GISwater

PROTOCOL DOCUMENT

E-13.2

How to add and configure a new *mapzone*

GOALS

Know which steps must be taken both in QGIS and in the database to add a new ***mapzone*** and subsequently **configure its use and impact**. This always taking into account the use of **dynamic** *mapzones*, managed by the user but created from an internal **Giswater algorithm**.

DESCRIPTION

Below are the steps to add a new mapzone. For the different types that exist, the steps to follow are exactly the same.

Background

1. In system variable ***utils_dynamicmapzones_status***, *om_dynamicmapzones_status (v3.3)* define the graph classes that are enabled:

{"SECTOR":true, "PRESSZONE":true, "DQA":true, "MINSECTOR":true, "DMA":true}

2. In the ***cat_feature_node table***, *node_type (v3.3)*, for each type of node we define graf_delimiter. The sample has:

id [PK] character varying(30)	type character varying(30)	ep ch	m ch	epa cha	ac bc	co bc	nu in	ch bc	de te	lir ch	isarcdivide boolean	graf_delimiter character varying(20)
ADAPTATION	JUNCTION	JU	ma	inj	FA	TR	2	TR	Ac		TRUE	NONE
AIR_VALVE	VALVE	JU	ma	inj	TR	TR	0	TR	Ai		FALSE	NONE
BYPASS_REGISTER	REGISTER	JU	ma	inj	FA	TR	2	TR	By		TRUE	NONE
CHECK_VALVE	VALVE	SH	ma	inj	TR	TR	2	TR	Ch		TRUE	MINSECTOR
CLORINATHOR	NETELEMENT	SH	ma	inj	FA	TR	2	TR	EL		TRUE	DQA
CONTROL_REGISTER	REGISTER	VA	ma	inj	TR	TR	2	TR	Co		TRUE	NONE
CURVE	JUNCTION	JU	ma	inj	TR	TR	2	TR	Cu		TRUE	NONE
ENDLINE	JUNCTION	JU	ma	inj	TR	TR	1	TR	En		TRUE	NONE
EXPANTANK	EXPANSIONTANK	JU	ma	inj	TR	TR	2	TR	Ex		TRUE	NONE
FILTER	FILTER	SH	ma	inj	TR	TR	2	TR	Fi		TRUE	NONE
FL_CONTR_VALVE	VALVE	VA	ma	inj	FA	TR	2	TR	FL		TRUE	MINSECTOR
FLEXUNION	FLEXUNION	JU	ma	inj	TR	TR	2	TR	FL		TRUE	NONE
FLOWMETER	METER	JU	ma	inj	TR	TR	2	TR	FL		TRUE	DMA
GEN_PURP_VALVE	VALVE	VA	ma	inj	FA	TR	2	TR	Ge		TRUE	MINSECTOR
GREEN_VALVE	VALVE	JU	ma	inj	TR	TR	2	TR	Gr		TRUE	NONE
HYDRANT	HYDRANT	JU	ma	inj	TR	TR	2	TR	Hy		TRUE	NONE
JUNCTION	JUNCTION	JU	ma	inj	TR	TR	2	TR	Ju		TRUE	NONE
MANHOLE	MANHOLE	JU	ma	inj	TR	TR	2	TR	In		TRUE	NONE
NETELEMENT	NETELEMENT	JU	ma	inj	TR	TR	2	TR	Ne		TRUE	NONE
NETSAMPLEPOINT	NETSAMPLEPOINT	JU	ma	inj	TR	TR	2	TR	Ne		TRUE	NONE
OUTFALL_VALVE	VALVE	JU	ma	inj	TR	TR	2	TR	Ou		TRUE	NONE
PR_BREAK_VALVE	VALVE	VA	ma	inj	TR	TR	2	TR	Pr		TRUE	PRESSZONE
PR_REDUC_VALVE	VALVE	VA	ma	inj	TR	TR	2	TR	Pr		TRUE	PRESSZONE
PR_SUSTA_VALVE	VALVE	VA	ma	inj	TR	TR	2	TR	Pr		TRUE	PRESSZONE
PRESSURE_METER	METER	JU	ma	inj	TR	TR	2	TR	Pr		TRUE	NONE
PUMP	PUMP	PU	ma	inj	TR	TR	2	TR	Pu		TRUE	NONE
REDUCTION	REDUCTION	JU	ma	inj	TR	TR	2	TR	Re		TRUE	NONE
REGISTER	REGISTER	JU	ma	inj	TR	TR	2	TR	Re		TRUE	NONE
SHUTOFF_VALVE	VALVE	SH	ma	inj	TR	TR	2	TR	Sh		TRUE	MINSECTOR
SOURCE	SOURCE	JU	ma	inj	TR	TR	2	TR	So		TRUE	SECTOR
T	JUNCTION	JU	ma	inj	TR	TR	3	TR	Ju		TRUE	NONE
TANK	TANK	TA	ma	inj	TR	TR	2	TR	Ta		TRUE	SECTOR
TAP	JUNCTION	JU	ma	inj	FA	TR	2	TR	Ta		TRUE	NONE
THROTTLE_VALVE	VALVE	VA	ma	inj	FA	TR	2	TR	Th		TRUE	MINSECTOR
VALVE_REGISTER	REGISTER	JU	ma	inj	FA	TR	2	TR	Va		TRUE	NONE

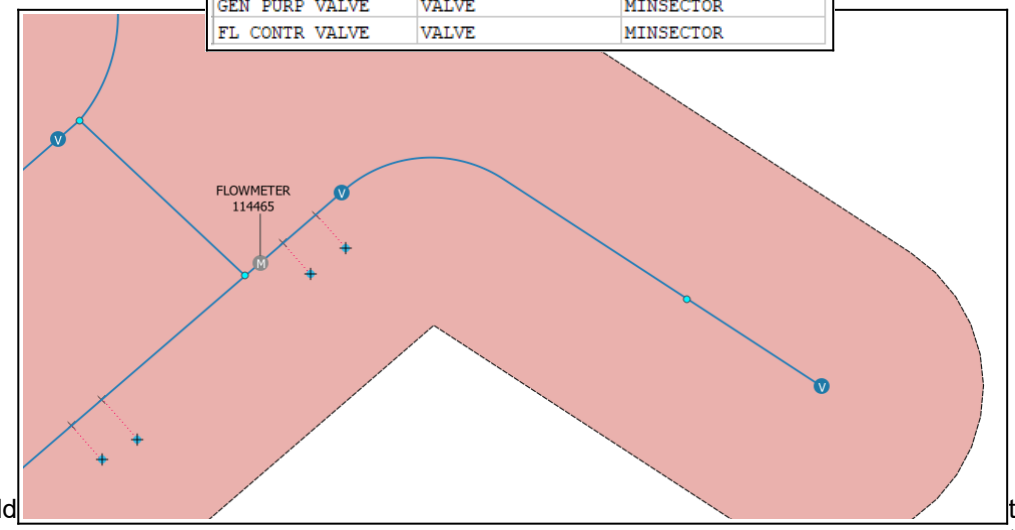
3. In the valve

configuration table ***config_valve***, *anl_mincut_selector_valve (v3.3)* define the valves that act as stoppers

Example to add a new DMA type mapzone:

1) Add in QGIS a **new node** in the desired location. The type of node must be one that in the *node_type* table has as a value in the column *graf_delimiter* '**DMA**', since this column establishes which areas of the map can delimit the different types of node. If we want to make an area of the map **different** than dma, the only necessary thing to do is to insert a node that is the delimiter of the desired area. For DMA it will usually be 'METER' type nodes.

id character varying(30)	type character varying(30)	graf_delimiter character varying(20)
FLOWMETER	METER	DMA
CLORINATHOR	NETELEMENT	DQA
CHECK VALVE	VALVE	MINSECTOR
GEN PURP VALVE	VALVE	MINSECTOR
FL CONTR VALVE	VALVE	MINSECTOR



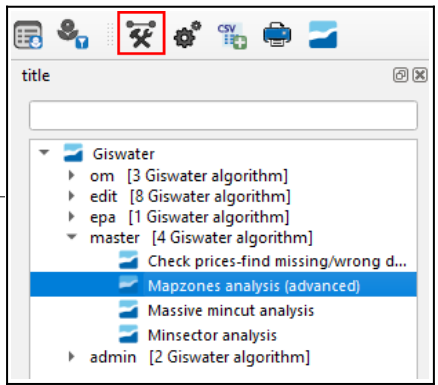
2) Add to the dma table (if we make another *mapzone* fill the corresponding table) a new value, with the name of the dma that we want to create and, above all, fill the **grafconfig** value with the following ***json***. In the value of **nodeParent** we must put the id of the node that we just inserted, which will be the header of the new dma and in **toArc** we indicate the direction to take by means of the id of the section. Example:

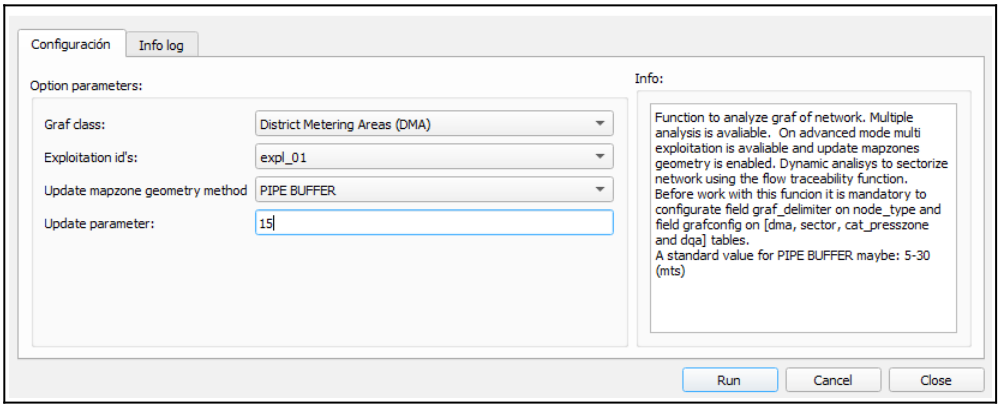
```
{"use":[{"nodeParent":"1080", "toArc":[2092]}], "ignore":[], "stopper":[1057,41,1060]}
```

In **ignore** (optional) we will put, if it is the case, those nodes that being grafdelimiter, we do not want them to participate in the algorithm (for being out of service or for not doing its job).

In **stopper** (optional) we will put, if it is the case, those nodes that we want to force stop the flood algorithm.

3) We use the **Giswater toolbox** to **recalculate** the *mapzones* with the modification that we have made for dma's, by selecting the shape of the geometry that we want.





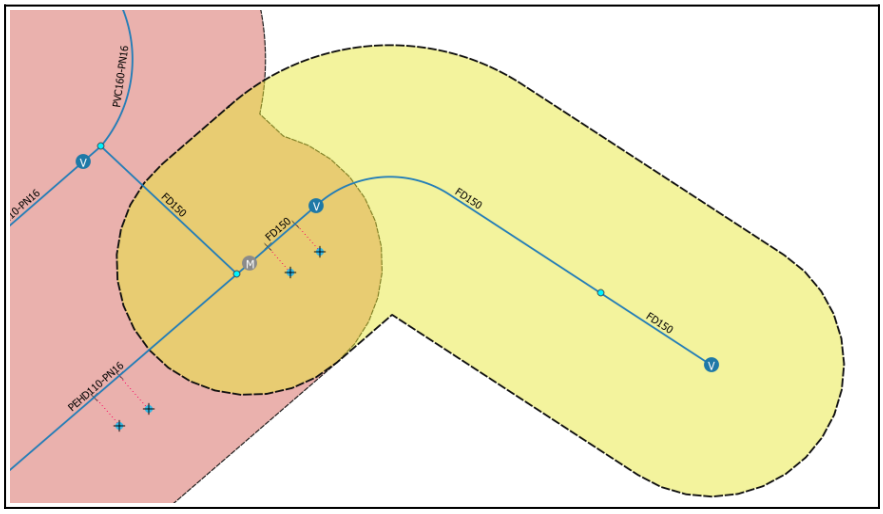
An alternative to using the Giswater toolbox, is to shoot the function directly from the database, making a call such that:

```
SELECT gw_fct_grafanalytics_mapzones('{"data":{"parameters":{"grafClass":"DMA",
"exploitation":[1], "macroExploitation":[1], "checkData":false,
"updateFeature":true, "updateMapZone":2, "geomParamUpdate":15,"debug":false,
"usePlanPsector":false, "forceOpen":[1,2,3], "forceClosed":[2,3,4]}}}');
```

where:

KEY	OBLIG.	DESCRIPTION	EXAMPLE	VALUE RANGE
grafClass	YES	Graph class to make	DMA	DMA, SECTOR, DQA, PRESSZONE
exploitation	YES	Exploitations to participate in the algorithm	[1,2]	All the available exploitations
macroExploitation	NO	Macroexplotations to participate in the algorithm	[1,2]	All the available macroexplotations
checkData	YES	If true, check if the system values are correct (topology, state_type, etc), In case there are errors, abort the process	false	false, true
updateFeature	YES	If true, update the values of dma_id, presszone_id, sector_id & dma_id of all NODES, ARCS, CONNEC that are flooded by the algorithm	true	false, true
updateMapZone	YES	0: does not update the geometry field (the_geom) of the mapzone 1: Updates by making an enveloping polygon with all the elements 2: Updates by buffering the sections with the value of geomParamUpdate 3: Updates by buffering the sections with the value of geomParamUpdate and incorporating the plot geometry (if it exists)	2	0,1,2,3
geomParamUpdate	YES	Value related to the options 2, 3 of the previous key	10	Any float between 0.1 – 100
usePlanPsector	YES	If true, use every psectors of the exploitation in the alogrithm analysis	false	false, true
forceOpen	YES	Valves that can be forced to open (e.g. for closed valves that for whatever reason we want to open)	[1,2,3]	Every closed valves
forceClosed	YES	Nodes in general which can be forced to close (e.g. in the debugging phase in case the trace gets out of control and does not converge as expected)	[1,2,3]	Every nodes (except closed valves)

4) Once the **process is finished**, we must check again the geometries of the **dma**. In our example, we see that the geometry has been generated for the new dma that has the FLOWMETER that we have added as a delimiter node.



REVIEWS

Action	User	Date
Created	Albert Bofill	09/04/2020
Updated	Xavier Torret	01/08/2020
Updated	Xavier Torret	01/12/2021