# Afbeeldingsresultaat voor Logo HHNK

# 

*Gebruikershandleiding*

+

-pr



De GeoDyn plug-in voor Qgis is ontwikkeld door Bart Kropf (BKGIS) in opdracht van

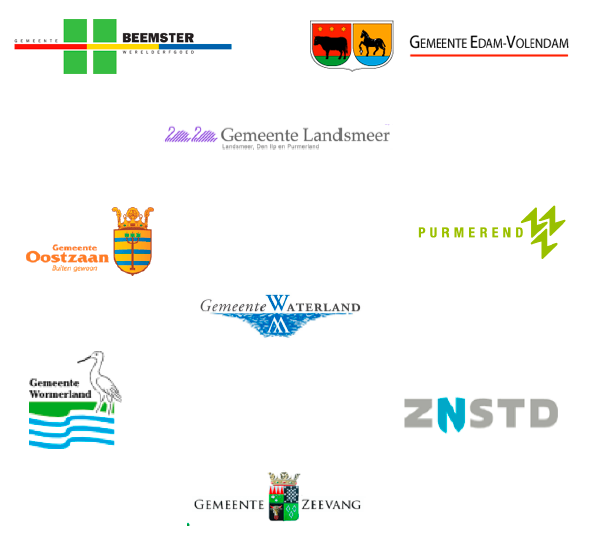
Koos Brouwer (gemeente Medemblik en de 8 gemeente van Zaanstreek Waterland)

De werking van GeoDyn is ontwikkeld door

Mark Lamers (Hoogheemraadschap Hollands Noorderkwartier)

Auteur: Bart Kropf

Datum: 16-01-2019



Inhoudsopgave

[1](#_Toc535411720)

[1](#_Toc535411721)

[Inleiding 4](#_Toc535411722)

[Installatie van GeoDyn plug-in in 7 stappen. 5](#_Toc535411723)

[Stap 1.) Start QGIS (v2.x) en open Plug-ins via het hoofdmenu 5](#_Toc535411724)

[Stap 2.) Ga naar settings en kruis aan “Show also experimental plug-ins” 5](#_Toc535411725)

[Stap 3.) Ga naar de zoekbalk en typ: “GeoDyn”. 5](#_Toc535411726)

[Stap 4.) Installeer de plug-in “GeoDyn gemeente” en klik op de link homepage of code repository. 5](#_Toc535411727)

[Stap 5.) Pak de shapefiles uit en voeg de data toe aan QGIS. 6](#_Toc535411728)

[Stap 6.) Open de plug-in door op het icoontje te klikken (op moment van schrijven een stekker). 7](#_Toc535411729)

[Stap 7.) De resultaten worden nu aan de Layers Panel toegevoegd en een popup verschijnt als het script klaar is. 8](#_Toc535411730)

[Verdieping analyse en aandachtspunten bij gebruik. 9](#_Toc535411731)

[Bepalen van bemalingsgebieden op basis van afvoerrelaties 9](#_Toc535411732)

[Bepalen eindgebieden 10](#_Toc535411733)

[Bemalingsgebieden zonder geldige afvoerrelatie 11](#_Toc535411734)

[2 knooppunten in hetzelfde bemalingsgebied 12](#_Toc535411735)

[2 bemalingsgebieden die overlappen 12](#_Toc535411736)

[Plancapaciteit in meerdere bemalingsgebieden 13](#_Toc535411737)

[Toelichting tussenresultaten 13](#_Toc535411738)

[Beheerdershandleiding 14](#_Toc535411739)

[Installatiemap plug-in 14](#_Toc535411740)

[Toelichting input velden 15](#_Toc535411741)

[Toelichting python-scripts 17](#_Toc535411742)

[Extra instellingen in local\_settings.py 18](#_Toc535411743)

# Inleiding

GeoDyn staat voor Geografisch Dynamisch Prognose systeem voor de afvalwaterketen. GeoDyn-gemeente is een applicatie die werkt als "plug-in" in QGIS en is beschikbaar in de QGIS plug-in-store.

Op basis van data van het drinkwaterbedrijf PWN, de gemeenten, HHNK en de provincie berekent deze tool afvalwaterhoeveelheden per gemeentelijk bemalingsgebied, voor het heden en de toekomst. De applicatie combineert deze resultaten, het verhardoppervlakte inventarisatie met stelsel- en gebiedsafgeleiden. Het resultaat is een zeer complete lijst met rioleringskenmerken per bemalingsgebied.

Deze handleiding is een stapsgewijze beschrijving van de werking van de GeoDyn plug-in in Qgis. Het beschrijft hoe je de plug-in installeert en hoe je de werking van de plug-in kan testen door gebruik te maken van ‘imaginary data’.

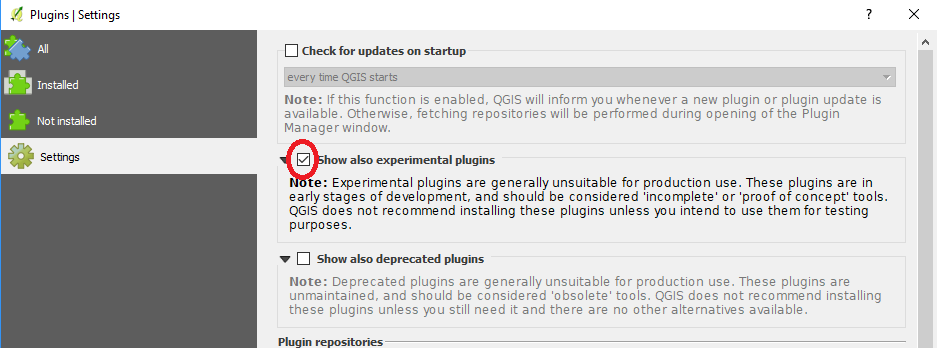
De meest recente handleiding is te vinden bij de overige bestanden van de plug-in op GitHub.

<https://github.com/bart147/GeodynGem_for_QGIS/tree/master/doc>

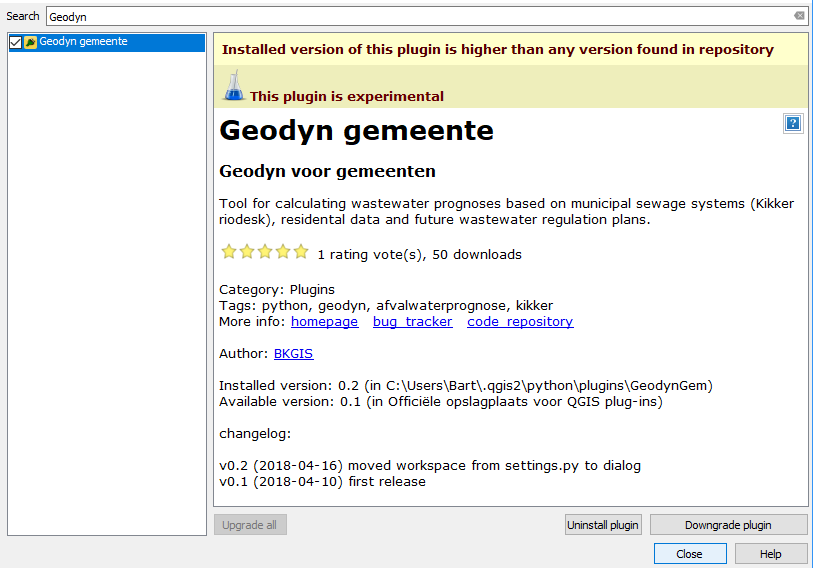
# Installatie van GeoDyn plug-in in 7 stappen.

## Stap 1.) Start QGIS (v2.x) en open Plug-ins via het hoofdmenu

## Stap 2.) Ga naar settings en kruis aan “Show also experimental plug-ins”



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



## Stap 4.) Installeer de plug-in “GeoDyn gemeente” en klik op de link homepage of code repository.

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

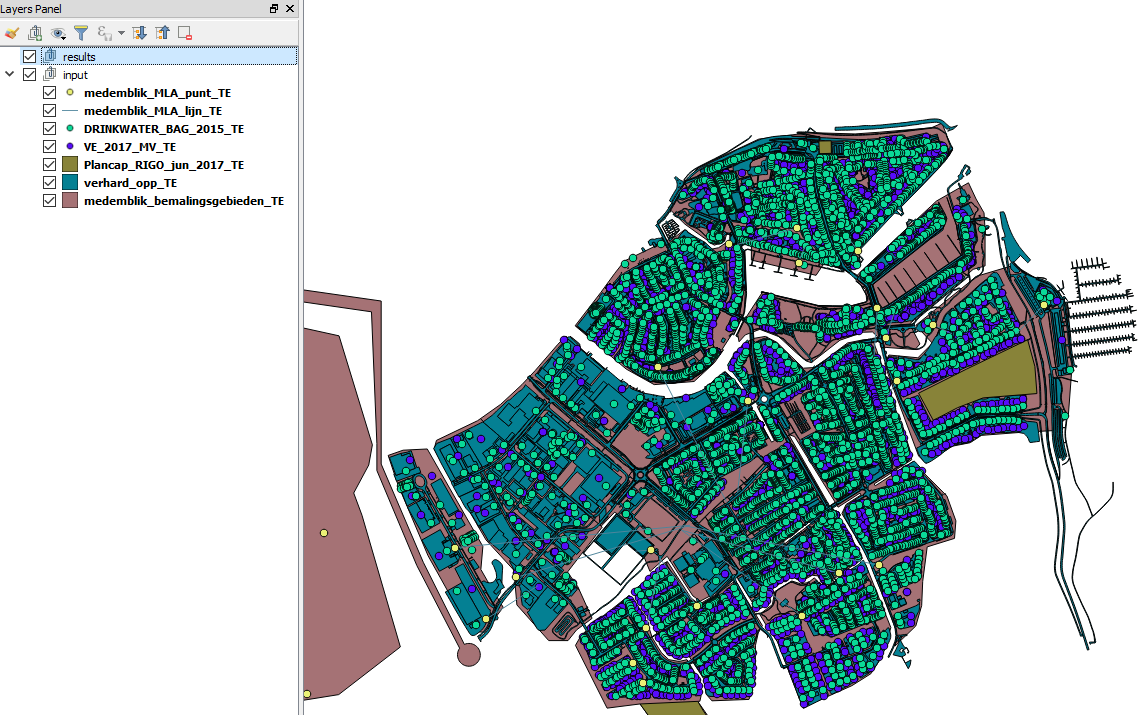
Klik op de link onder kopje Test om shapefiles te downloaden om mee te testen.

## 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 plug-in komen nu automatisch hierin terecht en zo blijft de input netjes gescheiden van de output.



## Stap 6.) Open de plug-in door op het icoontje te klikken (op moment van schrijven een stekker).

Let op!

Bij het openen van de plug-in worden reeds aanwezige layers met resultaten uit de Layer Panel verwijderd om te voorkomen dat bugs optreden met het overschrijven van data.

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:

“kikker” (voor puntenbestand uit Kikker)

“kikker” (voor lijnenbestand uit Kikker)

“BAG” (voor drinkwatergegevens)

“VE” (voor de vervuilingseenheden)

“RIGO” (voor de plancapaciteiten)

“opp” (voor verhard oppervlak)

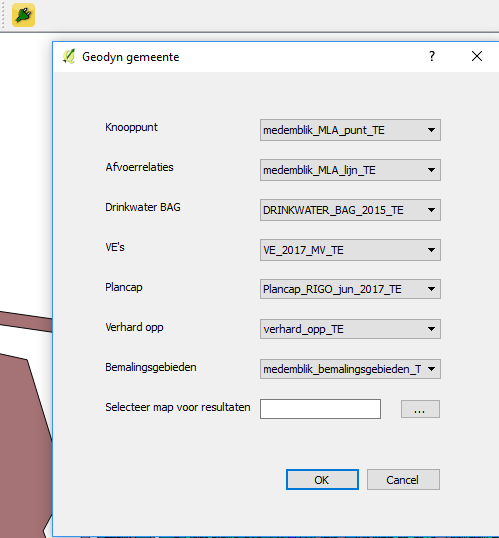
“bem” (voor de bemalingsgebieden)

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.

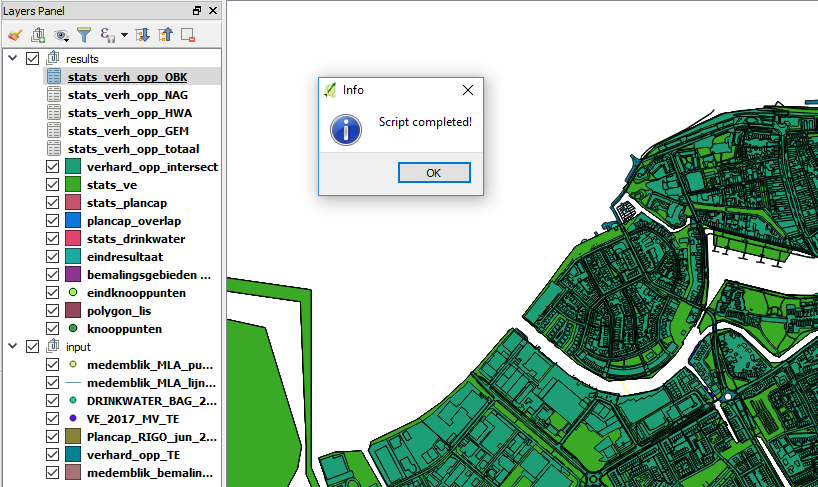
Klik op OK.



## 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.

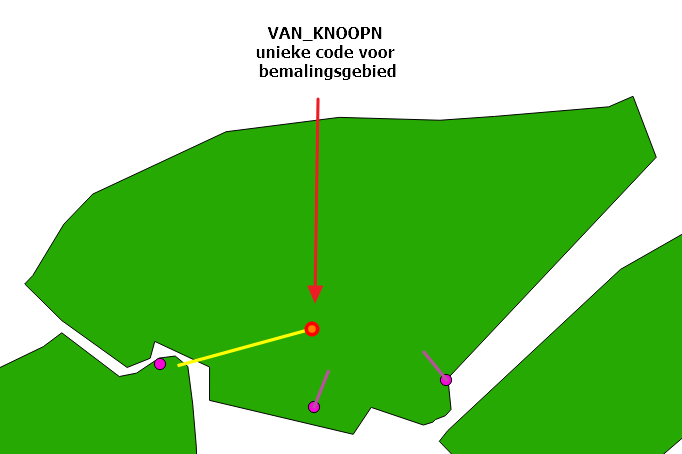


# Verdieping analyse en aandachtspunten bij gebruik.

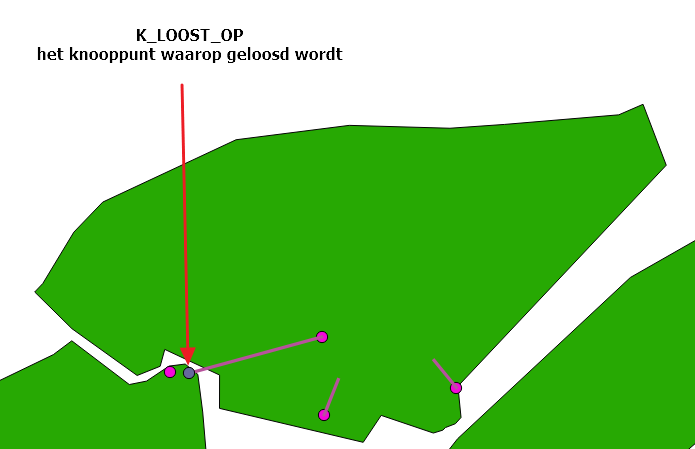
### Bepalen van bemalingsgebieden op basis van afvoerrelaties

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.



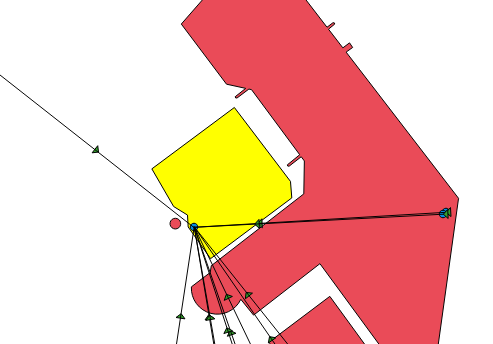
Vervolgens wordt het eindknooppunt bepaald waarop afgevoerd wordt. De code wordt overgenomen van de VAN\_KNOOPN van het bemalingsgebied waarin deze valt en opgeslagen als attribuut K\_LOOST\_OP.



### Bepalen eindgebieden

Bemalingsgebieden worden dus bepaald op basis van beginpunten van afvoerrelaties. In sommige gevallen, bijvoorbeeld bij een rwzi, is er geen afvoer meer uit het gebied. Om deze gebieden ook mee te nemen worden eindgebieden bepaald.

De benaming van dit bemalingsgebied (VAN\_KNOOPN) wordt gehaald uit het knooppunten bestand (in plaats van uit het afvoerlijnen bestand).



Daarbij zijn 2 zaken van belang:

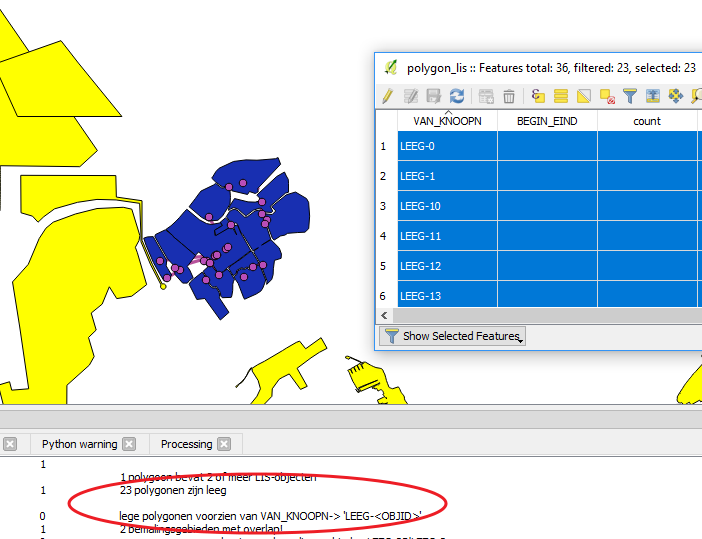
1. De RWZI moet in een eigen bemalingsgebied liggen.
2. De eindpunten van afvoerrelaties (meestal een rwzi) moeten topologisch (1 m nauwkeurig) aansluiten op een kikker knooppunt. Dat is normaal gesproken ook altijd het geval.

Het resultaat van deze analyse is te controleren in het tussenresultaat: **eindgebieden.shp (zie**

### Bemalingsgebieden zonder geldige afvoerrelatie

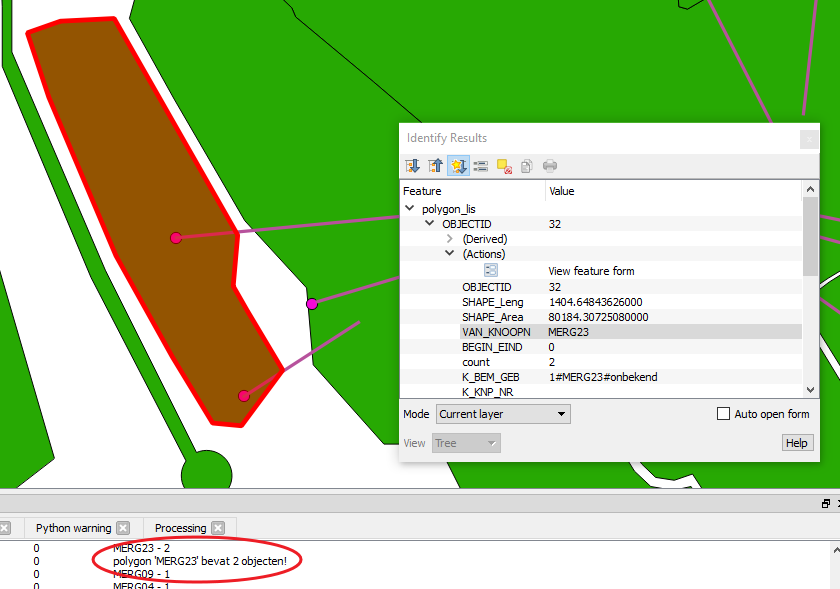
Voor bemalingsgebieden zonder geldige afvoerrelatie (die niet afvoeren buiten het bemalingsgebied) en ook geen eindgebied zijn (rwzi) wordt een unieke code gegenereerd. Dat is nodig als referentie in de vervolgstappen om het script goed te kunnen uitvoeren. In het logboek wordt melding gemaakt van het aantal “lege” bemalingsgebieden. Voor lege gebieden worden alleen berekeningen gedaan voor drinkwater, ve's en verhard opp, maar deze kunnen niet worden vergelijken met gegevens uit kikker.

Voor een goed eindresultaat is het wenselijk dat alle bemalingsgebieden een eigen knooppunt en afvoerrelatie hebben.



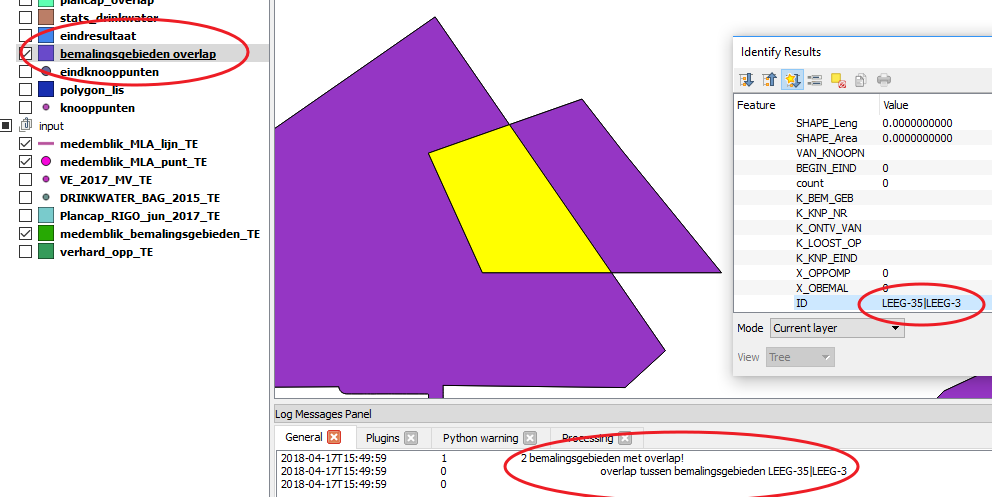
### 2 knooppunten in hetzelfde bemalingsgebied

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.

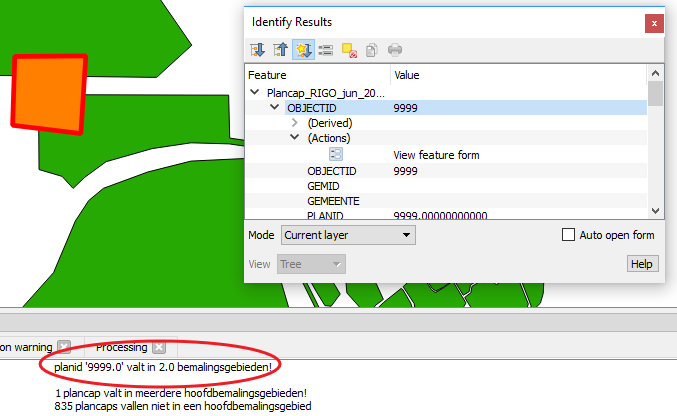


### Plancapaciteit in meerdere bemalingsgebieden

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

Denk erom dat de waardes voor extra afvoer in dat geval dubbel worden meegeteld!

Het is dus van belang om een aanpassing te maken in het bemalingsgebied of het woningbouwplan.



### Toelichting tussenresultaten

**Knooppunten.shp**

Alle begin- en eindpunten van afvoerrelaties

**Knooppunten\_sel1.shp**

Alle beginpunten van afvoerrelaties, aangevuld met rwzi knooppunten

**Knooppunten\_sel2.shp**

Alle eindpunten van afvoerrelaties

**Eindknooppunten.shp**

Alle eindpunten van afvoerrelaties ( VAN\_KNOO\_1), aangevuld met de code van het beginpunten van de afvoerrelaties als attribuut VAN\_KNOOPN

**Polygon\_kikker.shp**

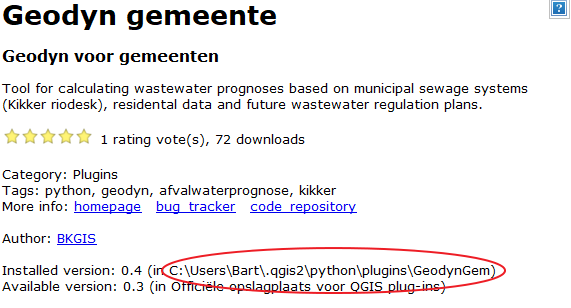
Eerste bemalingsgebied output met afvoerrelaties en knooppunt codes erin.

Tussenresultaten worden alleen bewaard met de volgende instelling in local\_settings.py b\_remove\_results\_after\_run = False (zie beheerdershandleiding – local settings)

# Beheerdershandleiding

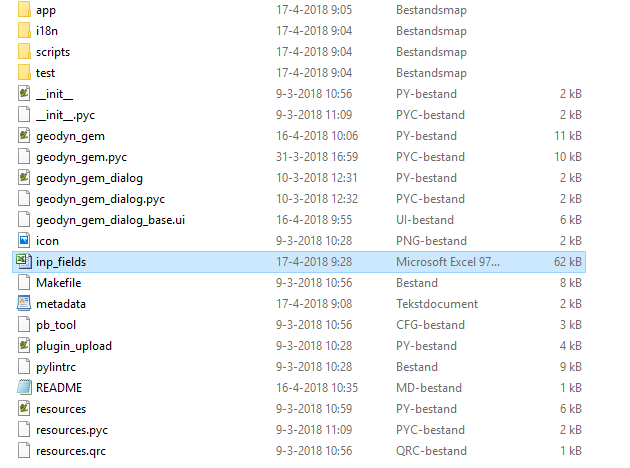
## Installatiemap plug-in

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



In deze map staan alle bestanden van de plug-in:

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



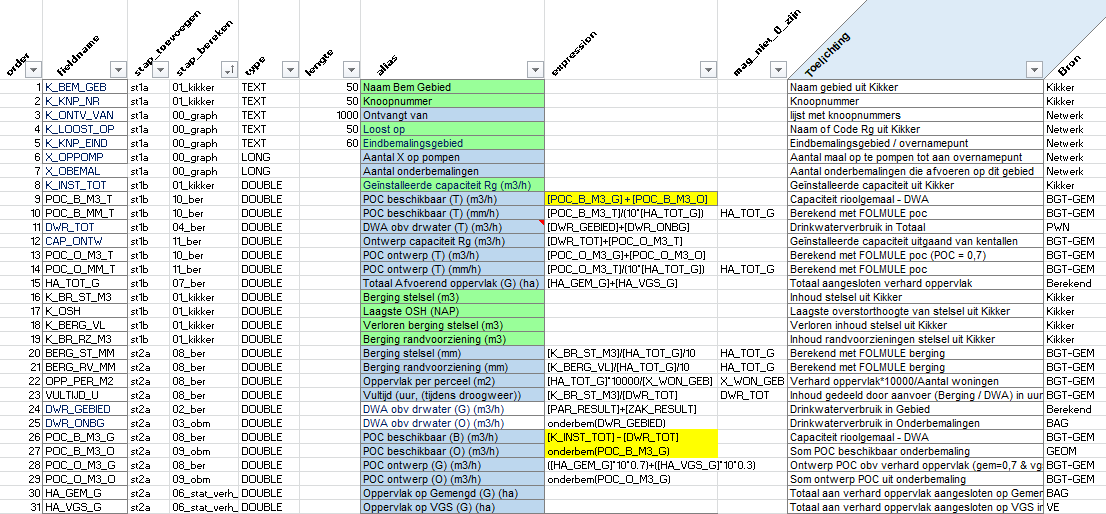
## Toelichting input velden

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

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

Let op! QGIS op iMac kan geen gebruik maken van het xls-bestand (“import xlrd error”).

Gebruiker van iMac moeten het bestand exporteren naar csv. Het inp\_fields.csv bestand wordt automatisch mee geïnstalleerd, maar wijzigen in Excel moeten handmatig worden doorgevoerd in het csv-bestand. (save as .csv met Excel)

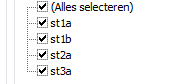


Hieronder volgt een korte toelichting van de betekenis van de velden:

**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.

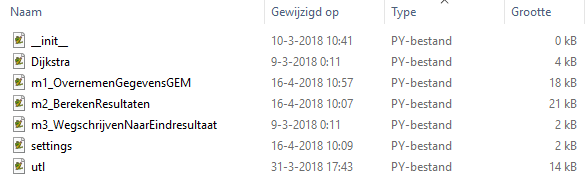
**Bron**: Ter verduidelijking van overzicht. Geeft aan wat de bron is van een veld. Vaak zijn het echter combinaties van bronnen.

## Toelichting python-scripts

In de rootfolder GeoDynGem staan een aantal py-scripts. De meeste daarvan zijn standaard voor iedere QGIS plug-in die met plug-in-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 kaart laag 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.

**Local\_settings.py** voor extra gebruikers-instellingen (zie volgende hoofdstuk)

**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.

## Extra instellingen in local\_settings.py

In het bestandje **local\_settings.py** zijn nog extra instellingen mogelijk.

Deze is te vinden in de app directory van de plug-in. De plug-in directory is te achterhalen via de plug-inmanager van qgis. De instelling zijn optioneel dus de plug-in werkt ook zonder local\_settings.py

Als het bestand nog niet bestaat kan een kopietje gemaakt worden van het l**ocal\_settings\_example.py**

**LOGGING\_LEVEL** staat standaard op INFO (20), maar kan ook op DEBUG ( 10) gezet worden. Met debug aan wordt veel meer informatie naar het log gestuurd.

**l\_result\_layers\_to\_remove** is de lijst met layer names die als resultaat te zien zijn in de layer panel in QGIS. Met de boolean kan worden aangegeven of het resultaat verwijderd moet worden of niet waarbij True staat voor verwijderen uit layer panel en False dus voor bewaren.

Met variabele **b\_remove\_results\_after\_run** kan ook worden ingesteld dat alle tussenresultaten bewaard moeten blijven.

In **result\_dir** kan een standaard resultatenmap worden opgegeven. Deze wordt dan bij het openen van de plug-in alvast weergegeven.

