**Macro-Enabled M-Line Drive Selection**

**Debugging and Troubleshooting**

I have included somewhat comprehensive error-checking and error-messaging within the main macros designed to scrape the Excel sheets for the necessary information. In other words, if a cell in the created VFD parameter sheet has an inaccurate value, there probably was already an error message alerting you to this. If this error keeps happening, and/or you can’t figure out what the error is, and/or the error message is unhelpful, see if the below tips/expectations are helpful. Otherwise, email [bmosher@cambridgeair.com](mailto:bmosher@cambridgeair.com) (during the summer) and see if I know what might be happening. If at any point you’d like to use the VBA macro editor, seethe bottom of this sheet.

**Common Errors:**

* **“There is nothing in the RPM field. Drive Selection unsuccessful:** More likely than not, Excel got “stressed out” by doing too many fan calculations. It will then refuse to print numbers yielded from the fan calculation into the RPM field within the Drive Sheet. You will notice, under your Excel tab, that one of the intermediate files was opened in the process before it was successfully closed out. If you open this file, **Temporary.xlsm**, the problematic cell will be highlighted. You should notice that this cell is blank, even though it has a formula embedded in it. To fix this, close Excel entirely, including all files (don’t save **Temporary.xlsm** if ever prompted) and then attempt to run the macros again. This method fixes the glitch about 99% of the time.
* **Nothing looks wrong, but no RPM, etc. calculation:** This seems like a bug, and one that almost never occurs since the addition of the drive selection macros. What you’ll see on the sheet is likely a small green triangle and nothing else on the inside. This isn’t actually an error! Excel just miscommunicated with itself in allowing macros. The easiest fix is to close the sheet and reopen it. This time you should see a yellow bar near the top of the sheet asking to enable macros. Once you enable, the values instantly populate into the sheet.
* **Error 400/File Read Error:** Most often, a sheet will have to be manually closed out if you’re getting this error. The macros include comprehensive file checking to determine if a file that is trying to be opened is already open. This would cause an issue. It catches most, but sometimes, sheets fall through the cracks. This usually happens because of a yellow warning in the ribbon above the sheet. The warning seems to not quite allow the workbook into Excel’s “list” of open workbooks, therefore the file is believed to be closed, strangely. Files that, if open in Excel, may need to be manually closed: **VFD\_Template.xlsm**, **Motor\_BOM\_Tool.xlsx**, **M\_Drives.xlsm**
* **#VALUE Error:** This has more to do with the **Lau** and **Comefri** modules built into the Excel VFD parameter sheet and their mysterious DLL’s. Contacting the creator of these could certainly help diagnose some errors, but otherwise you should go into the **Setup and Repair** folder and look at its information to see if your software is not yet compatible

**Macro Expectations and Error Messages:**

 The collection of functions within the macro generally expect to find the necessary information in the same place each time. So, if the SMART paperwork format changes even a little bit from what the macro thinks it should be, there could be unexpected results. This won’t “break” the creation of a new sheet, it just might be missing a few things. The sheet will still be successfully created if the message that pops up has this symbol:

A sheet will not be created if the message contains this symbol:

The error message for these warning symbols is quite specific, so additional information would be redundant.

For all messages with a symbol, here’s what the macro expects for each message type:

* **08-Numbered Motor:** There will be a text substring within the order breakdown table that is formatted as “08-###” where # indicates any number of numerical (integer) values ended by a space. It should have the Motor’s name in an adjacent cell. The Motor Name should contain either “ODP” or “TEFC”
* **Airflow:** There will be a text substring within the large cell in Excel that is formatted as “AIRFLOW: ##,### CFM”, where # indicates a numerical (integer) value
* **Altitude:** There will be a text substring within the large cell in Excel that is formatted as “ELEVATION: ###.## ft.”, where # indicates a numerical (integer) value, and . indicates an optional decimal based on whether the altitude is fractional or not
* **Blast Type:** There will be a text substring within the large cell in Excel that is formatted as “BLAST: XXXX Blast”, where X indicates either “Horizontal”, “Up”, or “Down”
* **HP:** There will be a text substring within the large cell in Excel that is formatted as “MOTOR: ##.## HP”, where # indicates a numerical (integer) value, and . indicates an optional decimal based on whether the HP is fractional or not
* **HP-Phase-Voltage Motor Not Found:** The worksheet “Motor\_BOM\_Tool.xlsm” will contain the necessary specifications to locate FLA and RPM parameters for the requested motor. As per the error message, there either is no entry, or there was an issue with what was trying to be found. This function is almost entirely self-sufficient so something bigger is wrong if there is an unexplained error. It’s possible that the Monday board exported to Excel changed formats.
* **Gas Type:** There will be a text substring within the large cell in Excel that is formatted as “GAS SUPPLY: XXXX”, where X indicates either “Natural Gas” or “Propane”
* **Geo Manifold Pressure:** There will be a text substring within the large cell in Excel that is formatted as “MANIFOLD PRESSURE @ GEO: ##.## “WC”, where # indicates a numerical (integer) value, and . indicates an optional decimal based on whether it is fractional or not
* **Job Date:** There will be a text substring within the large cell in Excel that is formatted as “DATE: MM/DD/YY”. If not found in a Unit Worksheet, the cell “A1” is searched.
* **Job Name:** There will be a text substring within the large cell in Excel that is formatted as “JOB NAME: XXXX”, where X indicates the job name. This should be IMMEDIATELY followed by a few spaces and the word “Epicor”. Without this, the macro does not know when to stop reading the job name and would read the rest of the cell’s contents
* **Max Manifold Pressure:** There will be a text substring within the large cell in Excel that is formatted as “MANIFOLD PRESSURE @ MAX: ##.## “WC”, where # indicates a numerical (integer) value, and . indicates an optional decimal based on whether it is fractional or not
* **Model Type:** There will be a text substring within the large cell in Excel that is formatted as “MODEL: X###”, where X indicates a letter (M for most units, hopefully) and # indicates a numerical (integer) value
* **Mount Type:** There will be a text substring within the large cell in Excel that is formatted as “MOUNTING: X”, where X indicates the singly-lettered mounting type (V or H)
* **Phase:** There will be a text substring within the large cell in Excel that is formatted as “VOLTAGE: ###V/#PH”, where # indicates any number of numerical (integer) values. This and voltage use the same substring
* **Quantity:** There will be a text substring within the large cell in Excel that is formatted as “QUANTITY: #”, where # indicates any number of numerical (integer) values ended by a space
* **Shop Order:** There will be a text substring within the large cell in Excel that is formatted as “Epicor #: ######”, where # (after the colon) indicates a numerical (integer) value. If this is missing, the sheet will be renamed so as to “pin” it to the top of the Drive Sheet folder, and will have the name: “000 – ORDER NUMBER MISSING” to indicate the number needs to be manually enter. A procedure for this can be found in **Manual Entry Procedure.docx**
* **SMART Unit Worksheets:** The macro looks for this text substring: “UNIT WEIGHT:” and does so because only SMART Unit worksheets will have these, and allows the macro to create separate sheets for each sub-order, numbered 01, 02, 03, and so on. If the SMART Unit Worksheet no longer includes “UNIT WEIGHT:”, this could fail.
* **Static Pressure:** There will be a text substring within the large cell in Excel that is formatted as “STATIC PRESSURE: ##.## “WC ”, where # indicates any number of numerical (integer) values separated by an optional decimal
* **VFD Reference:** There will be a text substring within the large cell in Excel (beneath the order breakdown table) that is formatted as one of the standard ACH580 VFD Reference types (see ACH580 for all types)
* **VI Type:** There will be a text substring within the large cell in Excel that is formatted as “VI: XXXX”, where X indicates the type of VI (External, Internal, or No)
* **Voltage:** There will be a text substring within the large cell in Excel that is formatted as “VOLTAGE: ###V/#PH”, where # indicates any number of numerical (integer) values. This and phase use the same substring
* **Other – Order Locator:** There isn’t a good way to print an error message for this. This is almost similar to the SMART Unit Worksheets error, but this function is designed to find the location of the start of each separate order, not count how many there are. It does look for the same substring as the other error, though
* **Other – VFD Locator:** There really isn’t a good way to print an error message for this. Since an order is allowed to both have or not have a SMART VFD Worksheet, there’s virtually no way to catch errors in this. But what this function does is find the location of the start of each separate SMART VFD Worksheet, if it exists. It looks for this text substring: “MAXIMUM AIRFLOW:” and does so because only SMART VFD worksheets will have these, and allows the macro to tack this information onto the existing VFD Parameter Sheet. If the SMART VFD Worksheet no longer includes “MAXIMUM AIRFLOW:”, this could fail, and the information could go undetected

This covers all the macro expectations that they require in order to function. Again, if the formatting of the SMART sheet changed recently, you may be able to fix this by going into the VBA macro editor within the “**DriveSelect.xlsm**” sheet and change the necessary function so that it will search in accordance with the new change. Looking up resources on “**VBA Regular Expressions**” can help you understand the syntax.

**Other Unusual Errors:**

**Path Not Found:** This is likely because the intended folder for path redirecting was moved or destroyed. The code makes it easier on the user by trying to jump to the folder most relevant to the intended operation (jump to $(User)\Downloads for Drive Selection, jump to S:\M Series Drive Selection\Drive Selection Sheets for email templating). To fix this, you would have to go into the VBA macro editor within Excel and search for the above file paths to either delete them or change them to be the updated path. Since there will never be an instance in which a computer doesn’t have a Downloads folder, only the Drive Selection Sheets folder may be changed. The easiest way to avoid the hassle of messing with VBA is to add a folder in the intended location (as written above).

**Wrong Motor BOM:** Hopefully this will never be an error, but the DriveSelection function I built within the macro set searches a predefined area of up to 4000 rows. Searching by the used-up rows would take entirely too long and could cause program malfunctions. So, if the M-Drives sheet in Monday.com ever somehow grows to exceed about 3996 entries, the correct drive may not be found. The code within the macro would need to be changed to reflect this.

**This site can’t be reached:** If the URL looks fine, unfortunately the PDF converter website is down either temporarily or permanently. Another tool may need to be used, such as Adobe Acrobat Pro, to convert the PDF to an Excel sheet manually. If this is the case, it’s most intuitive to use the old version of the macros, called **Drive Select – no PDF converter**, in the **Backup** folder. This will do the same Drive Selection as before, just without the website redirection.

**Unknown Error:** If the error message is either blank, has only a number, or does not look like the two symbols above (blue question mark or yellow triangle), Excel hit an unusual error that I was not able to catch. If you’re really having trouble figuring out what is the matter with the process (and there can be a lot of things that go bad along the way), you can try stepping through the individual macros until you hit such an error message. You can accomplish this very simply by inserting the following code into the VBA macro editor within “**DriveSelect.xlsm**”:

Sub TestMacros()

Call CreateVFDSheets()

End Sub

If, for some reason, the name of the main module was changed from **CreateVFDSheets** to something else, all you must do is change the name. This allows you to step through the code until you hit a problem. The controls are easy:

* Begin by placing your cursor on the indented line within the provided code.
* Press F8 to step into each line (this will step into functions within functions, so use this ONLY when you want to see the function being called), OR
* Press shift+F8 to step over the current line (this will NOT step into any function calls, which may speed up debugging).
* If at any time you need to start over, just hit the “stop” square to quit debugging, and go back to the **TestMacros()** subroutine to start testing over again.

If you go this route, you *will* find the error, even if it isn’t explicitly killing the program.

**The VBA Macro Editor (aka Visual Basic, within Excel):**

If nothing else has worked so far, and the only remaining possibility to fix the errors is to look at the VBA code, you can access it by opening any Excel sheet, and then navigating to **File >> Options >> Customize Ribbon**. From here, the right-hand pane has an un-checked box for **Developer**. Check this box, and move it wherever you like on the tab bar by using the arrow buttons on-screen. This will create a brand-new tab with the **Visual Basic** editor as the left-most button. Once clicking on it, almost the entirety of the code can be found in **ThisWorkbook** (roughly 2000 lines). Even if code is called from outside the scope of the workbook, stepping through the code will allow you to see each individual step as it happens. Some sections in this document may help with understanding the lines of code in the workbook.

If you go this route, make sure to duplicate and save a backup copy of the **DriveSelect.xlsm** workbook and place it in the **Backup** folder within **VFD\_Parse**. **If you overwrite the code and make it unusable without creating a backup, the original functionality will be lost forever.** Same goes for **VFD\_Template.xlsm**, and/or any other sheets in the process. Be verycareful when you try this.