Table of Contents

[Database Design 3](#_Toc393790938)

[Microsoft Access Development 3](#_Toc393790939)

[Crops Database 3](#_Toc393790940)

[Crops Database Design & Functionality 3](#_Toc393790941)

[Weather Database 4](#_Toc393790942)

[Weather Database Design & Functionality 4](#_Toc393790943)

[Java Development 5](#_Toc393790944)

[Program Design 5](#_Toc393790945)

[Database Connection from Java 5](#_Toc393790946)

[Support 6](#_Toc393790947)

[Crop Type or Crop Variety is not found 6](#_Toc393790948)

[Country, Region, or Weather File not found 6](#_Toc393790949)

[Simulation Run 8](#_Toc393790950)

[Simulation Run Output 8](#_Toc393790951)

[Error Code Definitions 9](#_Toc393790952)

[Main Program Error 9](#_Toc393790953)

[Input Tab 9](#_Toc393790954)

[Run Tab 9](#_Toc393790955)

[Non-Tabbed Errors 9](#_Toc393790956)

[Add Crop or Variety 10](#_Toc393790957)

[Add Crop Type Tab 10](#_Toc393790958)

[Add Crop Variety Tab 10](#_Toc393790959)

[Error Code Locations by Section 11](#_Toc393790960)

[Main Program 11](#_Toc393790961)

[CropAndLocationDropDowns() 11](#_Toc393790962)

[CropType: Action Performed 11](#_Toc393790963)

[Country: Action Performed 11](#_Toc393790964)

[Region: Property Change 11](#_Toc393790965)

[Weather File: Action Performed 11](#_Toc393790966)

[Add Button: Input Tab 11](#_Toc393790967)

[Fix: Run Tab 11](#_Toc393790968)

[Run: Run tab 11](#_Toc393790969)

[Add Crop or Variety 11](#_Toc393790970)

[Add Crop Type Tab 11](#_Toc393790971)

[Add Crop Variety Tab 11](#_Toc393790972)

# Database Design

## Microsoft Access Development

### Crops Database

The crops database is to hold specific information about each individual crop. Although this database could have been built using the relational properties of Microsoft Access, it was built being a relational database by the means of variables. The \_CropTypes table will hold the generic name of a specific crop type *(ex: Corn, well there are many types of Corn, so the generic is just Corn)*. Each generic crop is related to a specific table, for the type of generic crop. Again let’s take Corn as our generic type, it’s related to the CornTypes table which will allows users to retrieve specific data about different types of corn, edit existing information about a specific corn variety, or add a new corn variety to the system.

I chose to use Microsoft Access for a database for many reasons.

* The database could easily be looked at with a graphical interface with anyone that has Microsoft Access, which is much cheaper than Buying Oracle or MS SQL. MySQL is an option, however, at this time there is no known good GUI for it. Moreover, it is much more likely that people will have Microsoft Access and not MySQL, thus makes a favorable option for farmers, scientists, etc.
* The Database was designed to not be a relational database for multiple reasons. First, the complexity of the Query Strings could have gotten a bit out of hand. Also if the scientists wanted to make modifications in the future it would have been harder for them. Moreover, the information that needed to be stored, some of the information could have been setup to be a relational database, but the other information couldn’t have been. Such as, we wanted all crops to have a specific table for all of their varieties. By dividing up the data to specific tables it takes the place of using a relationship between the tables. This will make it easier to parse through information, retrieve information, and store information. It also make it simpler to understand for farmers, scientists, etc.
* Due to the limitation of the connection between the Java program and the database, it doesn’t allow me to return database schemas. Therefore I am unable to retrieve the column names for any table. This limiting factor forced me to create an additional table that will store the column names which will then be used to populate the labels in the program.

### Crops Database Design & Functionality

The first table was designed for drop down menu 1 in the “Crop Type and Location” section. After populating that drop down, the user will have those choices, the choice will be compared to what was brought back and that will tell the system what table to go look at in the crops database to populate the 4th drop down menu for the Growth Stages, loading the very first entry into the program as a default.

### Weather Database

The weather database will contain 3 tables before reaching destination. Some people may only have to go through 2 tables. The first table is the Country table. This asks what country do they live in? Depending on the country and if they have regions *(ie: United States has States)*. If the country doesn’t have regions then it only contains 2 tables, because there is the country and then their data. If the country is divided up into regions, then there is a region database that will help separate the data between regions.

The first table will hold all of the countries in the world. The second table will be named after the   
Alpha-3 code from the ISO 3166-1 encoding list *(*[*http://en.wikipedia.org/wiki/ISO\_3166-1*](http://en.wikipedia.org/wiki/ISO_3166-1)*)*. If there is a third table, that table is name of the region that the crop belongs to. The Weather

### Weather Database Design & Functionality

Much like the cro

# Java Development

## Program Design

PhenologyMMS 2.0 was designed using Java. I started a new design that would make the program easier to use and retrieve data very quickly. Java Swing was used along with Netbeans to create a quick user interface. The user interface starts with the first tab, which is input data. The input data allows the user to choose the type and variety of crop. Once the type of crop is chosen, the planting information is then grabbed from the database and it populates the Planting Information section of the Inputs tab. Once the variety has been chosen then it will read the Variety Growth Stage information from the appropriate Crop Type table. I weather file must be selected so that the run can take place. The weather file is selected by choosing the appropriate Country, and if that country has regions or states then that must be selected as well. If the Country does not have states or regions then the weather file should be found in their country, on the other hand if they have regions then the weather file is found in the region where that file should be. The entire purpose of this newer design is not necessarily to create simplicity, but to keep everything organized and continue to keep data without getting rid of it. The old system was reading of text files that have lots of data in them of mixed types and varieties. So if you needed to modify a value, it was cumbersome, whereas now all data is in specific tables of other like data.

## Database Connection from Java

I chose to use UCANACCESS library that depends on 4 other libraries to operate. This method allowed me to connect to Microsoft Access with the functionality that I needed. However, during the implementation and research, none of the support documents mentioned that it wasn’t capable of getting database schematic information. Thus, we retrieve label information from the database differently.

## Support

### Crop Type or Crop Variety is not found

It is likely that there is not a crop type or crop variety that is not yet in the system. We only put specific crops in the system that we already had data for. If you have all the write data, adding a crop is simple. Under the menu bar at the top of the program click on “Tools”, and then click on “Add Crop or Variety”. When this window opens, you will see 2 tabs. One is for adding a crop type and the other is for adding a crop variety. For instance we do not have “Carrots” in the system. For the purpose of this program, it’s appropriate to stick a carrot variety under Carrots, which means a carrot type would have to be created.

#### Creating a Crop Type

[cont…] Sure you could create a carrot variety and stick it under corn, beans, or wheat, but that would just silly. When creating a new crop ALL “Plant Information” fields must be filled out. After the Plant Information is finished, you MUST have a minimum of 5 growth stages, which you will enter the growth stage names in order of occurrence. If you are unsure of growth stage names, just click the automate button and we will put in our basic 5 growth stages that should be available for almost all crops. As with all things, since we have a minimum of 5 Growth Stages, there is Maximum of 20.

#### Creating a Crop Variety

[cont…] Now that you have a crop type to hold your crop variety, you need to choose your Crop Type, which in this example would be “Carrots”, when you did this you should have noticed that some rows in the first column were populated; this is the crop Growth Stage information. At the very least you should know the GDD *(Growing Degree Days)* for each of the growth stages listed for both non-stressed and stressed conditions. As you put in the GDD values, the values of NoL *(Number of Leaves)* will be automatically populated.

### Country, Region, or Weather File not found

If in the event a weather file is not found, then a weather file must be added. At the top of the screen under “Tools” there are several options to choose from. One of them is to “Add Weather File”, this will open a new screen that will allow you to add a weather file, a country or a region. Let’s say we want to add a new weather file for China. We did not setup China in the original program, so on this screen before we can add the weather file, we must add China first. Before doing that, we should discuss organization real fast because it becomes important. If you are doing lots of research in a large area or country, image if I had 50 weather files all under the country China. That means every time I want to find a weather file I have to search through a worst possible scenario of n weather files and on average o n/2, which n = 50 in this case. If you can divide the weather files into regions that is highly recommended. Now let’s look at China again and let’s say I am doing work in East China, North China, and Northeast China. Let’s say that by putting in 1 minute of extra work I created regions for the 3 previously mentioned regions, and now let’s say there are 15 weather files for East China, 25 weather files for North China, and finally there are 10 weather files for the Northeast Region. Now my worst case situation is n=15, 25, 10 and an average would be n/2. Taking a few extra seconds to keep put in regions is worth your time.

#### Creating a new Country

[cont…] Sure you could add your weather file under another country, but that too would seem silly because the entire idea of this program is to keep things organized and somewhat efficient so that more time can be done doing research. On the screen you will see select a country and you won’t see China, so in the text box just to the right, please type in China and click on the Add button to the right of that. This will now add your country and your drop down button for “Select Country:” will now show “China”. Click on China, now depending on how accurate you want to be you could stop here and add a weather file. At this point all of your weather files that you bring in would be in:

(your drive)/PhenologyMMS 2.0/Interface/MMSWeather/China/

However, since China is very large, like Russia and the U.S.A, it might be more appropriate to create regions. Now I am using regions based on <http://en.wikipedia.org/wiki/List_of_regions_of_China>, the regions could also be the provinces of China, which by my count they have 33. So we will use the previous URL and say we are in the South Central China. We should probable create our Region.

#### Creating a new Region

[cont…] Click on the drop down menu for “Select Country” and now choose “China”. You still won’t find any region information under the “Select Region (State)” drop down menu, so it’s necessary to type in your region in the text box on the right of “Add A Region”, In our case we will type in “South Central” and then click on the Add button to the right of that text box. Now if you go back to the drop down menu’s for “Select Country” and choose “China” and then for “Select Region (State)” choose “South Central”, now you can click on “Add Weather File”. Now you will find your weather files in this location:

(your drive)/PhenologyMMS 2.0/Interface/MMSWeather/China/South Central/

Fun, Right?

***NOTE:*** This entire program is database driven. If it is not in the database then it doesn’t exist. For instance. You take the following path: (your drive)/PhenologyMMS 2.0/Interface/MMSWeather/China/South Central/ and you decided that you wanted to just put a file in there and read it. This won’t work. Every weather file in the database is associated with 3 or 4 components: 1. Country, 2. Region, 3. Latitude, and lastly 4. name of the weather file. Therefore, by placing a file in to that folder it will not automatically let you see it. Also, if you change the name of the weather file or delete it, the system can have a varied response.

## Simulation Run

Simulation runs take all the data that was given and put it into a file for a Fortran code to read. The Fortran code is designed only for a few specific crops. However, you can reach us here at the USDA and we can add in addition Fortran code for your specific crop.

### Simulation Run Output

An output file is created for the Fortran code to read, this output file contains the following summarized data:

1. Crop Name
2. Crop Variety
3. Location / Weather
4. Planting Day
5. Planting Year
6. DOY of Planting Date
7. Planting Depth (in)
8. Rate (#/m2)
9. Rows 9-30 for Emergence Data
10. Latitude
11. GDD Method
12. Base Temperature
13. Lower Optimum Temperature
14. Upper Optimum Temperature
15. Maximum Temperature
16. GDD / Leaf
17. Maximum Canopy Height
18. Rows 39-44 used for Vernalization Data
19. Canopy Height
20. Rows 46 – EOF are used for Growth Stage Selections

# Error Code Definitions

Below are the definitions of error codes as of: 7/22/2014. These error codes are used all over the program, however, where the first type of the below function occurs is its home. So you may find 100, 200, and 900 level error codes outside of the main program.

## Main Program Error

### Input Tab

100 Level Errors: Trouble trying to access the Crops Database

(100) : Unable to Read Row from crops database

(101) : Unable to retrieve Column of Data from crops database  
 (102) : Unable to retrieve Column Headers from crops database

200 Level Errors: Trouble trying to access the Weather Database  
 (200) : Unable to retrieve Table from Weather Database

(201) : Unable to retrieve Column of Data from Weather Database  
 (202) : Unable to read a Row from Weather Database

300 Level Errors: Populating Plant Information

(300) : Unable to Populate Planting Information

400 Level Errors: Add Button Checking Status of Input Tab

(400) : Filename Error *(no filename provided or the filename is not unique)*

### Run Tab

500 Level Errors: Fix Crop Button on Run tab

(500) : Reloading a Crop Error *(no crop selected or more than 1 crop selected)*

(501) : Improperly Formatted Date

600 Level Errors: Run Batch Action Performed

(600) : Writing Output File failed

### Visualize Tab

### Non-Tabbed Errors

700 Level Errors: Crops Database code

(700) : Unable to connect to Database

(701) : SQL String could not be executed

800 Level Errors: Weather Database code

900 Level Errors: Miscellaneous Errors

(900) : Null Pointer Exceptions

## Add Crop or Variety Tab

### Add Crop Type Tab

1000 Level Errors: Add Button for Crop Type  
 (1000) :Data was not added to database “\_CropTypes”  
 (1001): Unable to Create Table in Crop Database

(4000, .00 - .11): Date Format Problem

(4001): Invalid Crop Name

(4002): Invalid Planting Depth

(4003): Invalid Plant Rate

(4004): Invalid Max Canopy Ht

(4005): Invalid Base Temperature

(4006): Invalid Lower Optimum Temperature

(4007): Invalid Upper Optimum Temperature

(4008): Invalid Max Temperature

(4009): Invalid Gdd Method

(4010): Invalid GDD / Leaf

(4011): Improper Growth Stage Fields, either not enough or there is enough just not side by side

### Add Crop Variety Tab

(4012): Growth Stage Fields are missing values

(4013): Vernalization Fields are missing values

(4014): Additional Data Fields are missing values

(4015): Invalid Crop Variety name

## Add Weather File, Country, Region Tab

Add Weather File tab

2000 Level Errors: Add Button for Weather File tab

Add Country tab

Add Region tab

# Error Code Locations by Section

## Main Program

CropAndLocationDropDowns()  
 100.1, 100.2, 101.1, 101.2, 102.1, 200.1, 201.1, 300.1, 900.1 , 900.2

### CropType: Action Performed

100.1, 100.2, 101.2, 102.1, 300.1  
  
CropVariety: Item State Changed

100.3

Country: Action Performed  
 200.1, 201.2, 201.3, 900.3

### Region: Property Change

200.2, 201.4

### Weather File: Action Performed

202.1

### Add Button: Input Tab

400, 401, 402, 403, 404.1, 404.2

Fix: Run Tab  
 500.1, 500.2, 501

### Run: Run tab

600

## Add Crop or Variety

### Add Crop Type Tab

100.4, 101.3, 102.2, 900.4, 1000.1, 1001.1, 4000-4011

### Add Crop Variety Tab

100.5, 100.6, 102.3, 1000.2, 4012-4015

## Add Weather File, Country, Region