**GUIDE TO THE “VSVS MACHINE”**

By Michael Herschbach

Test Version 3, 9/22/2025

---------------------------------------------------------------------------

**Table of contents:**

1. **Python**
2. **Folder and files**
3. **Startup**
4. **Importing (from sorter and datafile)**
5. **Reports**
6. **Email wizard**
7. **Troubleshooting**
8. **Wish/issue list**
9. **Importing from Access**

**PYTHON:**

At this point in time the code has not yet been compiled into a single executable. Unfortunately, this means that you will have to have Python installed onto your computer with all of the required dependencies (packages).

There are many ways to install/run Python. If you are familiar with Python then you likely know how to install packages and run the code. If not, follow these steps:

1. Install the Spyder IDE from <https://www.spyder-ide.org/>
2. Open the “appShell.py” document in Spyder
3. In the console (see diagram below), type each of these lines to install packages:
   1. “!pip install selenium”
   2. “!pip install webdriver\_manager”
   3. “!pip install reportlab”
   4. “!pip install keyboard”

*Spyder should look something like this (don’t worry about specifics, just the console area and “run” button:*

A screenshot of a computer

AI-generated content may be incorrect.

**FOLDER AND FILES:**

The folder with the VSVS Machine should look something like this:

A screenshot of a computer

AI-generated content may be incorrect.

The datafile is what stores all of the info (teams, teachers, volunteers)

* You can choose the datafile yourself. What is important is that it is in the expected format (which will be generated upon import from the sorter) and that it is a **CSV**

The green highlighted files are the outputs of the program (sticker labels, tables, etc)

**appShell.py is the main VSVS Machine python file**

All other “.py” documents are essential to the program but you will not have to directly interface with them.

Note: other “.csv” files and documents pictured in the above screenshot are not necessary and were used for testing.

**Format of main file:**

Group Number, First Name, Last Name, Email, Phone Number, Team Leader, Board Member, Teacher, Day, Start Time, End Time, Teacher Email, Teacher Phone, School, Grade Level, Lesson 1, Lesson 2, Lesson 3, Lesson 4, Lesson 5, Lesson 6

A screenshot of a computer

AI-generated content may be incorrect.

**STARTUP:**

Just open appShell.py in your Python environment and run it.

It should automatically open the application window:

*Mac:*

A screenshot of a computer

AI-generated content may be incorrect.

*Windows:*

*A screenshot of a computer

AI-generated content may be incorrect.*

**IMPORT:**

**Import from sorter**:

Click the “Import from Sorter” button and select the “assignments.csv” file produced by the sorter.

Test data from S25 is in the folder as “25SorterExampleData.csv”.

This will create a new csv in the “MAIN” format, which is accepted by the program for future tasks.

* Will be saved in “MAIN\_SHEEET.csv” in folder
* **Many columns will be empty** since the sorter outputs limited information
* **You will have to manually copy this information** 
  + Teacher email, school, lessons, etc

**Import from datafile:**

Every time you open the application you must reimport the datafile.

Simply click “Import from MAIN” button and select the .csv file that you are keeping the datafile in

You should get some feedback in the console:

A screenshot of a computer

AI-generated content may be incorrect.

Now you are ready to generate reports or write emails!

**REPORTS:**

Just click the “Generate Reports” button (after importing!).

All reports (labels, tables, etc) will now appear in the folder that the application is in.

You will receive feedback in the console:

A screenshot of a computer

AI-generated content may be incorrect.

All reports are in pdf format.

**See next pages for report formats…**

**Sticker labels:**

Week 1 labels: label\_w1.pdf

Week 2 labels: label\_w2.pdf (and so on…)

A white paper with black text

AI-generated content may be incorrect.

(prints 2 columns to a page, Avery 5162 format)

**Table of teams:** teams\_table.pdf



**Full table of teams (has each group member):** teams\_table\_full.pdf

A table of blood pressure

AI-generated content may be incorrect.

**Checkout sheets:**

Week 1: checkout\_week1.pdf

Week 2: checkout\_week2.pdf (and so on…)

**A table with text and words

AI-generated content may be incorrect.**

*Note on checkout sheets:*

For the additional information column, the program compares the first letter of the lesson name to the lessons in the CSV file “quick\_check.csv” which should be in the folder. This populates the “additional information” column of the checkout sheet with the information in the “pick-up additions” column from “quick\_check.csv”

* **Double check to make sure lessons all start with a unique word or this will not work**

**EMAIL WIZARD:**

As soon as you click the “Email Wizard” button, a chrome window will open that is being controlled by Python. Do not touch anything!

A screenshot of a computer

AI-generated content may be incorrect.

First the program will open the Outlook home page and **automatically** will click the sign-in button for you. This opens a new tab with the sign-in.

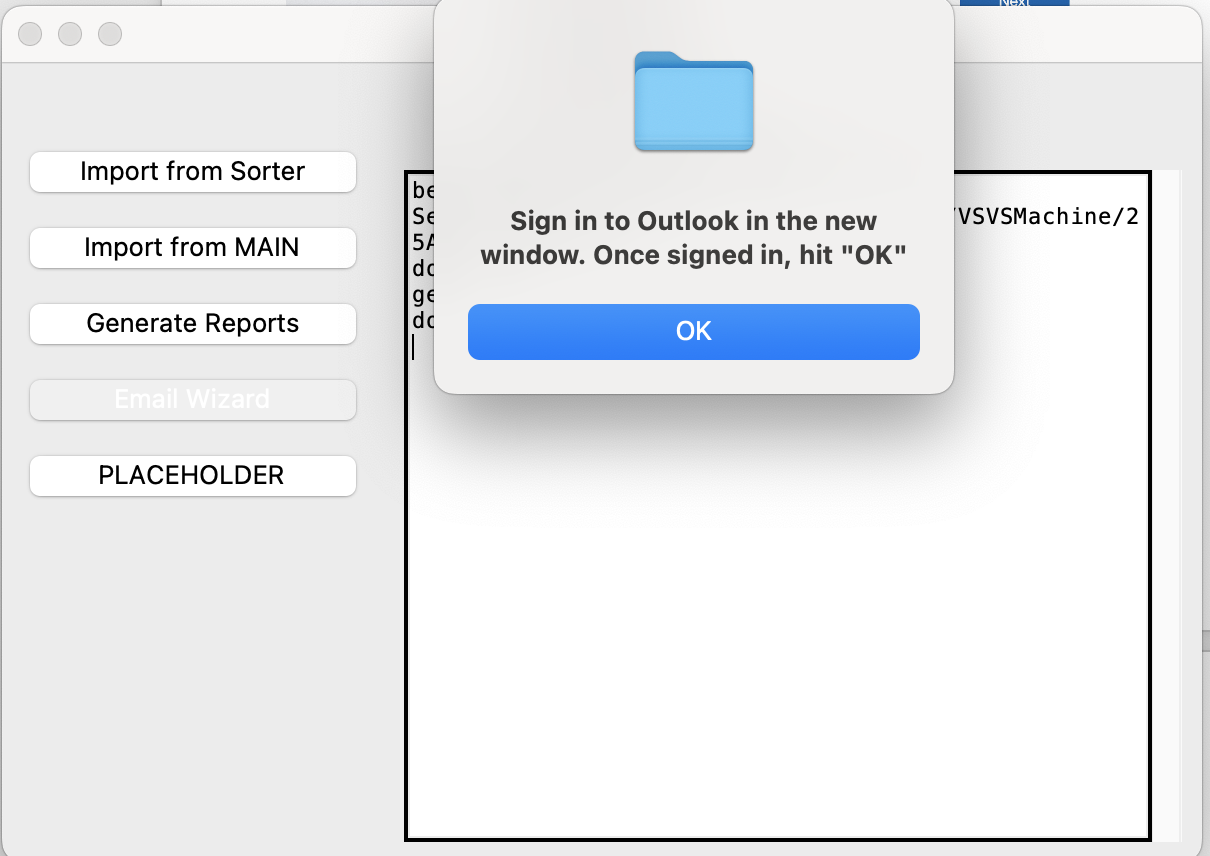
Now, enter your login details and Outlook will open. Do not close the window or tabs!

A screenshot of a computer

AI-generated content may be incorrect.

Return to the program. You should see a pop-up:

*Mac:*  *Windows:*

 A screenshot of a computer

AI-generated content may be incorrect.

Click the OK button if you are signed in. This will open a new application window with the wizard:

A screenshot of a computer

AI-generated content may be incorrect.

Here, you can put in the team numbers you want to email, as well as the subject and email.

For emailing multiple groups, separate the numbers by commas (ex: “1, 3, 17”)

Once you click send the program will go back to the outlook window and write the email. A pop-up box will appear. Check that the email looks correct, and then **manually send the email**. Click the pop-up box once the email is sent and the program will write the next email (if multiple groups were inputted, or the All Teams option was checked).

**Message placeholders:**

For the subject line and message, you can use variable placeholders. Simply put the bracketed placeholder and the program will replace it with the proper value.

Example: Hello team {team\_number} 🡪 Hello team 3

**{team\_number}** – team number (just the number!!)

**{matrix}** – team information:

A close-up of a phone number

AI-generated content may be incorrect.

**{tblock}** – teacher information:

Grade level: 7

A black text on a white background

AI-generated content may be incorrect.

**Subject line** supports **ONLY the group # keyword:** {team\_number}

**Example of sending email using wizard:**

A screenshot of a computer

AI-generated content may be incorrect.

Produces:

A screenshot of a email

AI-generated content may be incorrect.

**Options for email wizard:**

**Include teacher:** adds teacher’s email to list of recepients

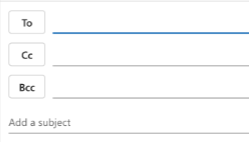
**Email all teams:** Emails all teams in spreadsheet EXCEPT teams with a group number of “0”. Overrides input in group # line.

**CC/BCC skip:**

By default this should be turned on.

Depending on the configuration of your Outlook page, you may have no CC/BCC section, or you may have one for each, see below:

*CC/BCC section:*



*No CC/BCC section:*

A screenshot of a email

AI-generated content may be incorrect.

If you have **NO** CC/BCC section, you should turn the CC/BCC skip **OFF**

If you have a CC/BCC section, you can keep the checkbox checked and the program will know to skip over these lines.

**Mac checkbox:** ignore for now, was meant for auto-send

**TROUBLESHOOTING:**

Sometimes the program gets frozen. To fix this, click the top bar and drag the program. For some reason this fixes it.

Once you run the program once, it is good to restart the Python kernel. You can do this in Spyder by right-clicking the console and hitting “quit”

**Importing from an Access database:**

To import from an active VSVS Access database you will need two reports:

* Teacher Information
* Member Information Tabular

Create a new CSV spreadsheet with the following headers in the first row:

Group Number, First Name, Last Name, Email, Phone Number, Team Leader, Board Member, Teacher, Day, Start Time, End Time, Teacher Email, Teacher Phone, School, Lesson 1, Lesson 2, Lesson 3, Lesson 4, Lesson 5, Lesson 6

Then, simply copy and paste the appropriate columns from the Access reports into your new spreadsheet.

You will notice you have more individuals than classrooms. Unfortunately, you will have to now move all of the teacher information so it is with the corresponding groups (but you only need it on the first line of each group, see below):

*After first copying:*

A screenshot of a calendar

AI-generated content may be incorrect.

*After moving teacher/classroom information to proper lines:*

A screenshot of a spreadsheet

AI-generated content may be incorrect.

Save as a CSV and you should be good to go!