QGIS-plugin Geodyn Gemeente

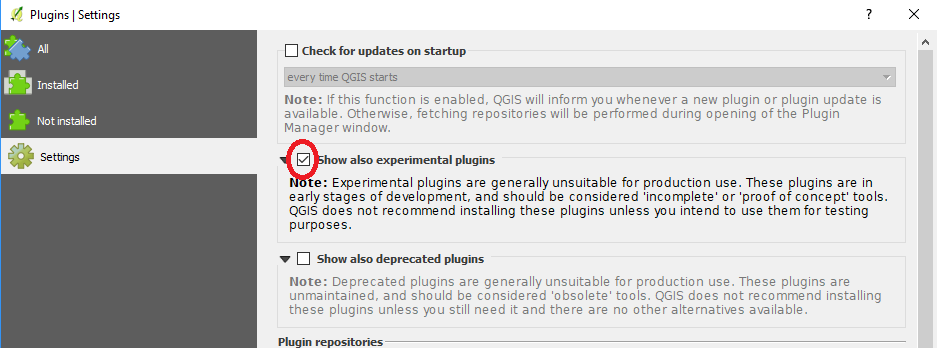
# Gebruikershandleiding

Auteur: B. Kropf

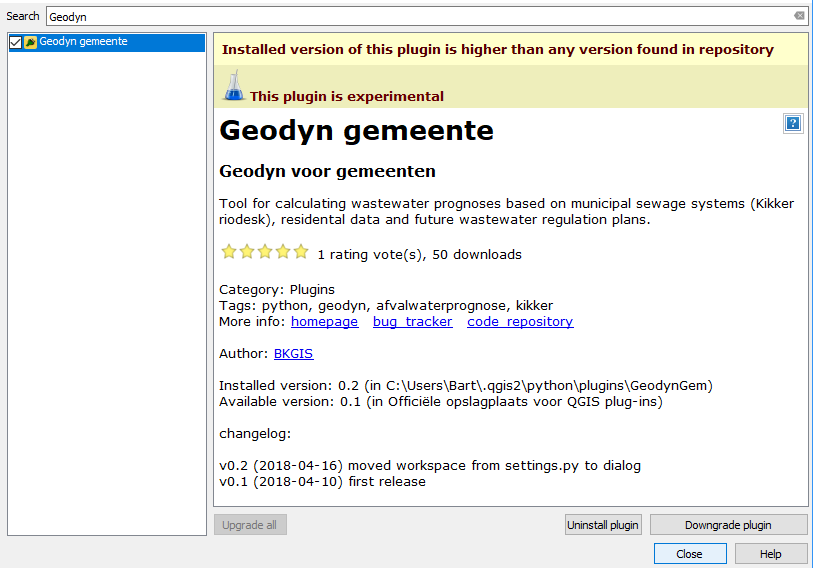
Datum: 17-04-2018

Stap 1.) Start QGIS (v2.x) en open Plugins via het hoofdmenu

Stap 2.) Ga naar settings en kruis aan “Show also experimental plugins”



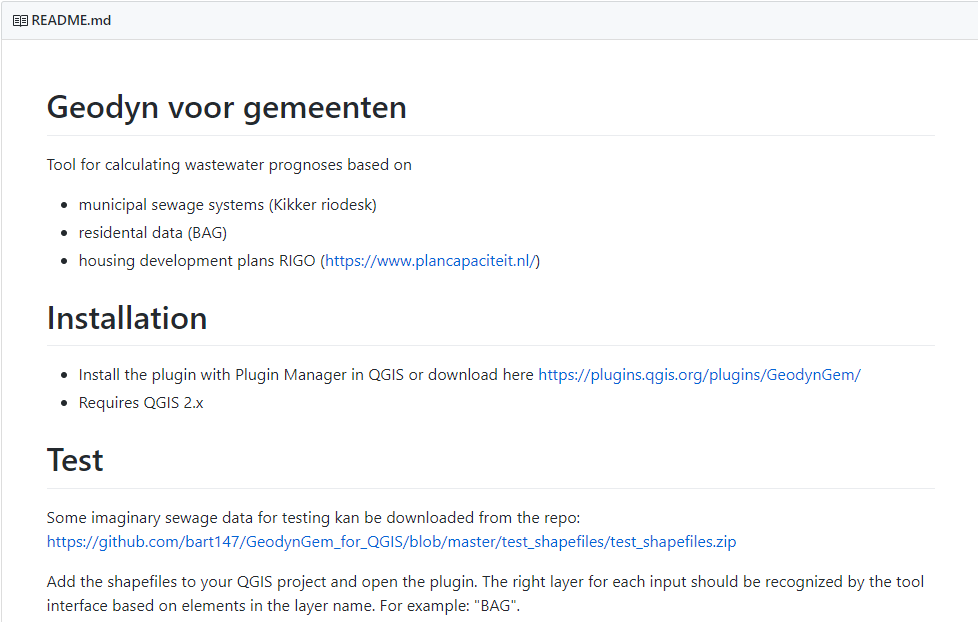
Stap 3.) Ga naar de zoekbalk en typ: “Geodyn”.



Stap 4.) Installeer de plugin “Geodyn gemeente” en klik op de link homepage of code repository.

Op de pagina die opent op GitHub is de broncode van de plugin te zien. Onderaan bij README.md staan o.a. instructies over de installatie en een link met testdata.

Klik op de link onder kopje Test om shapefiles te downloaden om mee te testen.

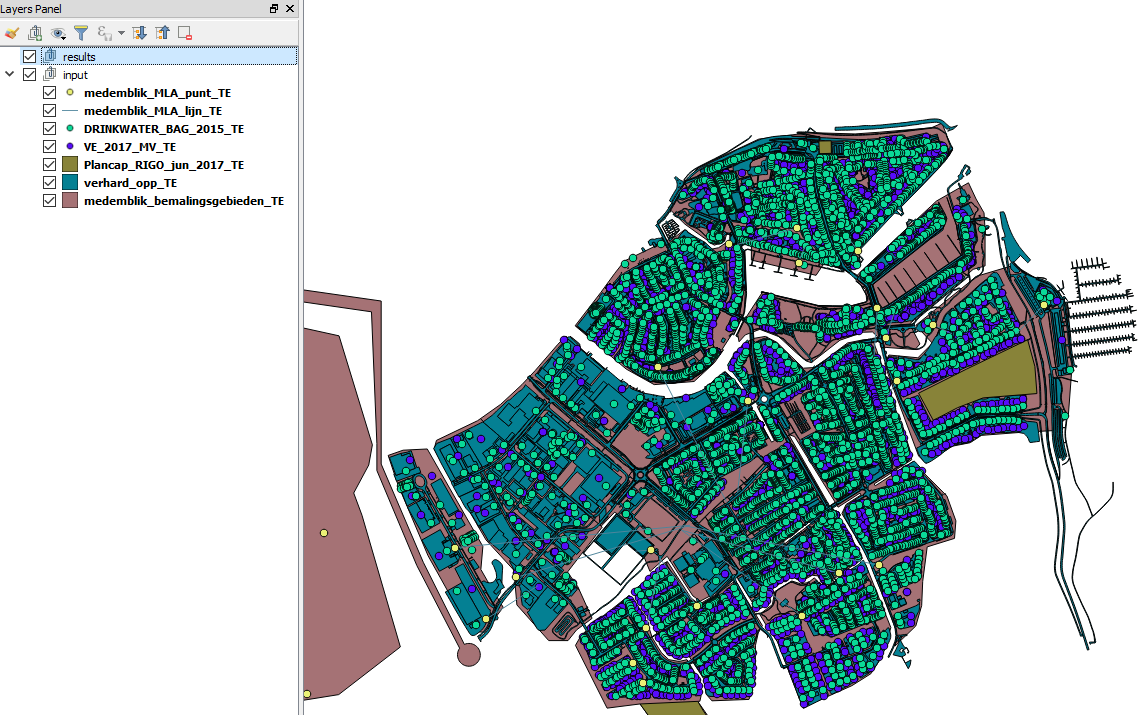


Stap 5.) Pak de shapefiles uit en voeg de data toe aan QGIS.

**Tip**: groepeer de shapefiles en noem de groep bijvoorbeeld ‘input’ (rechtermuismenu)

Maak alvast een nieuwe groep genaamd ‘results’ en selecteer deze door erop te klikken.

De resultaten van de plugin komen nu automatisch hierin terecht en zo blijft de input netjes gescheiden van de output.



Stap 6.) Open de plugin door op het icoontje te klikken (op moment van schrijven een stekker).

De juiste lagen worden als het goed is automatisch herkend in de dropdown lists.

Dat gebeurd op basis van bepaalde stukken tekst in de laagnaam.

Achtereenvolgend zijn dat:

“MLA” (voor punten en lijnen export uit Kikker)

“BAG” (voor drinkwatergegevens)

“VE” (voor de vervuilingseenheden)

“RIGO” (voor de plancapaciteiten)

“opp” (voor verhard oppervlak)

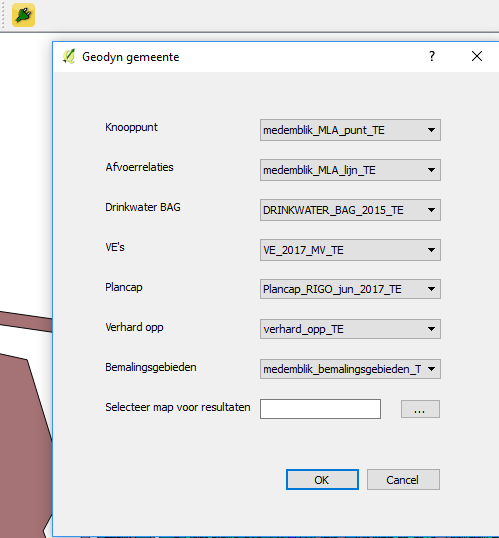
“bemaling” (voor de bemalingsgebieden)

Indien de laagnaam afwijkt en er geen match gevonden wordt, komt gewoon een willekeurige laag bovenaan en moet de juiste laag met de hand gekozen worden.

Controleer altijd of de juiste lagen geselecteerd zijn.

Onderin moet een output folder geselecteerd worden waarin alle resultaten terecht komen (als shapefile).

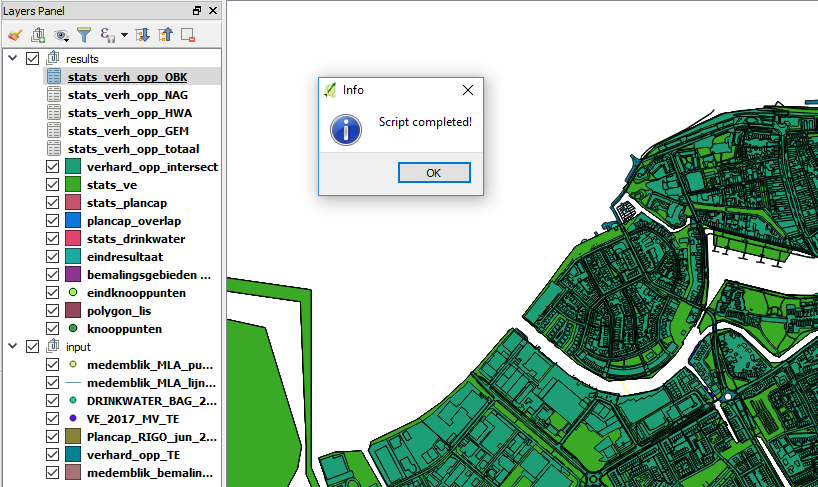
Klik op OK.



Stap 7.) De resultaten worden nu aan de Layers Panel toegevoegd en een popup verschijnt als het script klaar is.

Het eindresultaat heet: “**eindresultaat**”.

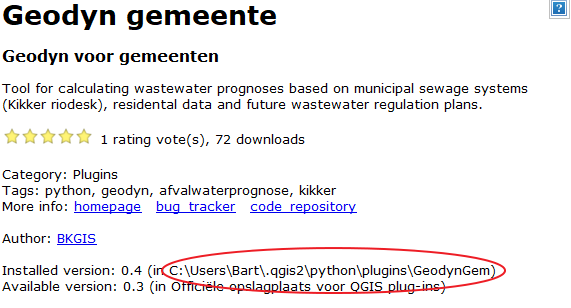
Alle andere gegevens zijn tussenresultaten en kunnen in principe weer uit de Layers panel verwijderd worden.



# Beheerdershandleiding

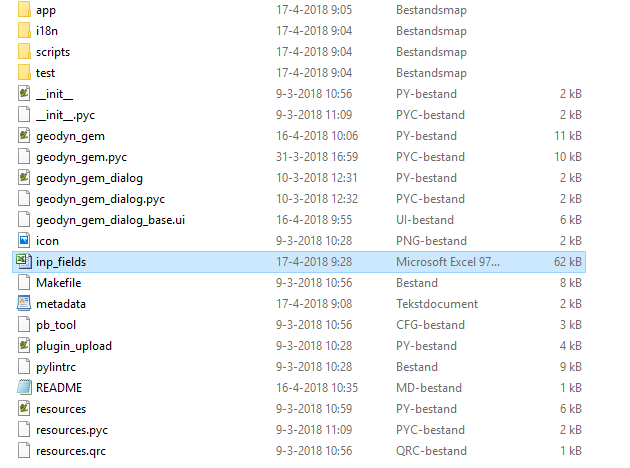
Installatiemap plugin

In het informatiescherm van de plugin in QGIS is het pad te achterhalen waar de plugin geïnstalleerd is.



In deze map staan alle bestanden van de plugin:

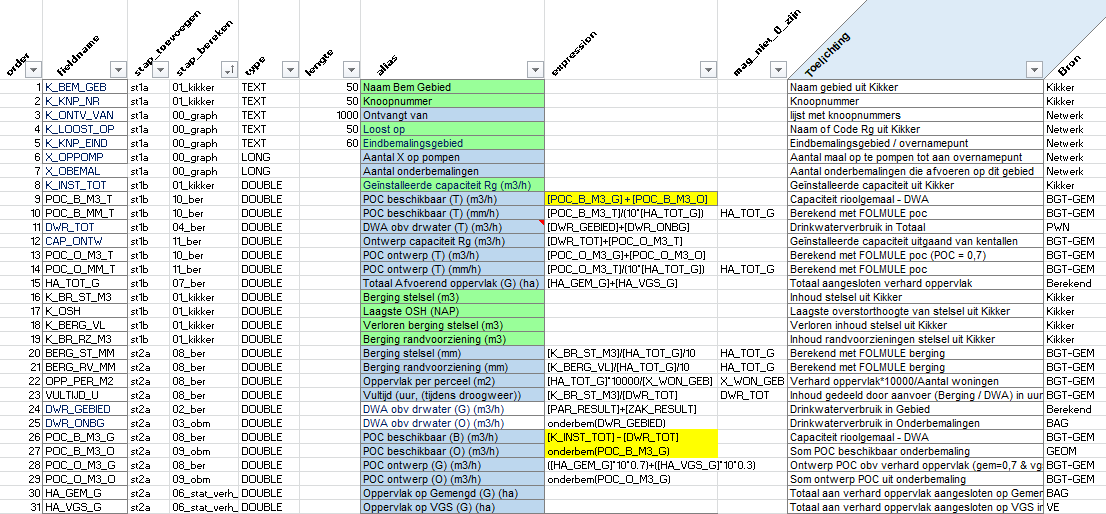
py-scripts, icon.png, readme.md, metadata.txt, wat installatiebestanden en een excel-bestandje **inp\_fields.xls**.



Toelichting input velden

**inp\_fields.xls** is het inputbestand met daarin alle velden die berekend worden in de plugin.

Berekeningen, veldvolgorde kunnen in dit overzicht worden aangepast (tot op zekere hoogte).

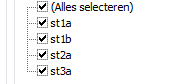


Hieronder volgt een korte toelichting van de betekenis van de velden:

**Order**: de volgorde van de velden waarin de velden in het eindresultaat terecht komen.

**Fieldname**: de veldnaam. Het is niet veilig om de veldnamen te wijzigen omdat een aantal ook hardcoded in het script worden gebruikt!

**Stap\_toevoegen**: de stap waarin de velden worden toegevoegd in het script.



De velden worden dus in 4 stappen toegevoegd aan het eindresultaat door het script. Binnen de stappen wordt de volgorde bepaald door veld ‘order’. De labels (bijv: ‘ st1a’) kunnen niet zomaar gewijzigd worden omdat ze worden gebruikt in de scripts. Binnen een stap is het redelijk veilig om de veldvolgorde te wijzigen. Het verschuiven van velden naar andere stappen is niet aan te bevelen en geeft risico op fouten.

**Stap\_bereken:** dit zijn de stappen waarin de analyse wordt uitgevoerd.



Veel resultaten zijn onderling afhankelijk en daarom is de volgorde van deze stappen van belang.

Alle labels met ‘ber’ erin zijn berekeningen en maken gebruik van veld ‘expression’.

**Type**: dit is het veld type: LONG, DOUBLE of TEXT of DATE.

**Lengte**: lengte veld in indien type TEXT

**Alias**: De veld-aliassen (n.v.t. in shapefiles)

**Expression**: De sommetjes die uitgevoerd worden. (voor alle stap\_bereken labels met ‘ber’ erin)

Berekeningen kunnen worden gewijzigd naar eigen inzicht.

**Mag\_niet\_0\_zijn:** Als hier een veldnaam is ingevuld dan heeft de berekening (expression) vaak een deling door een veldwaarde. Om te voorkomen dat er door 0 gedeeld kan worden geeft dit veld aan dat de veldwaarde niet 0 mag zijn.

**Toelichting**: is puur een toelichting ter verduidelijking van het overzicht, wordt niet in script gebruikt en kan naar eigen inzicht aangepast worden.

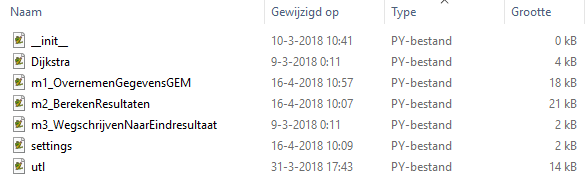
**Bron**: Ter verduidelijking van overzicht. Geeft aan wat de bron is van een veld. Vaak zijn het echter combinaties van bronnen.

Toelichting python-scripts

In de rootfolder GeodynGem staan een aantal py-scripts. De meeste daarvan zijn standaard voor iedere QGIS plugin die met plugin-builder zijn gemaakt.

**geodyn\_gem.py**: hierin wordt de communicatie met de gui dialog geregeld. Bijvoorbeeld de kaartlagen die als input van de analyse dienen en het automatisch herkennen van de juiste kaartlaag op basis van de naam.

In de map **app** staande alle scripts die voor de analyse gebruikt worden.



**Utl.py** voor de utilities (help-functies).

**Settings.py** voor instellingen.

**Dijkstra.py** voor het gebruik van Graph-objects om onderbemalingen te berekenen.

De analyse zelf vindt plaats in twee stappen: m1 en m2

**m1\_OvernemenGegevensGEM.py** voor het bepalen van het netwerk, de afvoerrelaties, waardes overnemen uit kikker. Koppelen id’s aan bemalingsgebieden.

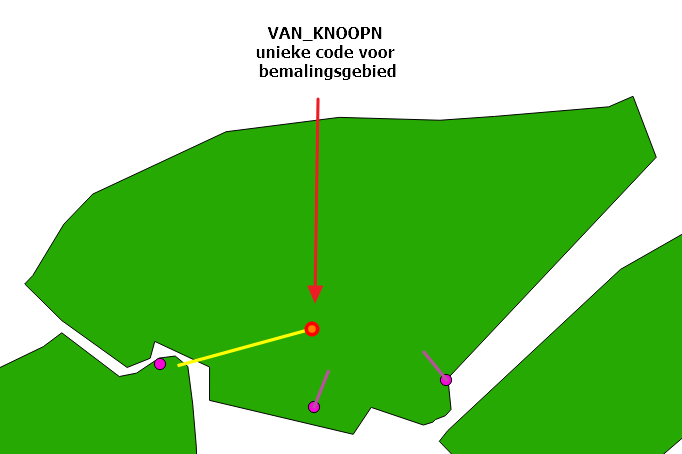
**m2\_BerekenResultaten.py** hierin worden de meeste berekeningen gedaan, onderbemalingen berekend en ruimtelijke koppelingen gedaan met drinkwatergegevens, VE’s en plancapaciteiten.

Verdieping analyse en aandachtspunten bij gebruik.

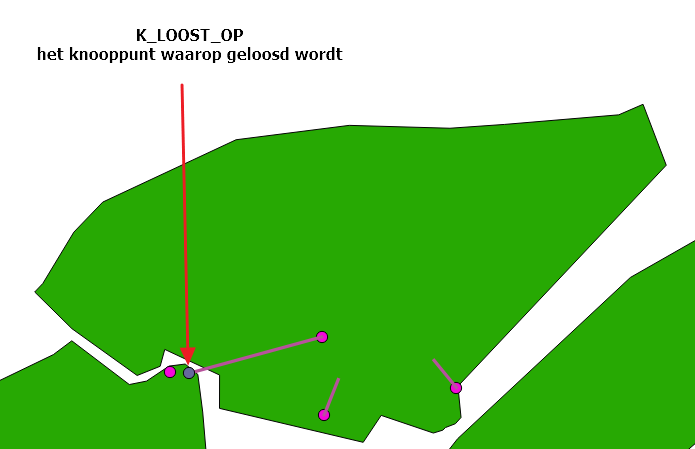
**Bepalen van knooppunten en afvoerrelaties**

Voor het bepalen van de code’s voor bemalingsgebieden en afvoerrelaties worden de export bestanden van Kikker gebruikt. Die bestaan uit knooppunten en afvoerlijnen.

Als er meerdere knooppunten in een bemalingsgebied vallen (bijv. bij drukriolering) is er altijd 1 knooppunt die leidend is voor het bemalingsgebied en waarvan de code “VAN\_KNOOPN” wordt overgenomen. Om deze te bepalen wordt gezocht naar het knooppunt dat afvoert op een ander bemalingsgebied. Ander knooppunten worden genegeerd.



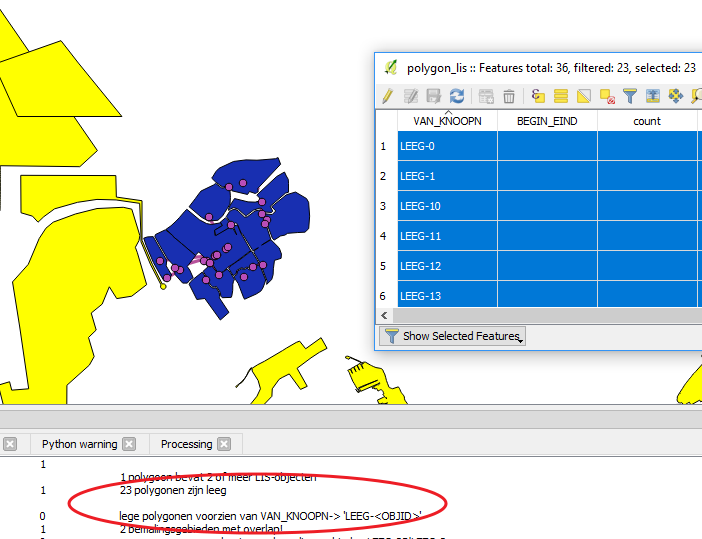
Vervolgens wordt het eindknooppunt bepaald waarop afgevoerd wordt. De code wordt overgenomen van de VAN\_KNOOPN van het bemalingsgebied waarin deze valt en opgeslagen als attribuut K\_LOOST\_OP.



**Bemalingsgebieden zonder knooppunt (met geldige afvoerrelatie)**

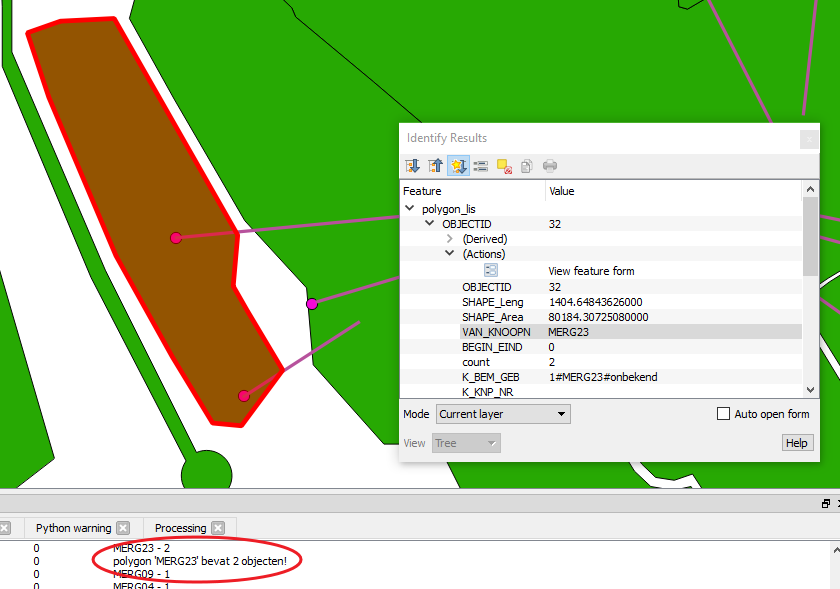
Voor bemalingsgebieden zonder knooppunt met geldige afvoerrelatie (knooppunten die afvoeren op andere bemalingsgebieden) wordt een unieke code gegenereerd. Dat is niet alleen handig maar ook nodig om de vervolgstappen in het script goed te kunnen uitvoeren. In het logboek wordt melding gemaakt van het aantal “lege” bemalingsgebieden.

Voor een goed eindresultaat is het natuurlijk de bedoeling dat alle bemalingsgebieden een eigen knooppunt en afvoerrelatie hebben.



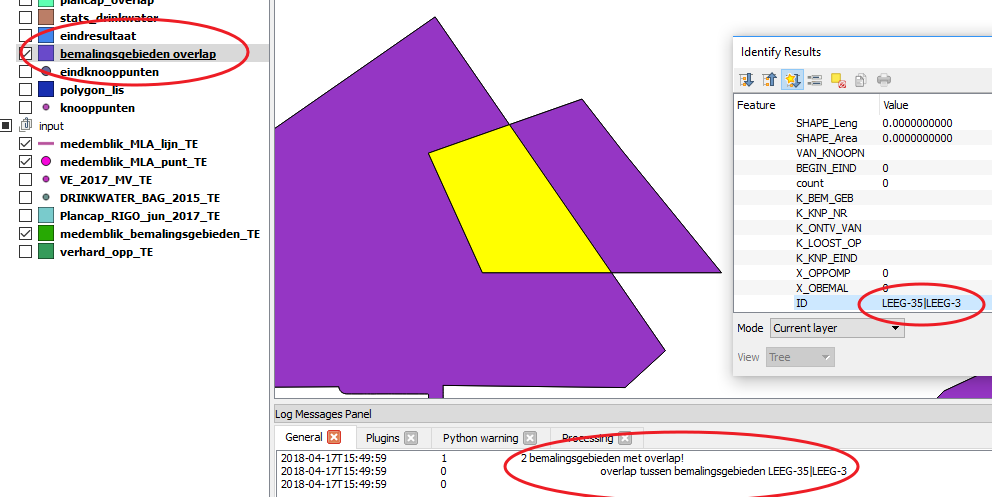
**2 knooppunten in hetzelfde bemalingsgebied**

Als er toch 2 knooppunten liggen in één bemalingsgebied die beide afvoeren op een ander bemalingsgebied, wordt daarvoor een fout gegenereerd in het log.



**2 bemalingsgebieden die overlappen**

Als er 2 bemalingsgebieden voorkomen die elkaar overlappen wordt hiervoor een extra output gegenereerd: “bemalingsgebieden\_overlap”. Er wordt ook melding gemaakt in het logboek.



**Plancapaciteit in meerdere bemalingsgebieden**

Als een plancap RIGO gebied in meerdere bemalingsgebieden valt wordt hiervan een melding gemaakt in het logboek.

