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. Because the texture class dll used to determine texture class to use in the lookup of tortuosity for solute transport only works in the 32 bit version of excel, I am not using it at this time but use an average value. Thus, the texture class dll is not needed at this time.

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 weather files are also created. See below in this document for more information.

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. This version works with the current version of 2dmaizsim.

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. (note that this program is not used currently until I can build a 64 bit version)

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

You can also run the 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

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

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

Installation:

Create a folder to hold your maizsim work. Under this folder create two sub-folders:

ExcelInterface and CreateSoils

In the ExcelInterface folder copy the files from the same named folder in the shared google drive or zip file

read plant filesV9\_mulch (integrated).xlsm

How to use the Excel tool.docx

Folder structure.docx

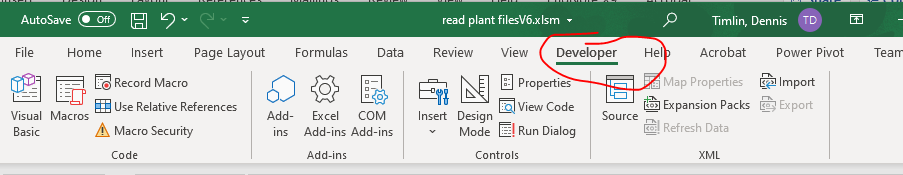
Textureclass(32).dll

Dispersivity lookup.xls

You have to edit the excel VBA program so it can find the helper files

Note that you currently do not need to do this step

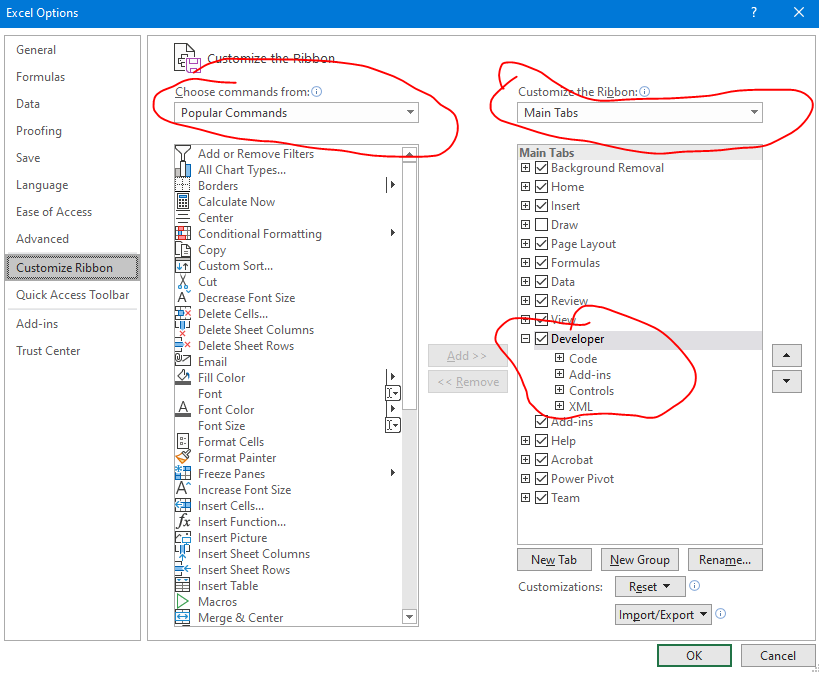
Open the file read plant filesV9\_mulch (integrated).xlsm and enable the developer tab if it is not already



It is not enabled by default, to enable it:

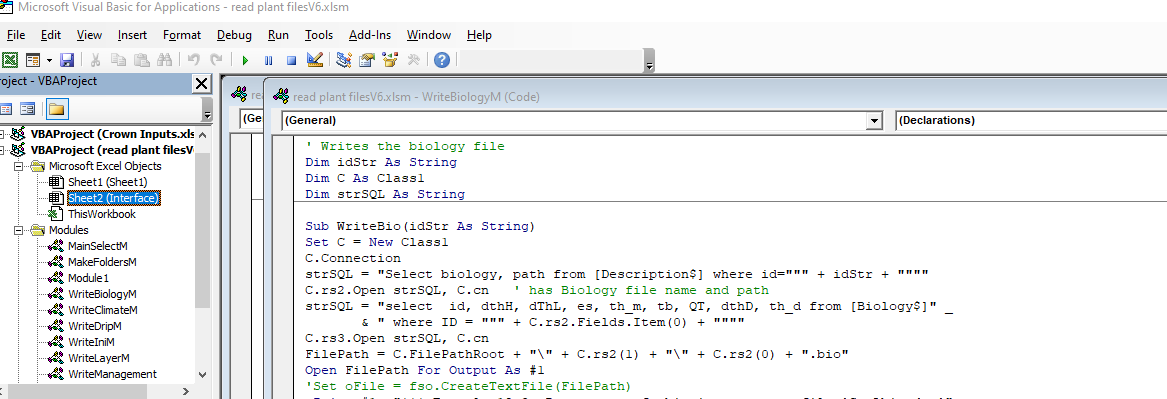
Right click anywhere in the ribbon/menu area and you should see a ‘customize ribbon’ option:

You’ll see this menu. Make sure ‘main tabs’ is selected on the right. Then place a check mark in the developer box if not there. (If it was checked you would have seen the ‘Developer’ Menu item. Click OK to exit.

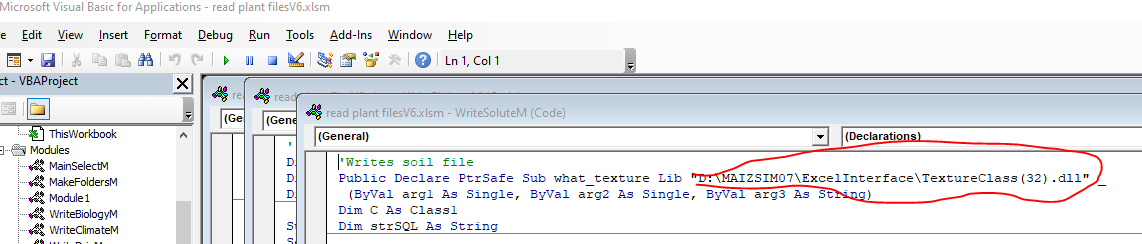


Now click on the ‘Developer’ menu item and you should see ‘Visual Basic’ as the first option. Click on it

This will bring up a new window that looks like this:



In the left navigation window under modules, look for ‘WriteSolute’ and double click on it. It should bring up the code window for this module:



In the circled area you will have to put your location of the textureclass(32).dll file. Make sure it is in quotes and there has to be a space before the “\_”. Put in the full path.

Click on the save button (blue disk icon)

This is a summary of the modules

|  |  |
| --- | --- |
| Module 1 | Imports a plant output file and inserts into spreadsheet – this one may not be up to date |
| WriteBiologyM | Writes the biology file |
| WriteIniM | Writes the initialization file |
| WriteSoluteM | 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 |
| WriteManagement | Writes management files |
| WriteTimeM | Writes the time file |
| WriteVarietyM | Writes Variety file. Note that it inserts some lines of data that are not in the database. This is for manually changing some of the advanced parameterr |
| WriteClimateM | Climate file |
| WriteNitM | Nitrogen file |
| WriteRunM | Writes the run file |
| MakeFoldersM | Creates subfolders |
| MainSelectM | Runs macros for selected id's - choose them from the excel page |
| WriteLayerM | Creates the layer file that is input to the CreateSoils program and writes the batch files needed to run the program |
| WriteDripM | Creates file for drip irrigation if used |
| WriteIrrigM | Creates file for irrigation other than drip |
| WriteWeaW | Creates the weather files using a CSV as input |

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 (see above) is called from the path where soil and grid files should be stored. The command line requires a fully formed path to 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. This makes the location of the layer file and location of the program independent of where it is run from. 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

Substitute 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 maizsim input files (output by the excel interface) can be specified here.

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

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

[Description]WeatherID [Weather]WeatherID

[Description]ClimateID [Climate]ClimateID and [Weather]ClimateID

[Description]SoilFIle [Soil]SoilName

[Description]Hybrid [Variety]Hybrid

[Description]Biology [Biology]BiologyID

[Description]Solute [Solute]ID

[Description]Tillage [Tillage]ID

[Description]

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

Description of variables (note that case should be consistent when using the variable in other sheets):

Description sheet

|  |  |
| --- | --- |
| Name | Description |
| ID | Main Identifyer – a key to relate to variables in other tables |
| WeatherID | Links to row in weather table with weather data, usually the site |
| Hybrid | Variety name |
| VarietyFIle | Name of variety file |
| SOilName | Name of soil |
| SOilFile | Name of the soil file |
| weatherfilename | Name that will be given to the weather file (can be the same for all sims) |
| climatefile | Name given to climate file |
| climateID | Row in climate file with data – one site can have different climateID’s if the climate data are different. |
| Location | Same as site – a name for a location with a lat – lon value |
| NitrogenFIle | Name of the file with the nitrogen solute data |
| Solute | Name of the file with the solute data |
| path | The path under the main folder where the scenario data will be written |
| Biology | ID of the source line in the biology table (sheet) |
| longName | Optional – used by external progams |

|  |  |  |
| --- | --- | --- |
| Name | Description | Commonly used |
| ClimateID | ID to relate to Description table | Y |
| site | Name or code of the site | Y |
| latitude | Latitute (degree, minute) | Y |
| longitude | Longitude (degree, minute) | Y |
| DailyBulb | 1 if daily or hourly wet bulb temps are available | N |
| DailyWInd | 1 if daily or hourly wind values are available | Y |
| RainIntensity | 1 if daily rain intensity values are available. Only used for daily weather data | N |
| DailyConc | 1 if daily rainfall Nitrogen is available | N |
| Furrow | 1 for furrow irrigation | N |
| RelHumid | 1 if Daily or hourly RH values available |  |
| DailyCO2 | 1 if daily or hourly CO2 values are available |  |
| Bsolar | Conversion of solar units converts input value to J/m2 | N |
| Btemp | Conversion of temperature converts to Celsius | N |
| Atemp | Conversion 2 of temperature converts to Celsius | N |
| Erain | Conversion of rain values to cm (if rain is input as mm then Erain is 0.1) | N |
| BWInd | Conversion factor for wind (usually 1) converts input value to km/hr | N |
| BIR | Conversion factor for verage rainfall intensity rate (cm/hr) – keep as 1 | N |
| AvgWind | Average wind (Km/hr) | Y |
| AvgRainRate | Average rainfall rate (used for daily data) cm/hr | Y |
| ChemCOnc | Nitrogen concentration in rainfall (usually zero) (mg/L) | N |
| RH | Average RH when daily/hourly values not available (%) | Y |
| AvgCO2 | Average CO2 when daily/average not available | Y |

Structure of Excel database

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

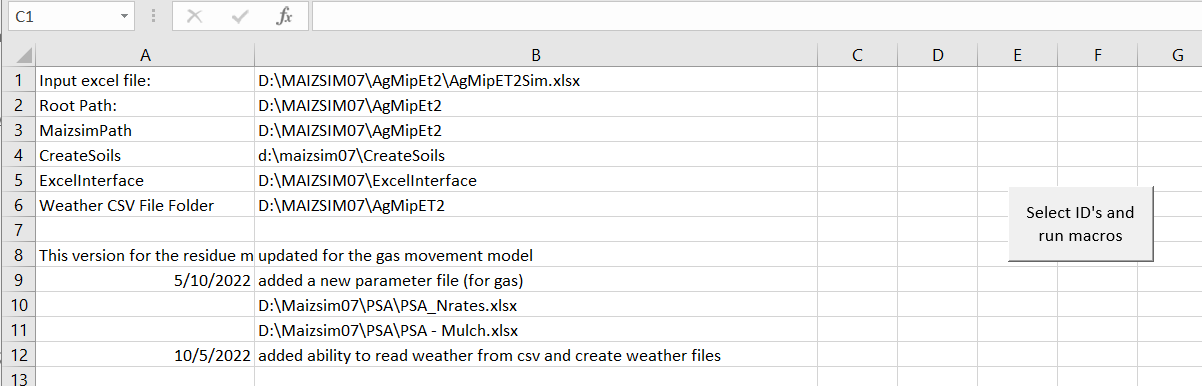
Root path is the path below which the paths for the simulations exist. Each simulation has its own path

The maizsim path is the path where the 2DMAIZSIM.exe and MAIZSIM.dll files exist, usually the same as the root path

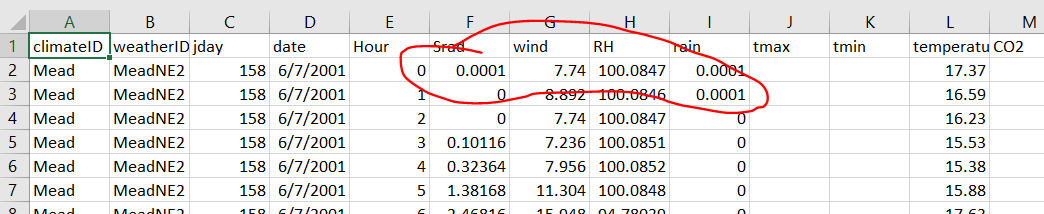
Create soils and Excel interface are the locations of the CreateSoilFiles.exe and the excel interface file (read plant filesV6.xlsm or read plant FilesV8.xlsm). the V8 version works with the mulch version of 2DMAIZSIM. The input excel file works with both versions of the macros.

The Weather CSV File Folder is the folder that contains the CSV file with weather data. The format of this file is below. The same format is used for daily and hourly data.

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.



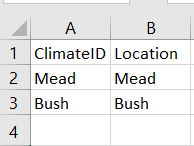
**Weather data format:**



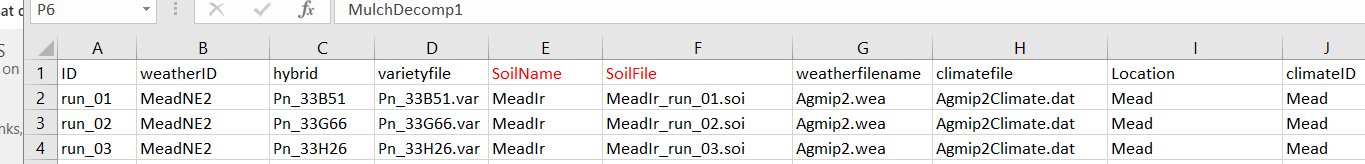
All the headings must be in the weather file no matter what data are available. wind, RH and CO2 can be blank for daily or hourly data. for hourly data, hour and temperature are required. the CSV extension is also required. The file name is entered into the filename column in the Weather sheet. Not that the first two values of rain and first value of Srad are 0.0001, small floats. This is to prevent the SQL engine from inferring these columns to be integers.

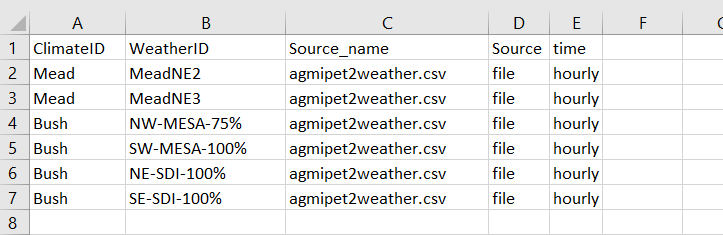
for Daily, tmax and tmin are required. The Column for WeatherID is used if there are different values of temperature and or rainfall for any one site.

in the climate sheet you have

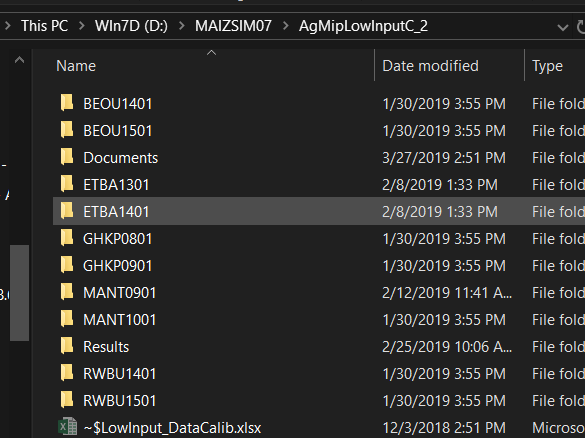


This ClimateID is the first column in the weather file. There is a sheet called ‘Weather’ in the excel file, this is where you specify the WeatherID. The weather and climate ID’s are also specified in the ‘Description’ sheet. In this case the Mead site has two weather ID’s. for this case they have different date ranges.





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.

