a ToolMap project instead of a SQL file (not implemented actually).
–**overwrite** when specified, the output file will be erased if existing.
–**language** =<**num**> specify the language column to use. Column numbering starts at 0. This option allows multilingual support.

5.2 Mandatory parameters

base structure sql file base_structure.sql file name
user structure sql file user_structure file name

user content txt file user_content.txt file name

result file either a sql file name for SQL output or a directory name for a ToolMap output.

5.3 Sample

A typical TmDmCreator command like will look like the following:

```
1 TmDmCreator --verbose --language=0 base_structure.sql
    user_structure.sql user_content.txt result.sql
```

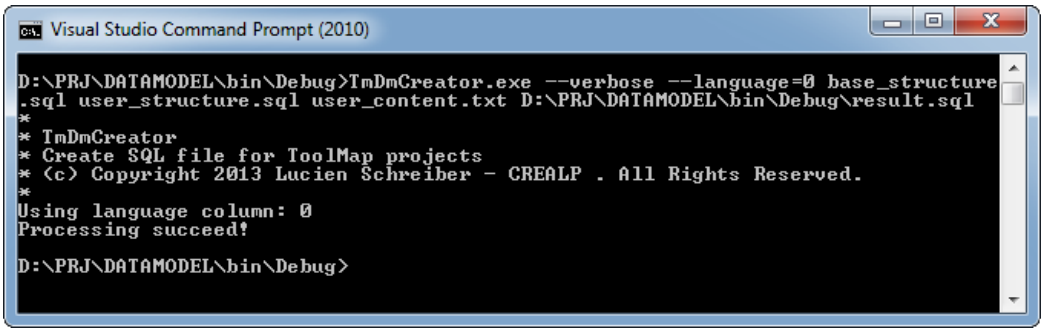

Figure 5: Command line output from TmDmCreator