



# The Flood Defense Toolkit Method Manual

# Author

- Angiola Fanelli – Thetis S.p.a. – [angiola.fanelli@thetis.it](mailto:angiola.fanelli@thetis.it)

## Summary

<b>1 Introduction</b> .....	4
<b>2 Plugin strategies and settings</b> .....	5
2.1 Necessary and optional data for the plugin .....	6
2.2 Data input into the plug-in .....	6
<b>3 The plug-in commands</b> .....	9
3.1 EP Elevated Perimeter .....	9
3.2 EA Elevated Area.....	12
3.3 WR – Water Receiving bodies .....	13
3.4 WDS - Water Discharge System (WD) .....	16
3.5 DS – Creation of shut-off valves.....	17
3.6 Report generation and query of spatial elements .....	18
<b>4 Customizable libraries</b> .....	20

# 1 Introduction

The Flood Defense Toolkit Method (FDTM) is a plug-in developed for Qgis (version 2.18.13) and represents a planning tool for urban spaces providing a general assessment on the feasibility of a flood protection strategy in an urban area.

The plug-in includes various types of technological solutions and their mutual combinations, which are site specific.

After the installation, the plug-in is presented with a command bar which shows all the functions provided and described below (Figure 1-1).

The plug-in is downloadable from the Qgis official repository version 2.18 and it is also available on Github at: <https://github.com/angiolafanelli/FloodDefense>.

To install the plug-in, simply select from the repository the Flood Defense plugin and press "install". After the installation, the manual and the libraries settings files are saved in the following path:

"C: \ Users \ username \ .qgis2 \ python \ plugins \ fdtm"



**Figure 1-1 Plug-in command bar before the settings.**

In the stand-alone versions of Qgis 2.18 for the Windows application system, the file named "qgis\_customwidgets.py" is not copied to the correct path of python. If the plug-in does not find this file, an error will occur. To solve the problem, simply copy the "qgis\_customwidgets.py" file into the folder:

[QGIS installation folder]: QGIS installation folder QGIS]\apps\python27\lib\site-packages\PyQt4\ui\widget-plugins\qgis\_customwidgets.py

The file "qgis\_customwidgets.py" is delivered together with the folder containing the installation library Flood Defense Toolkit Method (fdtm).

# 2 Plug-in strategies and settings

The user can choose one of the following flood protection strategies available in the plug-in:

1. raise the height of the perimeter which contains an area of particular interest (Elevated Perimeter - EP button);
2. raise an area of a new lot with a specific height in order to secure it from a flooding event (Elevated Area - EA button);
3. outline flooding areas or build tanks for the storage of the collected waters in the EP and EA areas previously designed (Water Receptors - WR key). The plug-in also includes the possibility of setting the green roofs over the buildings inside the EP perimeter;
4. Design and setting different types of pipes to transfer water from EP and EA to receiving bodies WR (WD key);
5. manage the different interconnection valves of the draining system (DS button).

For each of the strategies listed above, the user can choose among different technical measures listed in the plug-in libraries. Within the plug-in are available customizable libraries, so that the user can choose the more suitable choices in order to protect a specific area. For example, each segment of an elevated protection perimeter can be further elevated by the planner through the use of different technologies: sandbags, concrete walls, inflatable dams, plastic dams etc.

At the end of the process for the identification of the areas that need protection, the plug-in shows the whole map of the chosen works and some synthesis tables, which show also the total costs of the protection measures implemented. The management of the prices is customizable as well with specific libraries of protection measures.

## 2.1 Necessary and optional data for the plugin

The plug-in requires some cartographic data; some are necessary, some are, instead, optional.

The necessary data are:

- DTM (Digital Terrain Model) for altitudes (meters): the higher the resolution of this file the better will be the plug-in results;
- Buildings (Buildings layer): polygonal shapefile of the buildings;
- Roads layer: linear shapefile of the roads;
- Rainfall of the considered event - average precipitation (in millimeters): this data has to be entered in the beginning setting mask.

The optional data managed by the plug-in are:

- Land Cover Use layer: polygonal shapefile which describes the characteristics of the land cover and the run-off coefficients associated with the different kind of soils;
- Polygonal shapefile containing flooded area;
- Drainage System Layer: linear shapefile that describes existing sewers and water drainage systems in the area;
- Railways (Railways layer): linear shapefile of the railways.

The plug-in needs that all the above listed data have the same geographical reference system (EPSG).

## 2.2 Data input into the plug-in

At the end of the shapefiles loading in the QGIS project, the user must save the project with a name and then set up the plug-in setting window by activating the button shown in Figure 2-1.



Figure 2-1 starting button for the plug-in setting phase

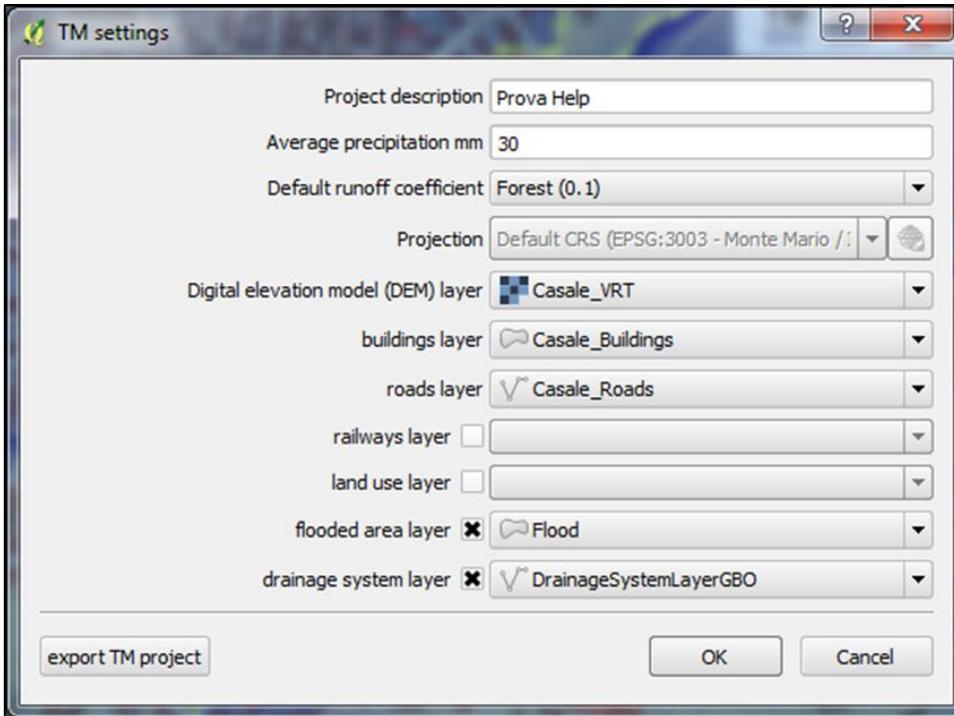


Figure 2-2 Plug-in setting window.

Figure 2-2 Plug-in setting window.

Figure 2-2 shows the setting window. In this windows the user sets:

1. Average rainfall (mm): information needed to calculate the amount of Cumulated Water (CW - in  $m^3$ ) during the event within protected perimeters and elevated areas;
2. all the mandatory and optional shapefiles;
3. The "default run-off coefficient" which will be used as initial run-off coefficient, in order to calculate the CW.

The run-off coefficient is a non-dimensional parameter correlated to the amount of water that flows out compared to the total amount received by the precipitation. Its value, between 1 and 0, is greater (close to 1) when the area has low soil infiltration and high runoff (waterproof flooring such as asphalt) and lower (close to 0) for permeable and well-vegetated areas (forest, public green).

At the end of the setting phase, all the shapefiles necessary for the plug-in are automatically generated and loaded into the QGIS project. The panel (Figure 2-3) shows the list of shapefiles that will later be populated by the individual protection measures designed by the user.

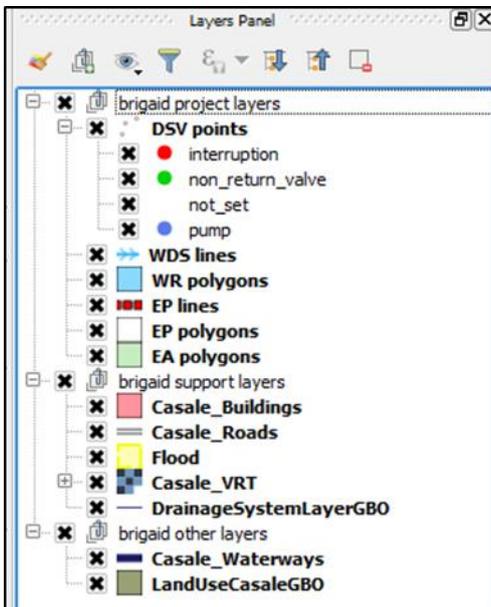


Figure 2-3 List of layers managed by the Plug-in.

Now all the plugin control buttons are activated and appear colored (Figure 2-4).



Figure 2-4 Plug-in bar after the conclusion of the input data validation.

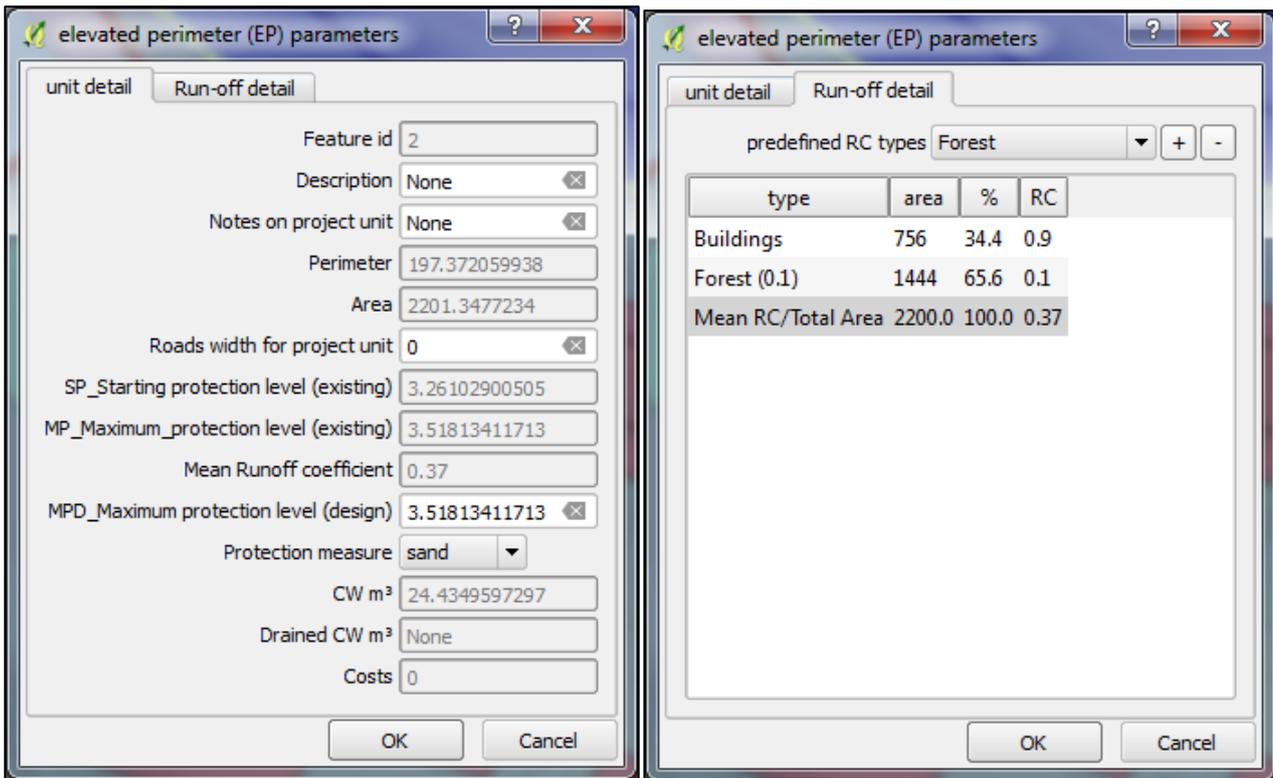
# 3 The plug-in commands

## 3.1 EP Elevated Perimeter

The first and simplest protection strategy of an area is represented by the creation of different kind of barriers along its perimeter which, therefore, will be higher compared to the protected area level. This strategy can be implemented within the plug-in using the EP button. By activating the EP command and then through the editing button drawing a polygonal feature, the user can identify an area of particular interest to be protected from flooding events. Once the area has been delimited, and thus the polygon feature is closed, the dialog box shown in Figure 3-1 will open. This window summarizes:

1. Geometric, altimetry and soil characteristics of the protected area;
2. Calculation of the water volumes to be managed;
- 3 Choice of the safeguard quota (MPD, in meters);
4. Technologies for the protection of that specific perimeter (there is a first default technology that can be modified later for each segment of the perimeter).

The Figure 3-1 shows the two setting windows of the EP characteristics. The user in this window is able to define all the geometric characteristics of the EP polygon just designed (area and perimeter). The window called "unit detail" also displays the minimum (SP) and maximum (MP) of the designed perimeter; in relation to these parameters the user will have to set the maximum protection quota (MPD). Furthermore, if there are roads that intersect the perimeter, the user can define their width (roads width): this data will be used to calculate the total run-off coefficient of the area.



**Figure 3-1 Windows for the definition of the characteristics of the EP areas.**

In the second summary window for the EP parameters (“Run-off detail”) the soil coverage percentages (specific for that area) are automatically calculated (if the land use layer is set), or manually entered by the user. With this information, the plug-in returns the value of the average Run-off coefficient of the area and consequently the system is able to calculate the volume of water accumulated (CW) during the event.

An example of the structure and a description of the coding fields for the shapefile “land use cover” is summarized in Table 3-1.

**Table 3-1 Structure of the Land use cover shapefile.**

FID	Shape	Area	Descr_RC	RC
6037	Polygon	2448	Private gardens, low residential	0.4
6038	Polygon	2600	Private gardens, low residential	0.4
6039	Polygon	3928	Private gardens, low residential	0.4
6040	Polygon	3020	Private gardens, low residential	0.4
6041	Polygon	204	Private gardens, low residential	0.4
6042	Polygon	6294	Private gardens, low residential	0.4
6043	Polygon	5763	Private gardens, low residential	0.4
5811	Polygon	2247	Paved urban areas, flat roofs	0.8
5812	Polygon	2632	Paved urban areas, flat roofs	0.8
5813	Polygon	9268	Paved urban areas, flat roofs	0.8
5814	Polygon	14739	Paved urban areas, flat roofs	0.8

The codification of the land use field (Desc\_RC) must respect the classification shown in Table 3 2.

**Table 3-2 Soil Coverage Classification and related Run off Coefficient (RC) value.**

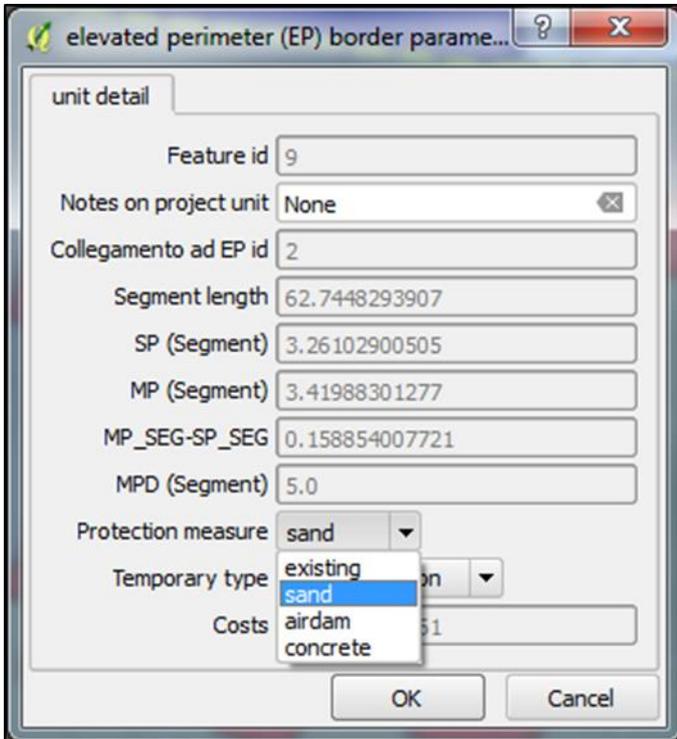
Descr_RC	RC
Forest	0.1
Grass, parks, cemeteries	0.2
Cultivated areas, turf	0.3
Private gardens, low residential	0.4
Bare soil, gravel roadways	0.5
Loam, macadam	0.6
Industrial, clay	0.7
Paved urban areas, flat roofs	0.8
Roads, Highways, pitched roofs	0.9
Completely impervious surfaces	1

Once all the parameters have been set in these two summary windows, the data will be saved within the shapefile summary table (EP\_poligon, Figure 3-3). The types (TYP) and costs (COST) related to the default technology that can be modified in the next phase, are also displayed.

DESC	NOTES	AREA	PER	SP	MP	MP_USER	ROADS_W	CW	CW_DRAIN	RC	RC_DEF	RC_USER
Nartece	None	6002	340	60	125	125	0	151	151	0.9	{"Buildings":[90,3633],"Roads, pitched roofs":[90,2368]}	0.9

**Figure 3-2 Elevated Perimeter layer attribute table**

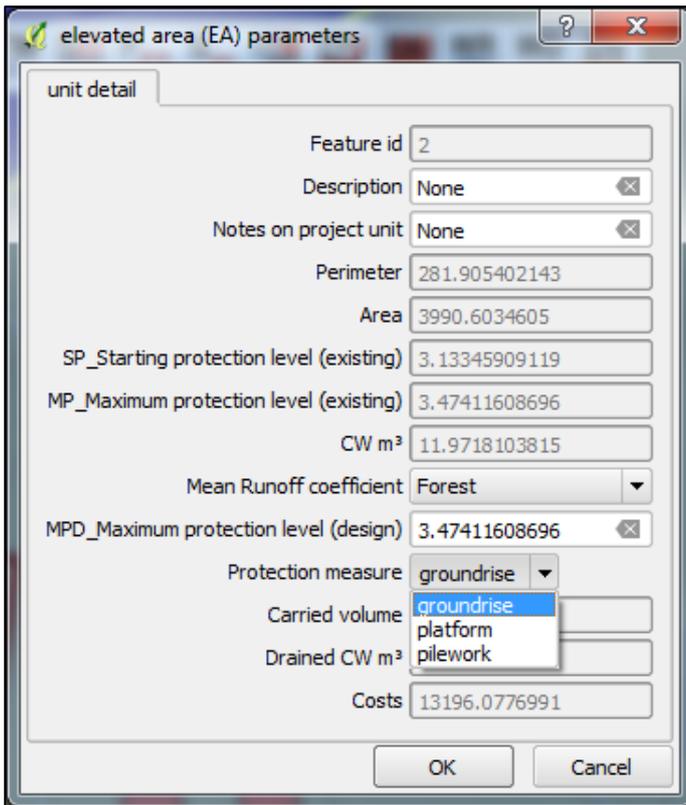
Subsequently, the user can set the technological solutions adopted for raising each segment's height along the perimeter (Figure 3-3). The user can choose through a drop-down menu "protection measure" one of the technologies within the plugin's Elevated Perimeter library.



**Figure 3-3 Window to define the protection measure adopted in a segment (EP\_lines) of the EP perimeter. For each segment of the perimeter, the user can choose a technology to be applied.**

## 3.2 EA Elevated Area

The plug-in allows also the identification of new urban expansion areas that can be imagined at a higher height (Elevated Area) in order to be preserved from flooding events. The command that activates this feature is called EA: when activated, new urban expansion polygons can be defined and set at a higher elevation than the ground level. EA polygons cannot contain houses and streets, because they are considered areas in new urban expansion. Once the EA polygon is closed, the summary window of the just drawn element will open (Figure 3-4). As seen before (EP case), the user must set the maximum height of the Elevated Area (EA) and, through a drop-down menu, choose the best technology protection for the area (Figure 3-4).



**Figure 3-4 Window to define the protection measure adopted in an elevated area (EA).**

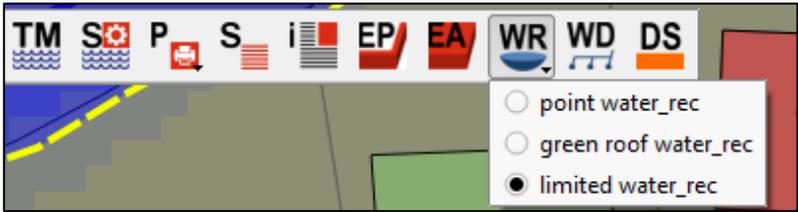
For EA polygons, a single coefficient run-off value can be set and the plug-in is able to calculate the water accumulated inside the polygon after the considered rain event.

### 3.3 WR – Water Receiving bodies

Once the perimeters and the areas are made safe, the user is called to evaluate how to dispose of the water masses accumulated in these areas. It is possible to plan some different solutions:

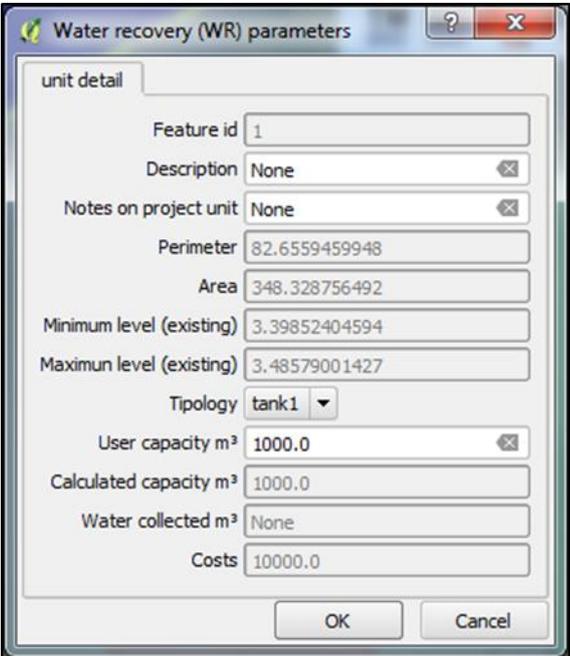
- Tank's storage;
- increase water soil retention though solutions that aim at the run-off coefficient decrease (green roof for example);
- The identification of "expansion areas" (green areas, basins, floodable parks) for the water storage.

In this phase it is necessary to activate the WR (Water Receptor) key which presents the three above options as shown in Figure 3-5.



**Figure 3-5 Three possible options for the Water Receptor**

When the first item called "point water\_rec" is active the user can design the tank for the water. After the button activation, the user can start drawing the tank with the command "edit polygon feature". There can be different types of cisterns that can be implemented by the plug-in (external, underground, raised etc.) and, again, there is a library that manages the description and the related costs for each cubic meter of water that can be accumulated. The Figure 3-6 shows the window for the above listed tanks' characteristics.



**Figure 3-6 Window for setting the tanks' characteristics.**

Another option for storing the water are the green roofs. When the "green roof water\_rec" button is active (Figure 3-6), the user will be able to draw them on the roof of the buildings inside the EP areas. Also in this case the user can choose from several available technologies listed in the customizable libraries. The Figure 3-7 shows the screen for the identification of the green roofs' characteristics.

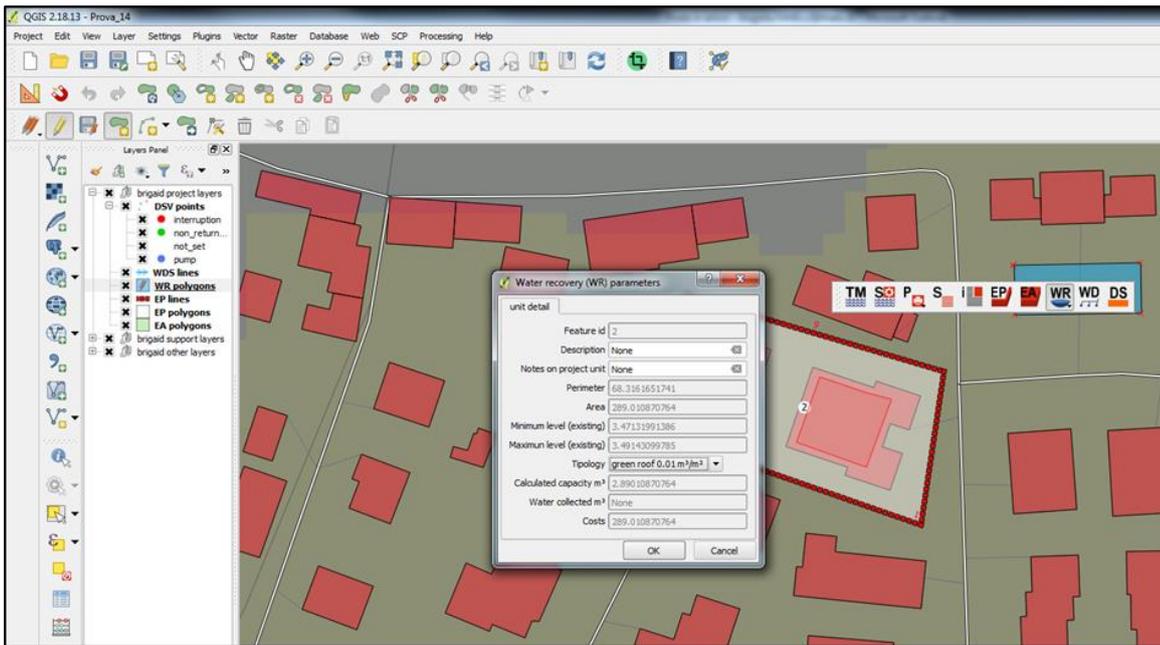


Figure 3-7 Window for setting the characteristics of the green roofs.

The last type of button is "limited water\_rec": the user outlines floodable areas (obviously without any type of buildings) and the plug-in calculates, considering the morphological characteristics of the identified area, how much water can be accumulated inside them. Another library manages the options that the planner can choose: floodable basin, floodable park and floodable garden (see Figure 3-8).

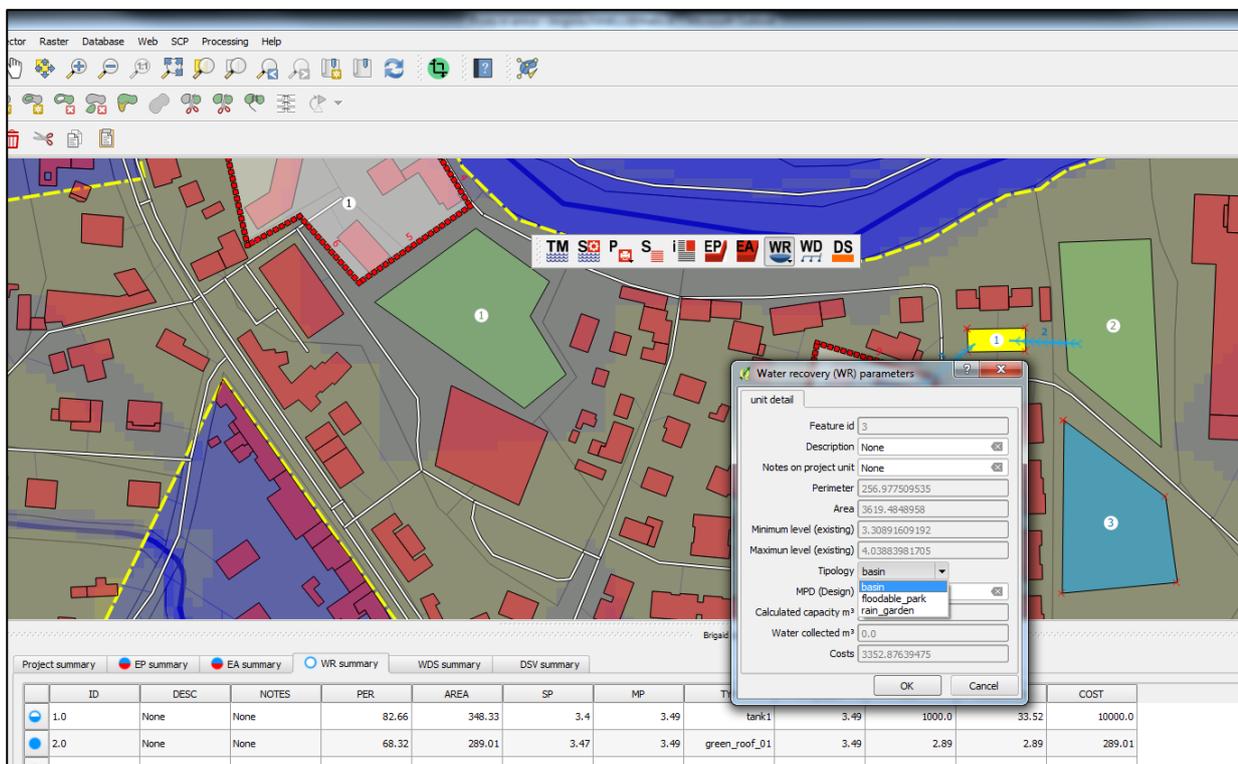
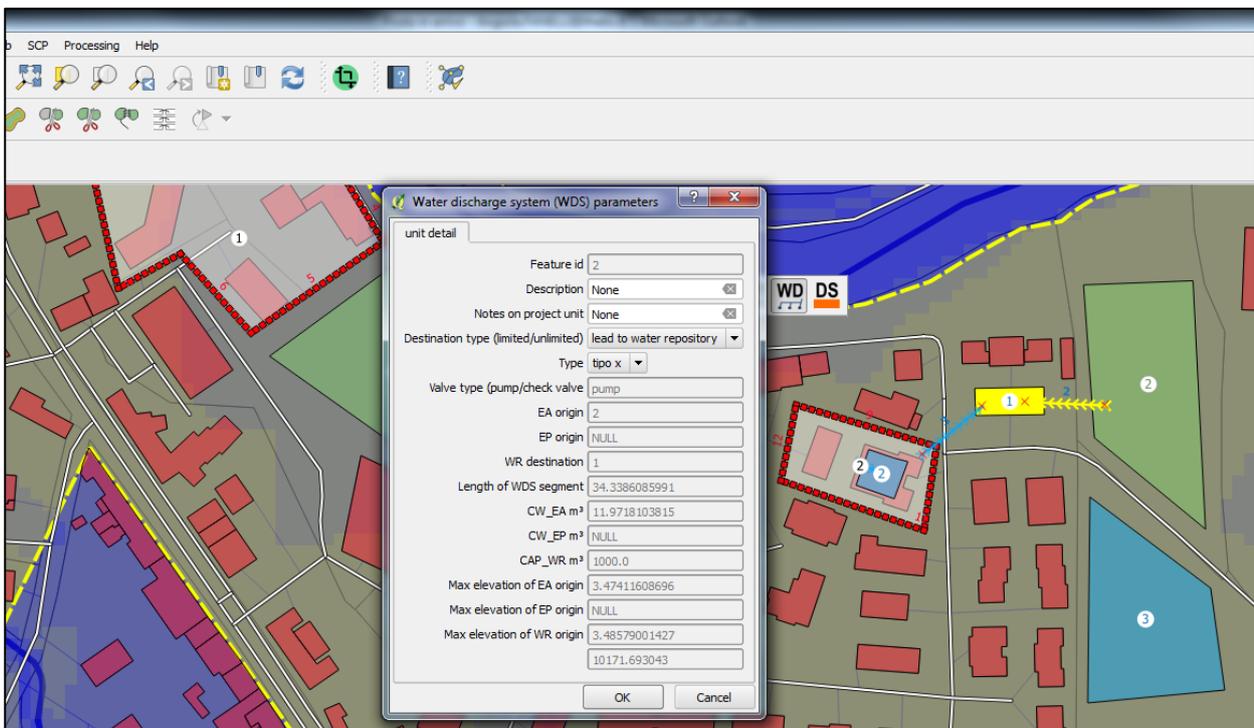


Figure 3-8 Window for defining options for "limited water\_rec".

### 3.4 WDS - Water Discharge System (WD)

Once the receiving water areas are defined, it is necessary to create connections between the EP or EA and the drainage areas. The WD (Water Discharge System) button allows to edit the linear shapefile of the WDS: the user designs where to drain the accumulated water. Once this command is activated, the user will be able to draw the linear path from an EA (or EP) area to a WR perimeter (defined in the previous step) and the plug-in returns the calculations of the water balance by "virtually" moving the masses of water to the receptor bodies. Figure 3-9 shows the screenshot for the characteristics of the connection pipes in order to drain the water.



**Figure 3-9 Window for defining the characteristics of the connecting pipes to move the masses of water.**

If there is a river or another water receptor (e.g. the sea) characterized by a significant capacity to receive water volumes, the user can draw a WD line that ends in this kind of receiving body, without the need to draw the receiving WR (Figure 3-10).

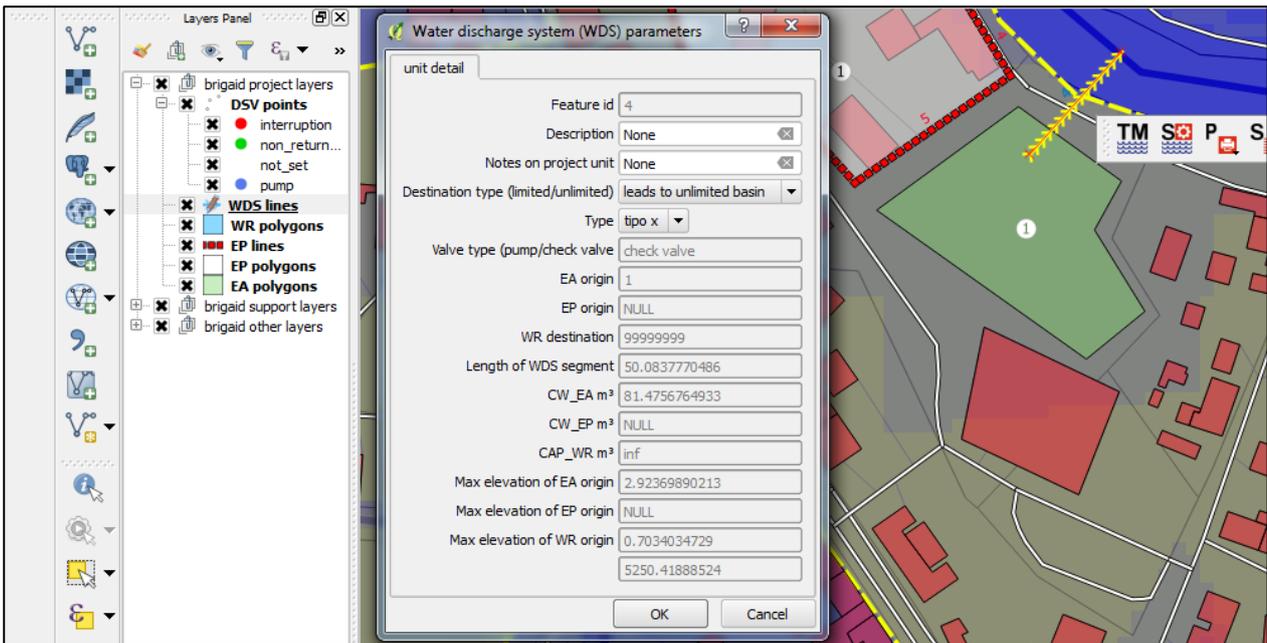


Figure 3-10 Setting up a receiving body with unlimited capacity

### 3.5 DS – Creation of shut-off valves

Once the design of the protection zones has been completed, if the shapefile of the sewage drains is available, it is possible to start the automatic fill with the shut-off valves; these valves will be automatically positioned in the intersection points among the perimeter EP, the sewers and the existing drainage system (Figure 3-11).

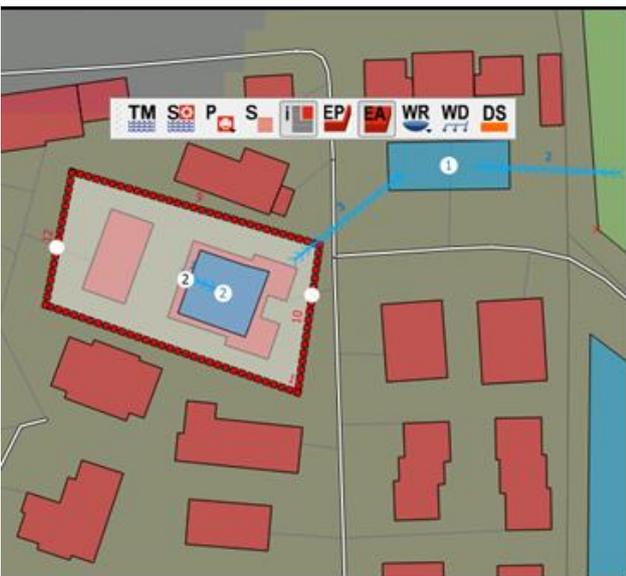


Figure 3-11 Creation of shut-off valves launched with the DS command.

The user can then set the different types of valves by choosing them from the options of the DSV point shapefile (Figure 3-12).

The types of valves that can be set are:

- Interruption valve;
- Non return valve;
- Pump.

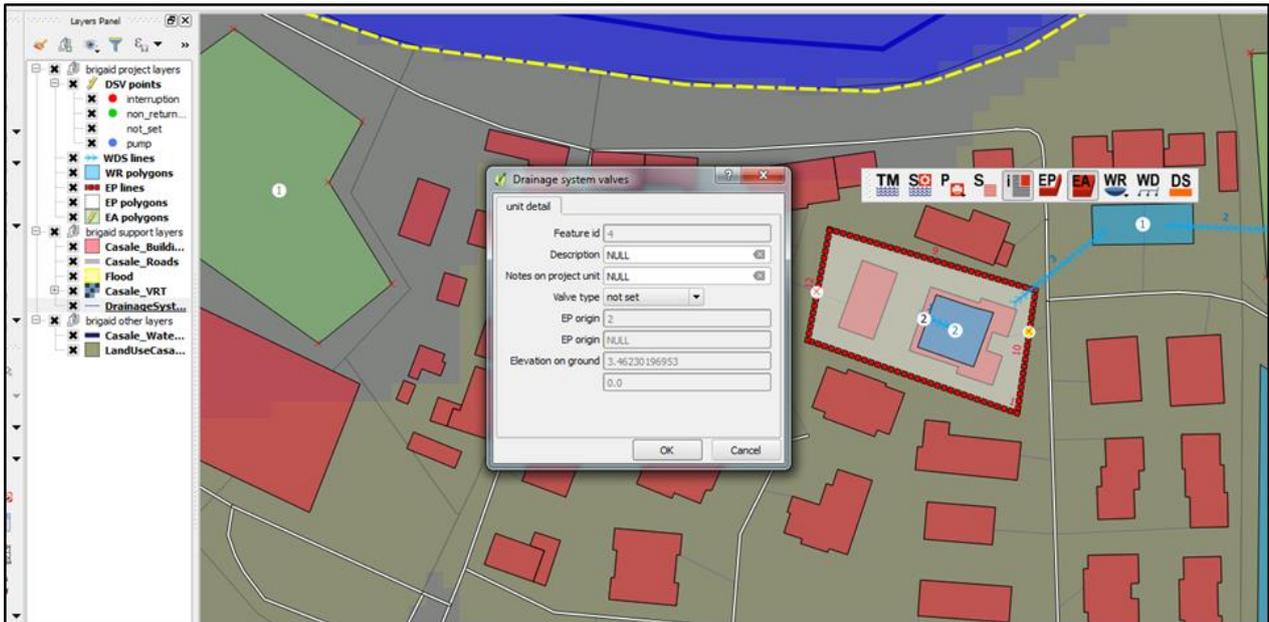


Figure 3-12 Window for defining the types of valves that can be implemented in the plug-in.

### 3.6 Report generation and spatial elements' query

In the plug-in bar shown in Figure 2-4 there are other buttons of the plug-in which perform the following functions:

-  By activating this button it is possible to query with the mouse all the features created with the plug-in. In particular, it is possible to see all the attributes and, if the user is in the writing mode of that particular shapefile, it is possible to modify it.
-  This button activates or disable the summary window of the tables generated by the plug-in.

-  With this command, the plug-in automatically generates a report containing all the maps and attribute tables of the elements created by the planner user.

# 4 Customizable libraries

The technological solutions that the user can apply are customizable and can be increased modifying specific libraries. The file that manages these libraries is an editable text file (with a simple notepad) which is located inside the installation library of the plug-in. Its path is:

```
"C: \ Users \ username \ .qgis2 \ python \ plugins \ fdtm"
```

The text file managing the libraries is called "fdtm\_definitions.py" and from line 161 to line 216 (see Figure 4-1) the different libraries developed for each element of the plug-in are listed.

Each line of the code represents a specific technology. The commented and green lines explain the meaning of the code sequence reported in each record.

The structural sequence that describes each element of the EP library is:

- Item value: design name of the element (in the plug-in it will be called as "Item\_value");
- Item label: label that is displayed in the drop-down menu
- Validation\_rules: Field not significant
- Mobile cost X unit: Field not significant
- Fixed cost X Unit: Unit cost per single element of measurement unit

Each element that describes a specific EP must be separated by a comma.

For example, for the EP elements library at the moment you can choose one of the following technologies described by the "item\_label" field: "existing" (existing wall), "sand" (sandbag), "airdam" (inflatable dam) and "concrete" (wall to be built).

Libraries for EA, WR, WDS elements are managed in the same way. What changes is the measurement unit to which the unit cost refers. For example, for the EA elements, the cost refers to the price for a cubic meter of raised surface, while for the linear element of the WDS elements it is referred to the linear meter of construction. The implemented valves, also managed with a special library, report the price for each one.

```

161 EP_TYPES_TABLE = [
162     #['item value', 'item label', 'validation_rule', 'mobile cost x unit', 'fixed cost x unit']
163     ['existing', 'existing', 'validation_rule:', '0.0', '0'],
164     ['sand', 'sand', 'validation_rule:', '2000.0', '15'],
165     ['airdam', 'airdam', 'validation_rule:', '5000.0', '10'],
166     ['concrete', 'concrete', 'validation_rule:', '10000.0', '150'],
167 ]
168
169 EP_TEMP_TYPES_TABLE = [
170     #['item value', 'item label', 'validation_rule', 'attribution_rule', 'cost x unit'],
171     ['fixed', 'fixed protection', 'validation_rule:', 'attribution_rule:', '0'],
172     ['mobile', 'mobile protection', 'validation_rule:', 'attribution_rule:', '0'],
173 ]
174
175 EA_TYPES_TABLE = [
176     #['item value', 'item label', 'validation_rule', 'attribution_rule', 'cost x unit'],
177     ['groundrise', 'groundrise', 'validation_rule:', 'attribution_rule:', '25'],
178     ['platform', 'platform', 'validation_rule:', 'attribution_rule:', '200'],
179     ['pilework', 'pilework', 'validation_rule:', 'attribution_rule:', '150'],
180 ]
181
182 WR_TYPES_TABLE_point = [
183     #['item value', 'item label', 'validation_rule', 'attribution_rule', 'cost x unit'],
184     ['tank', 'tank', 'validation_rule:', 'attribution_rule:', '1'],
185     ['tank1', 'tank1', 'validation_rule:', 'attribution_rule:', '10'],
186     #['rain_garden', 'giardino pluviale', 'validation_rule:', 'attribution_rule:', '150'],
187 ]
188
189 WR_TYPES_TABLE_greenroof = [
190     #['item value', 'item label', 'validation_rule', 'attribution_rule', 'cost x unit'],
191     ['green_roof_01', 'green roof 0.01 m\u00B3/m\u00B2', 'validation_rule:', '0.01', '100'],
192     ['green_roof_02', 'green roof 0.02 m\u00B3/m\u00B2', 'validation_rule:', '0.02', '150'],
193     ['green_roof_03', 'green roof 0.03 m\u00B3/m\u00B2', 'validation_rule:', '0.03', '200'],
194     ['green_roof_04', 'green roof 0.04 m\u00B3/m\u00B2', 'validation_rule:', '0.04', '250'],
195     ['green_roof_05', 'green roof 0.05 m\u00B3/m\u00B2', 'validation_rule:', '0.05', '300'],
196     #['rain_garden', 'giardino pluviale', 'validation_rule:', 'attribution_rule:', '150'],
197 ]
198
199 WR_TYPES_TABLE_limited = [
200     #['item value', 'item label', 'validation_rule', 'attribution_rule', 'cost x unit'],
201     ['basin', 'basin', 'validation_rule:', 'attribution_rule:', '2'],
202     ['floodable_park', 'floodable_park', 'validation_rule:', 'attribution_rule:', '4'],
203     ['rain_garden', 'rain_garden', 'validation_rule:', 'attribution_rule:', '3'],
204 ]
205
206 WR_TYPES_TABLE_unlimited = [
207     #['item value', 'item label', 'validation_rule', 'attribution_rule', 'cost x unit'],
208     ['unlimited_basin', 'unlimited_basin', 'validation_rule:', 'attribution_rule:', '0'],
209 ]
210
211 WDS_TYPES_TABLE = [
212     #['item value', 'item label', 'validation_rule', 'attribution_rule', 'cost x unit'],

```

Python file

length:11478 lines:255

Ln:202

Figure 4-1 Text file for the management of the libraries within the plug-in.