===============================================================================================

Contents

[BEST approach 1](#_Toc11828057)

[Procedure to transform the municipality csv file to an Excel file 5](#_Toc11828058)

[Content of folder “Data” 10](#_Toc11828059)

[The result file 14](#_Toc11828060)

===============================================================================================

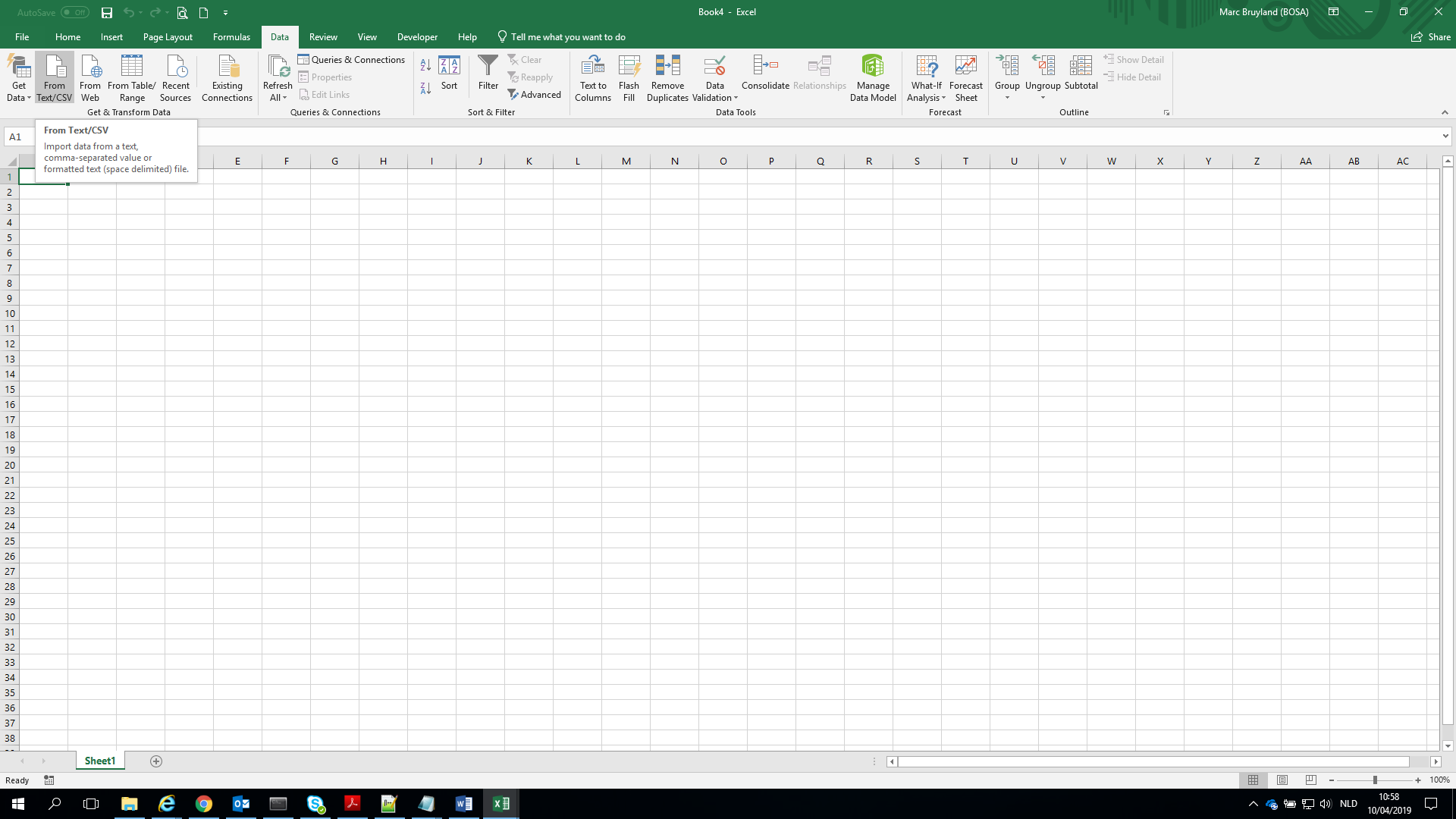
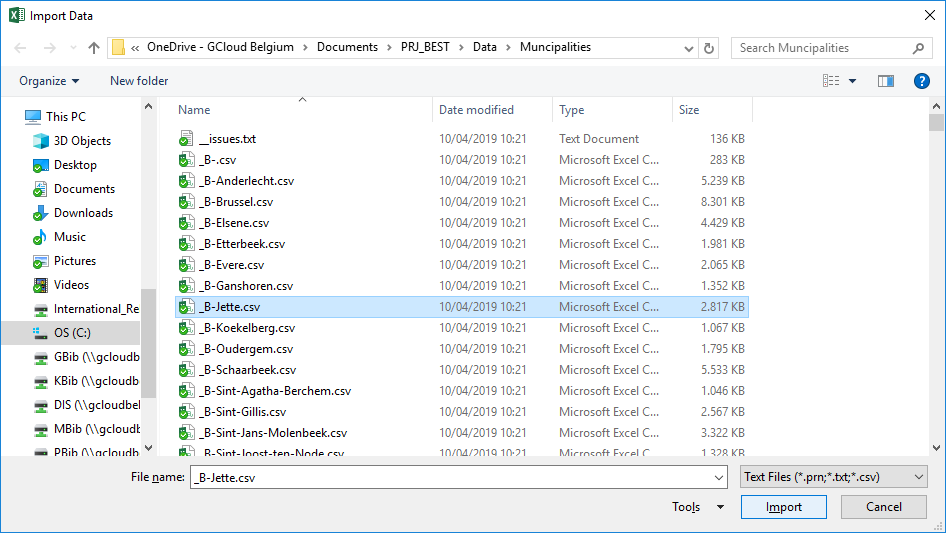
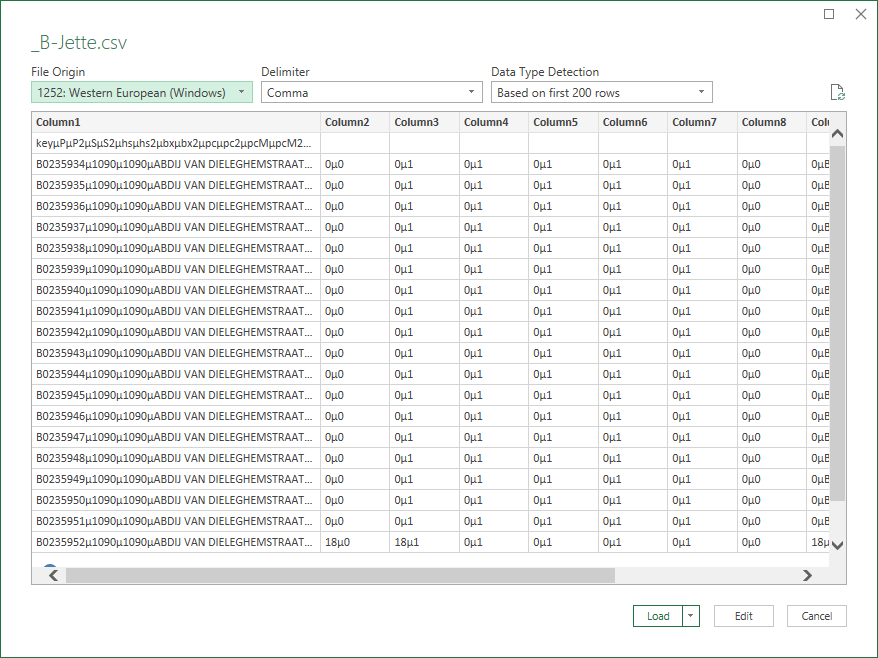
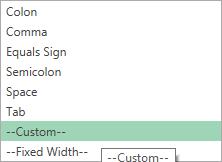
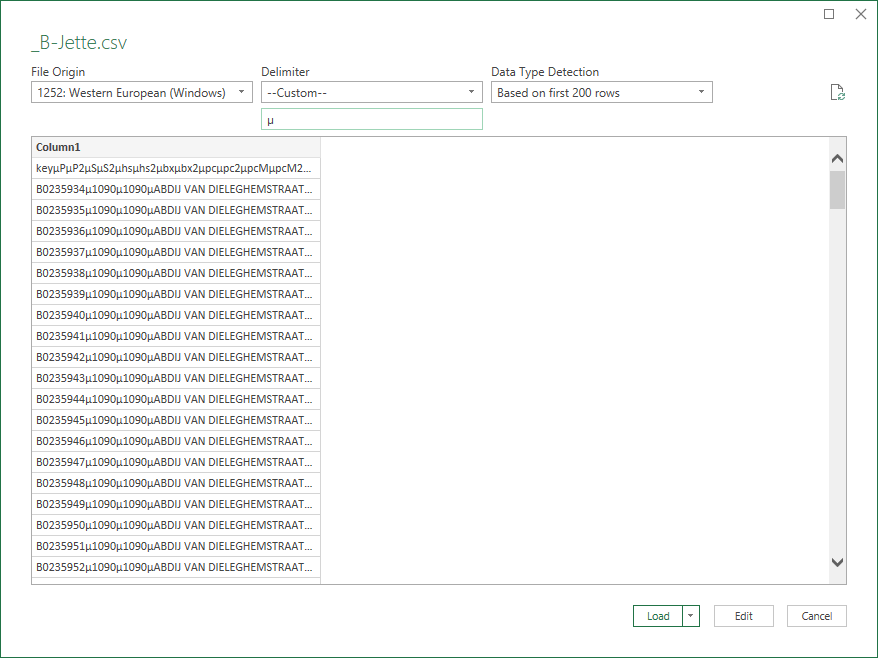
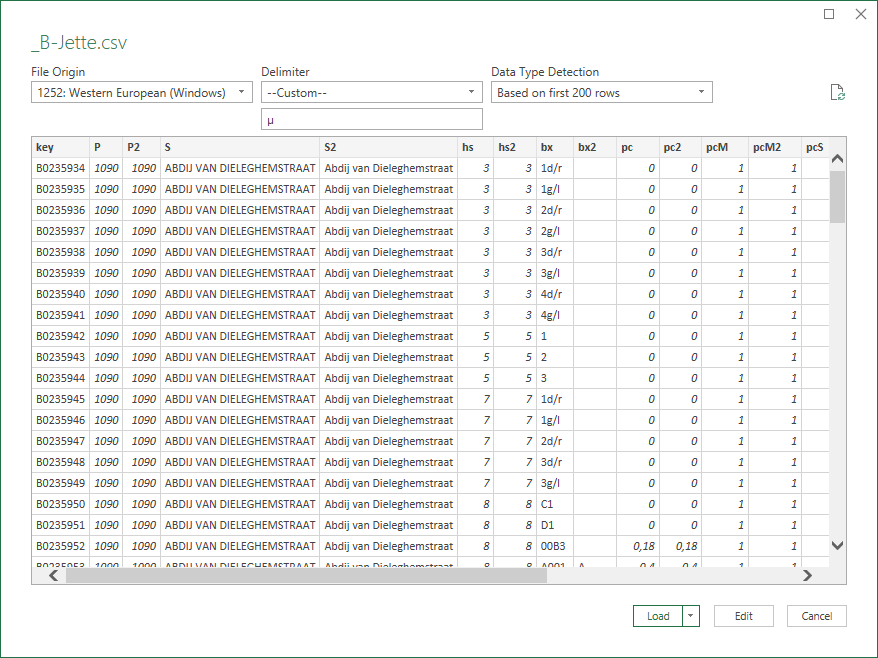
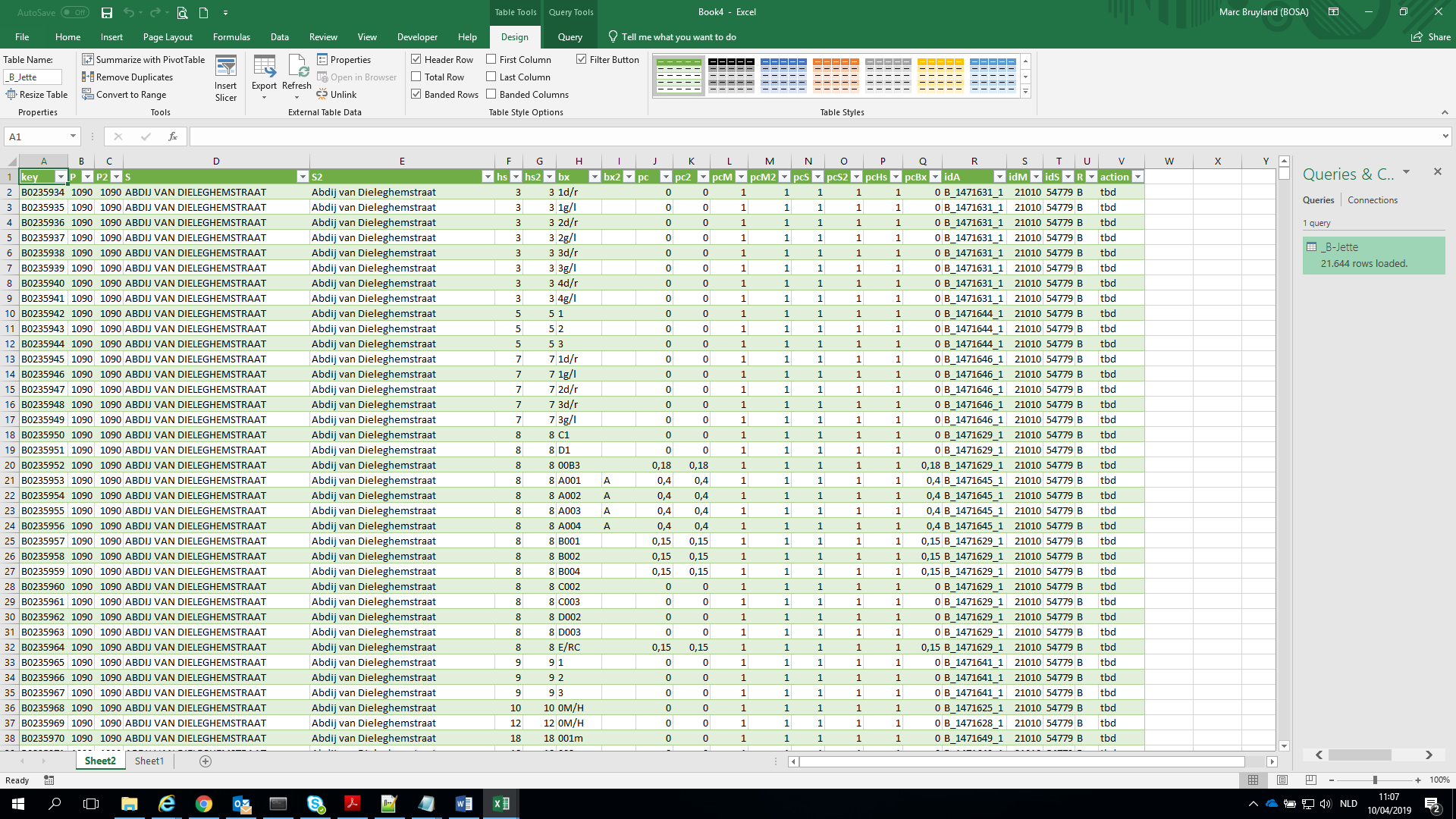
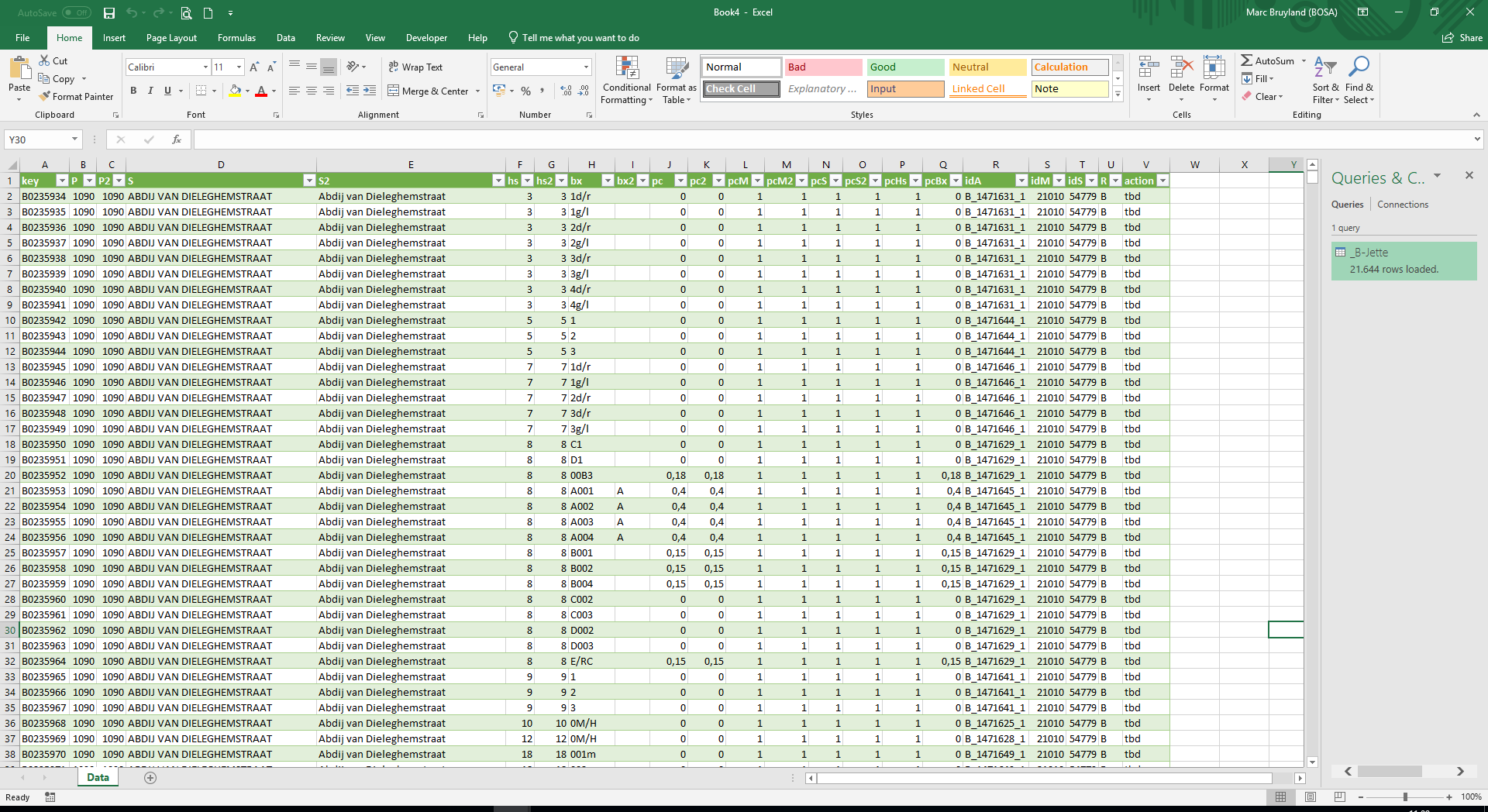
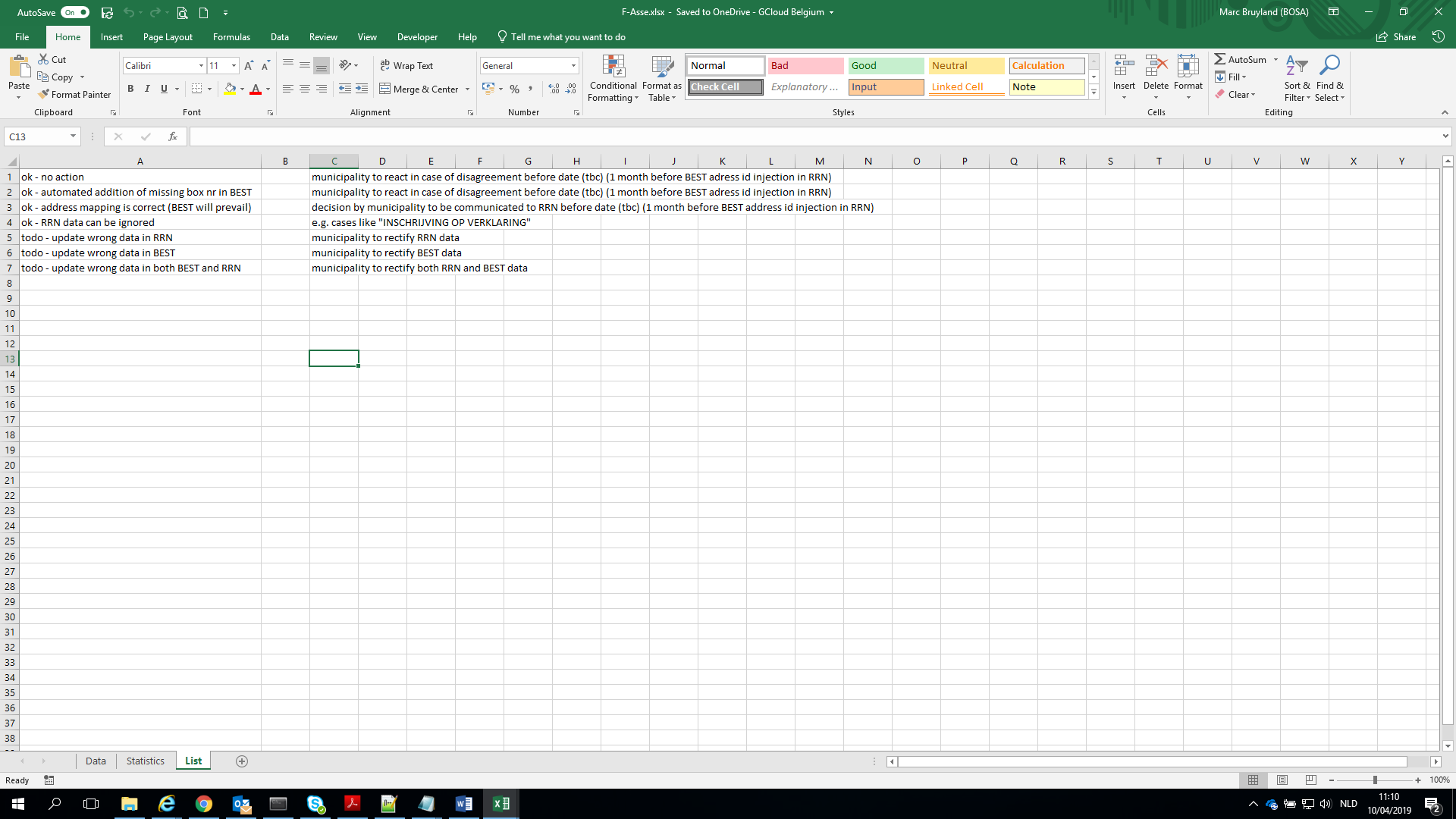
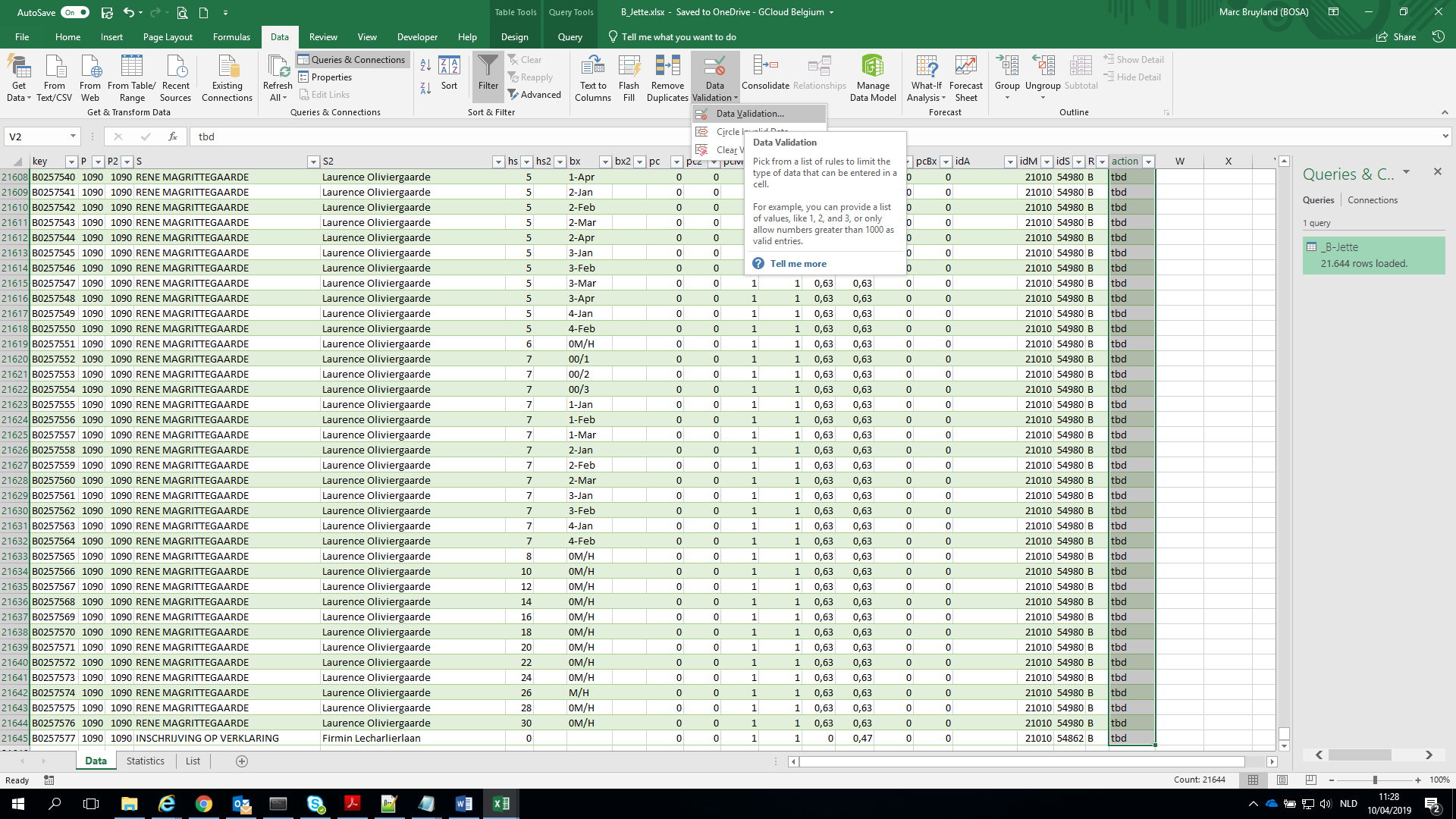
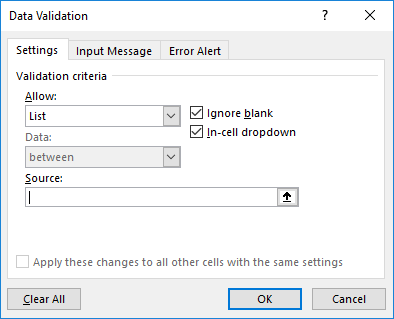
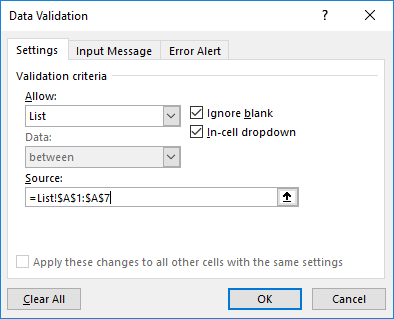
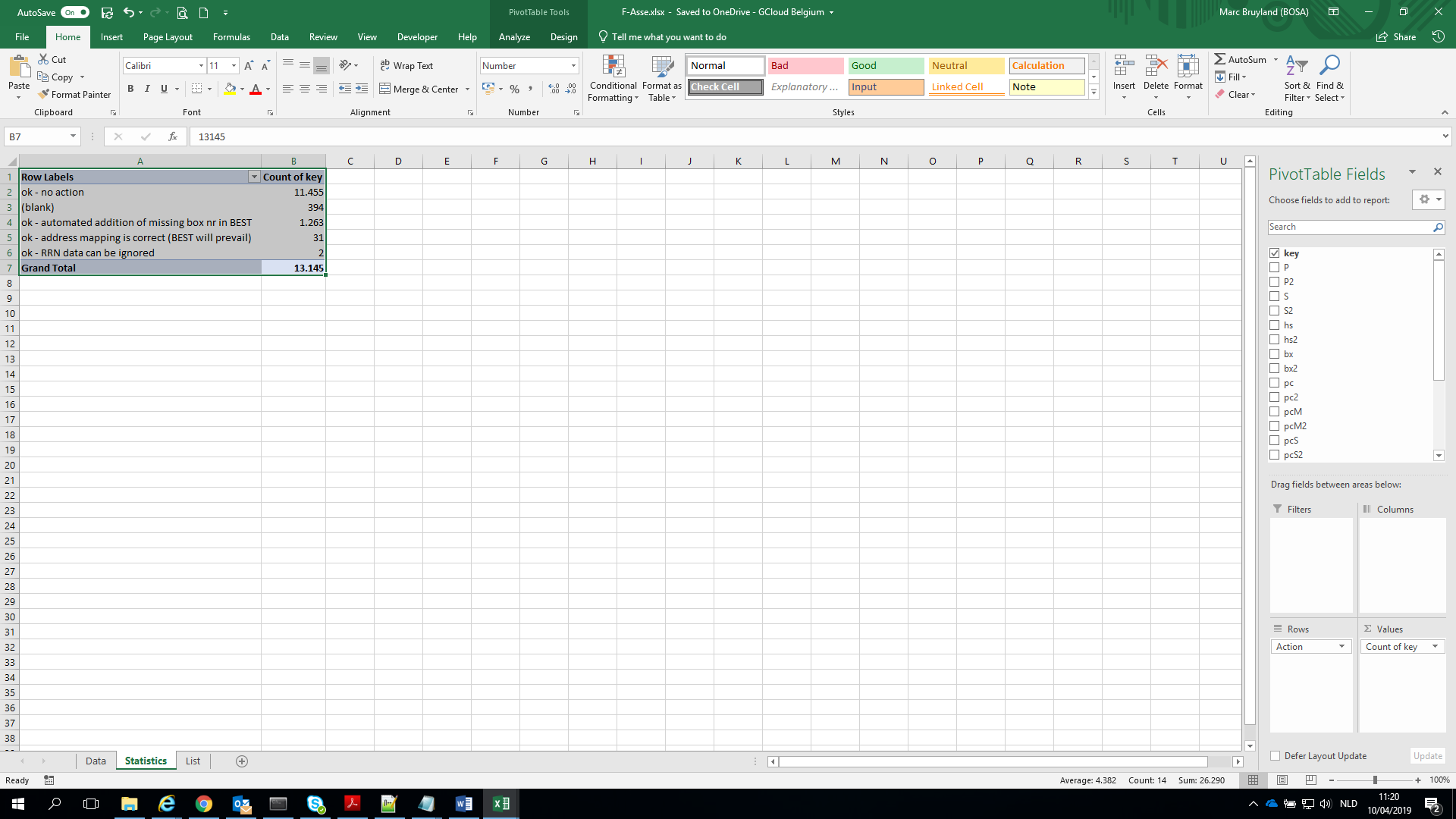
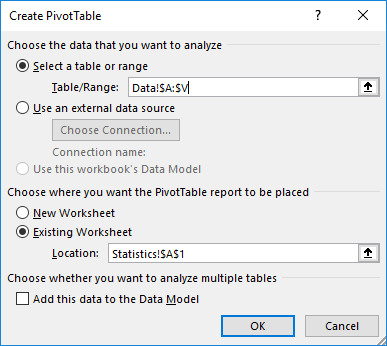
# BEST approach

* **Event: arrival of new xml files**
* unzip + copy following files with these filenames in Data folder:
  + BrusselsAddress.xml
  + BrusselsMunicipality.xml
  + BrusselsPartOfMunicipality.xml
  + BrusselsPostalinfo.xml
  + BrusselsStreetname.xml
  + FlandersAddress.xml
  + FlandersMunicipality.xml
  + FlandersPartOfMunicipality.xml
  + FlandersPostalinfo.xml
  + FlandersStreetname.xml
  + WalloniaAddress.xml
  + WalloniaMunicipality.xml
  + WalloniaPartOfMunicipality.xml
  + WalloniaPostalinfo.xml
  + WalloniaStreetname.xml
* launch python script “BEST\_XML\_Municipality.py“
  + Create “LST\_Municipalities.txt”
  + Create “DIC\_Municipalities.txt”
  + Create “OUT\_Municipalities.txt”
  + Create “STAT\_Municipalities.txt”
* launch python script “BEST\_XML\_ PartOfMunicipality.py”
  + Create “LST\_PartOfMunicipalities.txt”
  + Create “DIC\_PartOfMunicipalities.txt”
  + Create “OUT\_PartOfMunicipalities.txt”
  + Create “STAT\_PartOfMunicipalities.txt”
* launch python script “BEST\_XML\_Streetname.py”
  + Create “LST\_Streetnames.txt”
  + Create “DIC\_ Streetnames.txt”
  + Create “OUT\_ Streetnames.txt”
  + Create “STAT\_ Streetnames.txt”
  + *Stationstraat*
  + *Streetnames of Asse*
  + *Status*
  + *streetnameType*
  + *homonym*
* launch python script “BEST\_XML\_ PostalInfo.py”
  + Create “LST\_PostalInfo.txt”
  + Create “DIC\_PostalInfo.txt”
  + Create “OUT\_ PostalInfo.txt”
  + Create “OUT\_Postcodes.txt”
  + Create “STAT\_PostalInfo.txt”
* launch python script “BEST\_XML\_ Address.py”
  + Create “LST\_Addresses.txt”
  + Create “DIC\_ Addresses.txt” (full)
  + Create “STAT\_Addresses.txt”
  + Create “DIC\_ Addresses100.txt” (1%)
  + Create “DIC\_MtoP.txt” (Municipality to PostalInfo)
  + Create “DIC\_MtoPM.txt” (Municipality to PartOfMunicipality)
  + Create “DIC\_PtoM.txt” (PostalInfo to Municipality)
  + Create “DIC\_PtoPM.txt” (PostalInfo to PartOfMunicipality)
  + Create “DIC\_PMtoM.txt” (PartOfMunicipality to Municipality)
  + Create “DIC\_PMtoP.txt” (PartOfMunicipality to PostalInfo)
  + Create “OUT\_PartOfMunicipalities\_Link with Municipality.txt”
* launch python script “BEST\_MappingFiles.py”
  + Create “MAP\_Streets.txt (R+idM ⇨ {streetname: idS})
  + Create “MAP\_StreetsRR.txt” (removal of special characters + uppercase)
  + Create “MAP\_Addresses.txt” (R+IdS ⇨ {numbers: addressId})
  + Create “MAP\_HouseNrsRR.txt” (R+idS ⇨{ hsNum:hs})
  + Create “MAP\_BoxNrsRR.txt” (R+idS+”\_”+hsNum ⇨ {hsNonNum+bx:idA})
  + Create “MAP\_MunToR.txt” ({idm:R})
  + Create “MAP\_StreetCode\_RRtoBEST.txt” ({R{NISM{codeS{idS, Snl, Sfr, Sde, pcS, pcS2}}}})
    - NISM = NIS-code municipality RR, codeS=code street RR, idS = id street BEST, pcS = mapping probability streetname, pcS2 = mapping probability modified streetname
* **Event: arrival of RR files for mapping**
* Unzip RR\_B, RR\_W, RR\_F files
* launch python script “BEST\_PrepareRRFiles.py”
  + Create “RR\_B.txt” (add key)
  + Create “RR\_W.txt” (add key)
  + Create “RR\_F.txt” (add key)
* launch python script “BEST\_ MapRR\_B.py” (mapping RR onto BEST)
  + Create “RR\_B\_Result.txt”
* launch python script “BEST\_ MapRR\_W.py” (mapping RR onto BEST)
  + Create “RR\_W\_Result.txt”
* launch python script “BEST\_ MapRR\_F.py” (mapping RR onto BEST)
  + Create “RR\_F\_Result.txt”
* launch python script “BEST\_MunicipalityFilesRR\_B.py”
  + Create files “\_B-MunicipalityName.csv” where MunicipalityName = name of Municipality
* launch python script “BEST\_MunicipalityFilesRR\_F.py”
  + Create files “\_F-MunicipalityName.csv” where MunicipalityName = name of Municipality
* launch python script “BEST\_MunicipalityFilesRR\_W.py”
  + Create files “\_W-MunicipalityName.csv” where MunicipalityName = name of Municipality
* **Event: arrival of KBO files for mapping**
* launch python script “BEST\_PrepareKBOFiles.py”
  + Create “KBO.txt” (add key)
* launch python script “BEST\_ MapKBO.py” (mapping KBO onto BEST)
  + Create “KBO\_Result.txt”
* launch python script “BEST\_MunicipalityFilesKBO.py”
  + Create files “\_R-MunicipalityName.csv” where R = region, MunicipalityName = name of Municipality
* **Event: arrival of Police files for mapping**
* launch python script “BEST\_PreparePOLFiles.py”
  + Create “POL.txt” (add key)
* launch python script “BEST\_MappingFilesPOL.py”
  + Create “MAP\_StreetCode\_POLtoBEST.txt” ({R{NISM{codeS{idS, Snl, Sfr, Sde, pcS, pcS2}}}})
    - NISM = NIS-code municipality RR, codeS=idStreet POL, idS = id street BEST, pcS = mapping probability streetname, pcS2 = mapping probability modified streetname
* launch python script “BEST\_ MapPOL.py” (mapping Police onto BEST)
  + Create “POL\_Result.txt”
* launch python script “BEST\_MunicipalityFilesPOL.py”
  + Create files “\_R-MunicipalityName.csv” where R = region, MunicipalityName = name of Municipality
* **Event: arrival of AAPD (Kadaster MinFin) files for mapping**
* launch python script “BEST\_PrepareAAPDFiles.py”
  + Create “AAPD.txt” (add key)
* launch python script “BEST\_MappingFilesAAPD.py”
  + Create “MAP\_StreetCode\_AAPDtoBEST.txt” ({R{NISM{codeS{idS, Snl, Sfr, Sde, pcS, pcS2}}}})
    - NISM = NIS-code municipality RR, codeS=idStreet AAPD, idS = id street BEST, pcS = mapping probability streetname, pcS2 = mapping probability modified streetname
    - particularity with streetNames in AAPD: uppercase + no acccents + lots of abbreviations
* launch python script “BEST\_ MapAAPD.py” (mapping AAPD onto BEST)
  + Create “AAPD\_Result.txt”
* launch python script “BEST\_MunicipalityFilesAAPD.py”
  + Create files “\_R-MunicipalityName.csv” where R = region, MunicipalityName = name of Municipality
* **Tools**
* BEST\_Tools\_ExtractFromResultsFile.py
  + required parameters: src, one or more selection criteria
    - src in ['AAPD', 'RR\_B', 'RR\_F', 'RR\_W', 'KBO', 'POL']
    - selection criteria:
      * either 'idA' for selecting correct address id's
      * or ['A', 'A1', 'A2', 'B', 'B1', 'B2, 'C', 'C1', 'C2', 'D', 'D1', 'D2', 'D3'] for selecting one or more warnings
  + example 1: BEST\_Tools\_ExtractFromResultsFile.py POL idA will extract correct BEST address id's from the results file of the Police
  + example 2: BEST\_Tools\_ExtractFromResultsFile.py POL A B2 will extract warnings A1, A2 and B2 from the results file of the Police
* BEST\_Tools\_SelectLineFromFile.py
  + required parameters: filename, one or more selection criteria
  + example 1: python BEST\_Tools\_SelectLineFromFile.py BrusselsAddress.xml Nieuwstraat will extract all lines from file BrusselsAddress.xml that contain 'Nieuwstraat'
  + example 2: python BEST\_Tools\_SelectLineFromFile.py AAPD.txt Nieuwstraat 22 will extract all lines from file AAPD.txt that contain both 'Nieuwstraat' and '22'
  + example 3: python BEST\_Tools\_SelectLineFromFile.py POL.txt Nieuwstraat \"hs\":\"22\" will extract all lines from file POL.txt that contain both 'Nieuwstraat' and '"hs":"22"'
* BEST\_Tools\_Get\_Numbers.py
  + required parameters: key [RR]
  + key for house numbers: R+idS
  + key for box numbers: R+idS\_hs"
  + RR is an optional parameter: this will allow to use a deviating mapping file specially for the RR
  + example 1: BEST\_Tools\_GetNumbers.py B56724 will extract all house nrs for street id 56724 in region B
  + example 2: BEST\_Tools\_GetNumbers.py B57022\_28 will extract all box nrs for house nr 28 in street id 57022 in region B
* BEST\_Tools\_GetById
  + with this tool you can get the details of an address or address component based on an id
    - Address id, Municipality id, Part of Municipality id, Postal code, Street id
    - condition: presence of the following dictionaries created from the BEST xml full download files:
      * DIC\_Addresses.txt
      * DIC\_MtoP.txt
      * DIC\_MtoPM.txt
      * DIC\_Municipalities.txt
      * DIC\_PartOfMunicipalities.txt
      * DIC\_PMtoM.txt
      * DIC\_PMtoP.txt
      * DIC\_PostalInfo.txt
      * DIC\_PtoM.txt
      * DIC\_PtoPM.txt
      * DIC\_Streetnames.txt
* TODO: launch python script “BEST\_Tools.py”
  + makeExtractAddressesOfStreet(dicA, R, idM, idS)
    - Create “Adresses\_F\_23002\_24377.txt” (all addresses in a street)
  + showStringInFile(filename, '72043')
* not needed:
  + Create “MAP\_Municipalities.txt” (postcode+Mun.name/Gehucht/PartOfMun ⇨ region + idM)
  + Create “MAP\_MunicipalitiesRR.txt” (postcode ⇨ list[ (R,idM)])
  + Create “MAP\_MunicipalitiesStreetsRR.txt” (postcode+Streetname ⇨ idM ev. StreetName BIS))

# Procedure to transform the municipality csv file to an Excel file

* import csv file
* rename sheet2 ⇨ Data + rename table Data
* rename sheet 1 ⇨ Statistics
* add sheet List
* [sheet Statistics]
  + insert pivot (table = Data)
  + add key, action – format numeric cells
  + select data + insert pie
    - change name ‘Total’ to Municipality name
    - add data labels
* [sheet List]
  + copy following :

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **action** |  |  |  | INDIRECT("Actions[Action]") |
| ok - no action |  | municipality to react in case of disagreement before date (tbc) (1 month before BEST adress id injection in RRN) |  |  |
| ok - automated addition of missing box nr in BEST |  | municipality to react in case of disagreement before date (tbc) (1 month before BEST adress id injection in RRN) |  |  |
| ok - address mapping is correct (BEST will prevail) |  | decision by municipality to be communicated to RRN before date (tbc) (1 month before BEST address id injection in RRN) |  |  |
| ok - RRN data can be ignored |  | e.g. cases like "INSCHRIJVING OP VERKLARING" |  |  |
| todo - update wrong data in RRN |  | municipality to rectify RRN data |  |  |
| todo - update wrong data in BEST |  | municipality to rectify BEST data |  |  |
| todo - update wrong data in both BEST and RRN |  | municipality to rectify both RRN and BEST data |  |  |

* [sheet Data]
  + select data under column “action”
  + Data Validation
    - List
    - INDIRECT("Actions[Action]")
* Open Excel
  + Data – From Text/CSV
  + 
  + Import CSV file
* 
* 
  + change Delimiter ‘Comma’ to '$' ~~‘µ’~~ via Custom
  + 
  + 
* 
  + Click button ‘Load’
* 
* delete Sheet1, rename Sheet2 to Data
* 
* Create sheet “List” with the following content
* 
* select all cells under column header ‘action’ (these cells are filled with ‘tbd’)
  + select Data validation
* 
  + Select ‘List’
  + 
  + Select Source
  + 
* Create sheet “Statistics” with the following pivot table
*  

# Content of folder “Data”

BEST-MappingProcedure.docx

BEST\_ConsolidatedResultForBEST.py

BEST\_Lib.py

BEST\_MapAAPD.py

BEST\_MapKBO.py

BEST\_MappingFiles.py

BEST\_MappingFilesAAPD.py

BEST\_MappingFilesPOL.py

BEST\_MapPOL.py

BEST\_MapRR\_B.py

BEST\_MapRR\_F.py

BEST\_MapRR\_W.py

BEST\_MunicipalityFilesAAPD.py

BEST\_MunicipalityFilesKBO.py

BEST\_MunicipalityFilesPOL.py

BEST\_MunicipalityFilesRR\_B.py

BEST\_MunicipalityFilesRR\_BFW.py

BEST\_MunicipalityFilesRR\_F.py

BEST\_MunicipalityFilesRR\_W.py

BEST\_PrepareAAPDFiles.py

BEST\_PrepareKBOFiles.py

BEST\_PreparePOLFiles.py

BEST\_PrepareRRFiles.py

BEST\_ToolsTestBoxNrs.py

BEST\_Tools\_CountLinesInFile.py

BEST\_Tools\_EncodeDecode.py

BEST\_Tools\_ExtractFromResultsFile.py

BEST\_Tools\_GetById.py

BEST\_Tools\_GetNumbers.py

BEST\_Tools\_PrepareTestbestand.py

BEST\_Tools\_SelectLineFromFile.py

BEST\_XML\_Address.py

BEST\_XML\_Municipality.py

BEST\_XML\_PartOfMunicipality.py

BEST\_XML\_PostalInfo.py

BEST\_XML\_Streetname.py

**BrusselsAddress.xml**

**BrusselsMunicipality.xml**

**BrusselsPartOfMunicipality.xml**

**BrusselsPostalinfo.xml**

**BrusselsStreetname.xml**

DIC\_Addresses.txt

DIC\_MtoP.txt

DIC\_MtoPM.txt

DIC\_Municipalities.txt

DIC\_PartOfMunicipalities.txt

DIC\_PMtoM.txt

DIC\_PMtoP.txt

DIC\_PostalInfo.txt

DIC\_PtoM.txt

DIC\_PtoPM.txt

DIC\_Streetnames.txt

DOC\_Address.docx

DOC\_Address.pdf

DOC\_Municipality.docx

DOC\_Municipality.pdf

DOC\_PartOfMunicipality.docx

DOC\_PartOfMunicipality.pdf

DOC\_PostalInfo.docx

DOC\_PostalInfo.pdf

DOC\_StreetName.docx

DOC\_StreetName.pdf

**FlandersAddress.xml**

**FlandersMunicipality.xml**

**FlandersPartOfMunicipality.xml**

**FlandersPostalinfo.xml**

**FlandersStreetname.xml**

LST\_Addresses.txt

LST\_Municipalities.txt

LST\_PartOfMunicipalities.txt

LST\_PostalInfo.txt

LST\_Streetnames.txt

**MAP\_Addresses.txt**

**MAP\_BoxNrsRR.txt**

**MAP\_HouseNrsRR.txt**

**MAP\_Municipalities.txt**

**MAP\_MunicipalitiesRR.txt**

**MAP\_MunicipalitiesStreetsRR.txt**

**MAP\_Streets.txt**

**MAP\_StreetsRR.txt**

OUT\_hsBx.txt

OUT\_Mun-PartOfMun-PostalInfo.txt

OUT\_Municipalities.txt

OUT\_PartOfMunicipalities.txt

OUT\_PostalInfo.txt

OUT\_Postcodes.txt

OUT\_Streetnames.txt

**RR\_B.txt**

**RR\_B\_result.txt**

**RR\_F.txt**

**RR\_F\_result.txt**

**RR\_W.txt**

**RR\_W\_result.txt**

**WalloniaAddress.xml**

**WalloniaMunicipality.xml**

**WalloniaPartOfMunicipality.xml**

**WalloniaPostalinfo.xml**

**WalloniaStreetname.xml**

# The result file

* The result file is composed of a number of lines, each containing a dictionary of key-value pairs.
* The significance of the keys (red means the data comes from the source file, green means the data is added by the mapping program):
* key: reference to the source file
* idM\_SRC: id municipality
* M: municipality name
* Mnl: municipality name in dutch
* Mfr: municipality name in french
* Mde: municipality name in german
* M2: municipality name BEST (either in dutch, french or german)
* P: postal code
* idS\_SRC: is street
* S: street name
* Snl: street name in dutch
* Sfr: street name in french
* Sde: street name in german
* idS: id street
* S2: streetname (either in dutch, french or german)
* hs: house number
* hs2: house number
* lstHs2: list of BEST house numbers that can be mapped on the numeric (only) house number in the RR e.g. ['2A', '2B']
* bx: box number (RR: index)
* bx2: box number
* pc: overall probability for exact match of address (without 'tricks')
* pc2: overall probability for exact match of address (with 'tricks')
* pcM: probability for exact match of the municipality
* pcS: probability for exact match of the street (without 'tricks')
* pcS2: probability for exact match of the street (with 'tricks')
* pcHs: : probability for exact match of the house number
* pcBx: probability for exact match of the box number
* idA: id of address (can only be injected in the source file if 'actionOk')
* idM: id of municipality
* idS: id street
* R: region (F=Flanders, W=Wallonia, B=Brussels)
* warningA: warning regarding municipality
  + warningA1 = 'A1. ISSUE Municipality irrelevant'
  + warningA2 = 'A2. ISSUE Municipality unknown'
* warningB: warning regarding street
  + warningB1 = 'B1. ISSUE Street irrelevant'
  + warningB2 = 'B2. ISSUE Street mapping'
* warningC: warning regarding house number
  + warningC1 = 'C1. ISSUE Nrs (Str.without nrs in BEST)'
  + warningC2 = 'C2. ISSUE Nrs (Hs <>)'
* warningD: warning regarding box number
  + warningD1 = 'D1. ISSUE Nrs (Hs =, Bx missing in BEST)'
  + warningD2 = 'D2. ISSUE Nrs (Hs =, Bx <>)'
  + warningD3 = 'D3. ISSUE Nrs (Hs =, Bx missing in SRC)'
* action:
  + actionOk = 'ok - no action'
  + actionA1 = 'ok - data can be ignored'
  + actionB1 = 'ok - data can be ignored'
  + actionD1 = 'ok - automated addition of missing box nr in BEST'
  + 'tbd' = default action for further manual examination