The excel interface tool is an Excel-based database/script program for MAIZSIM that holds the data for 1 or more simulations and builds the input files and file structure to run the model.

The scripts are written in VBA and are located in the VBA Code section of the Excel file (turn on the Developer menu item). Code and data are also included to automatically build the grid files needed for the finite element domain. Each input file has its own sheet. The only files that are not created are the weather files. These must be done separately.

The executables and secondary files needed are described below.

The following files are used to create the grid and soil files. These are not executed by the vba code but are executed in a separate DOS command file:

CreateSoilFIles.exe This program calls GridGen.DLL and Rosetta.exe

GridGen.DLL a fortran dll that creates the finite element mesh

Rosetta.exe A program that uses pedotransfer functions and a neural net to estimate soil hydraulic properties from soil texture data. The program (source code was obtained from:

<https://www.ars.usda.gov/pacific-west-area/riverside-ca/agricultural-water-efficiency-and-salinity-research-unit/docs/model/rosetta-model/>

The program was mostly written by Marcel Schapp

The following files are needed by the Excel VBA code:

Dispersivity Lookup.xls This contains one sheet that is a lookup table for soil dispersivity vs soil texture class. It should be in the same folder as the ExcelInterface file.

TextureClass(32).dll This program takes sand, silt and clay values and gives the texture class (silty clay loam, etc). It was originally written by Aris Gerakis, 2001 with help from Brian Baer

(<https://nowlin.css.msu.edu/software/triangle_form.html>)

TextureClass.dll finds the soil texture class and this is used to lookup the correct value of dispersivity in the Dispersivity lookup file.

CreateSoilFiles.exe usage:

The excel interface creates a batch file called grid1.bat. This is an example of the contents:

D:\Maizsim07\CreateSoils\CreateSoilFiles.exe "D:\MAIZSIM07\AgMipEt\Iowa06\Iowa06.lyr" /GN Iowa06 /SN Harps

del output

del element\_elm

del grid\_bnd

del datagen2.dat

Dir \*.\* >dir.txt

To use CreateSoilFiles.exe on its own. The command is called from the path where soil and grid files should be stored. The command line requires a fully formed path with the executable’s source because the executable is in a different folder than the one where the files are created. The input file (…lyr) also needs a fully formed path. The arguments GN and SN are the grid name and soil name.

Once the grid1.bat files are created in all the subdirectories, you can iterate through the subdirectories and call all the grid files with this script:

*for /R "d:\maizsim07\AgMipLowInputC" %%g in (.) do (*

*pushd %%g*

*echo now in %%g*

*grid1.bat*

*popd*

*)*

Save this into a file called, for example, RunGridBatchFiles.bat

Substituted your file structure for “D:\maizsim07\AgMipLowInputC”

Run the batch file from the root directory (D:\maizsim07 \AgmipLowInputC) in this case).

Excel file sheets (tables)

The first sheet is the Description. This sheet contains the id’s that link the various tables and are used for lookup. The number of rows in this table has to be equal to the number of simulations to be carried out. Also, the filenames for the output files can be specified here.

The ID is the main identifier and there is one for each simulation. There is no requirement other than each is unique.

The linkages among the rows of data in the sheets are as follows:

[Description]WeatherID

[Description]ClimateID [Climate]ClimateID

[Description]SoilFIle [Soil]SoilName

[Description]Hybrid [Variety]Hybrid

[Description]Biology [Biology]BiologyID

[Description]Solute [Solute]ID

The variable [Description]path is the path name for the simulation files in the directory structure.

The first page (sheet) of the excel interface has a form for entry of the parameters needed to run the interface. The parameters are file names and locations.

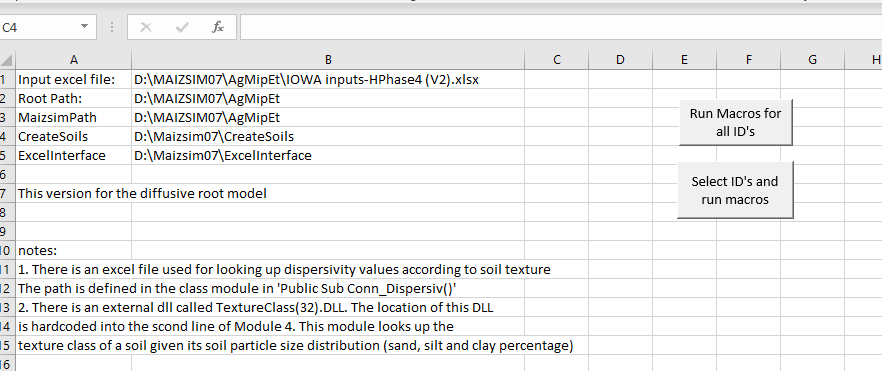
The “input excel file” is the excel file with the data for the simulations

Root path is the path below which the paths for the simulations exist. Each simulation has its own path

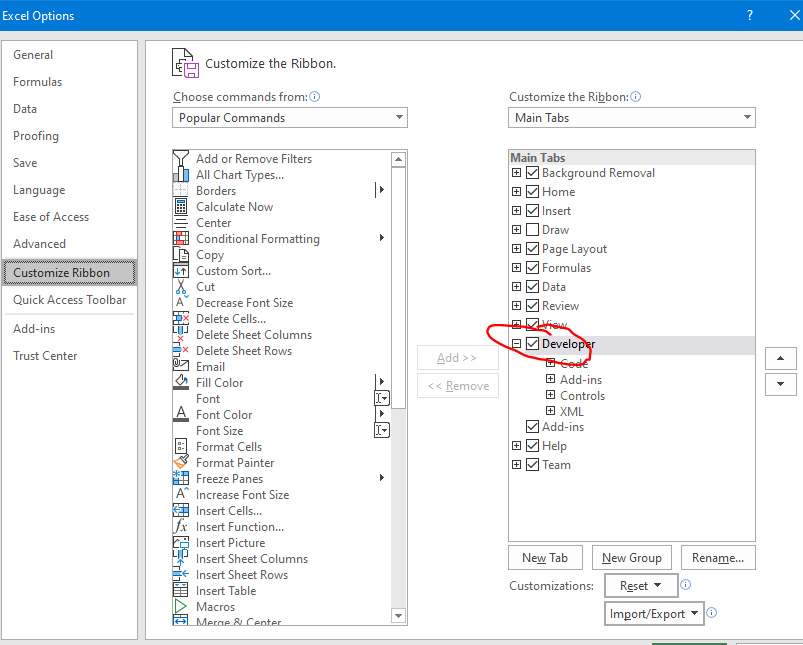
The maizsim path is the path where the 2dsoil.exe file exists, usually the same as the root path

Create soils and Excel interface are the locations of the CreateSoilFiles.exe and the excel interface file (read plant filesV4.xlsm).

If you click the select ID’s and run macros button, the column C will fill with ID’s and a selection window will open allowing you to select one or more of the sims to run. The ‘Run Macros for all ID’s’ is not fully functional yet.



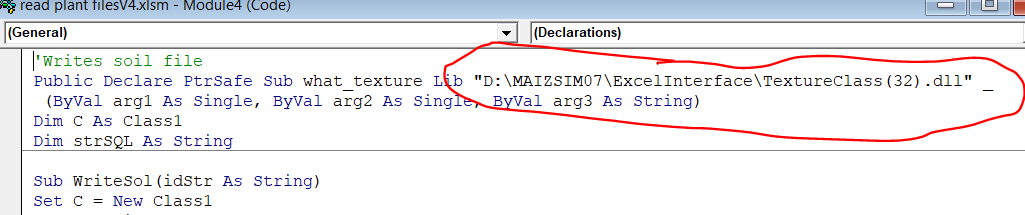
You’ll have to set the folders for the helper module triangle. Choose ‘Developer’ from the menu. If you don’t see developer, right click in the ribbon and choose ‘customize the ribbon’ Then enable developer to the right. It is not checked by default. If necessary, close that and then click on the visual basic icon on the left. This will bring up the code editor in a new window.



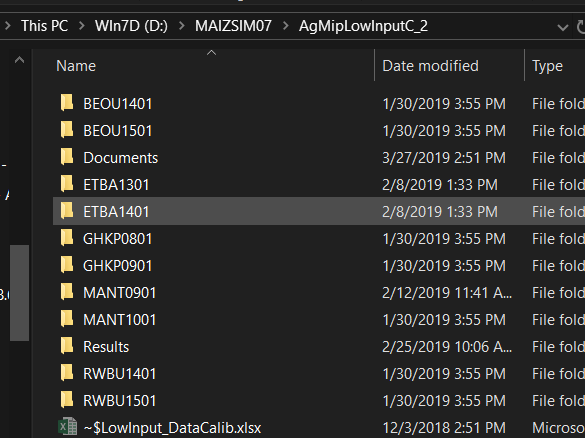
This is a summary of the modules

|  |  |
| --- | --- |
| Module 1 | Imports a plant output file and inserts into spreadsheet |
| Module 2 | Writes the biology file |
| Module 3 | Writes the initialization file |
| Module 4 | Writes the solute file - calls a dll to find soil texture class and also opens  Another spreadsheet to look up dispersivity based on texture class |
| Module 5 | Writes management files |
| Module 6 | Writes the time file |
| Module 7 | Writes Variety file |
| Module 8 | Climate file |
| Module 9 | Nitrogen file |
| Module 10 | Writes the run file |
| Module 11 | Creates subfolders |
| Module 12 | Runs macros for all ID''s as a main calling program |
| Module 13 | Runs macros for selected id's - choose them from the excel page |
| Module 14 | Creates the soil files |
| Module 15 | Creates drip irrigation files |

Open module 4 and change the location for TextureClass(32).dll



Here the base folder is D:\maizsim07\AgmipLowInputC\_2 and the interface created the paths BEOU1401, etc and filled them with the input files.



This is what a folder looks like after it is filled with files.

