https://github.com/QuinnB-tm/TM_ObsoletePartsBot

Documentation for the Obsolete Parts Bot

This report includes both a basic summary of how to use the program as well as a step by step description.

Summary

The program should come in the form of an application that prompts the user to select needed files upon opening it. The necessary files are:

- All parts
    - Contains: every part in the database
    - Export form: Update Data
    - Export field: External ID, Internal Reference, Name, Product type, TWM active, Products/External ID, Product Category
- Star Lord BoM
    - Contains: every active(in use, not Odoo's active) part and assembly
        - Probably best to get the active assemblies from configuration spreadsheets and quotation templates, then load those into a BOM and then export it as a flat BOM

The other, *optional*, files are:

- Excluded Parts
    - Contains: All the parts that should be excluded and weren't in the Star Lord BOM
    - The program only reads the furthest left column
    - Probably worth formatting the entire thing as text so that 0's don't drop off the front of part numbers
- Manufacturing Orders (MO's)
    - Contains: Manufacturing Orders from the past, user to decide how far back in time to reference
    - Export form: Use Data

        - Export fields (included): Product/Display Name, Product/Bill of Materials/BOM Lines
- Reordering Rules
    - Contains: export of all reordering rules in the database
    - Export form: Update Data
    - Export fields (included): Product/Name
- Sales Orders (SO's)
    - Contains: Sales Orders from the past, user to specify how far back in time to reference
    - Export form: Use Data
    - Export fields: Order Lines/Product/Display Name
- Purchase Orders (PO's)
    - Contains: Purchase Orders from the past, user to specify how far back in time to reference
    - Export form: Use Data
    - Export fields: Product/Display Name or Description or Product/Product.Display Name

All files must be saved as a .csv, and the program will error out if they aren't. Assuming everything works, four new files will be in the folder when it's done.

- Obsolete_update.csv
    - Every obsolete part, with updated names, activities, and product types.
    - Can be imported into Odoo(under products) to update the parts
    - Will have the same format as the imported All Parts file
- Questionable.csv
    - All parts that were marked as obsolete and were also on MO's, SO's, or PO's.
    - All parts that were marked as obsolete and weren't "Storable Product"
- Updated Reordering Rules.csv
    - All parts that had reordering rules, now marked to 0,0,1
    - Can be imported to Odoo (under reordering rules) to update the parts
- New Reordering Rules.csv
    - All parts that didn't have reordering rules, with new reordering rules of 0,0,1
    - Can be imported to Odoo (under reordering rules) to update the parts

- Has to be a separate import from the first Reordering Rules import

Preparing the Program

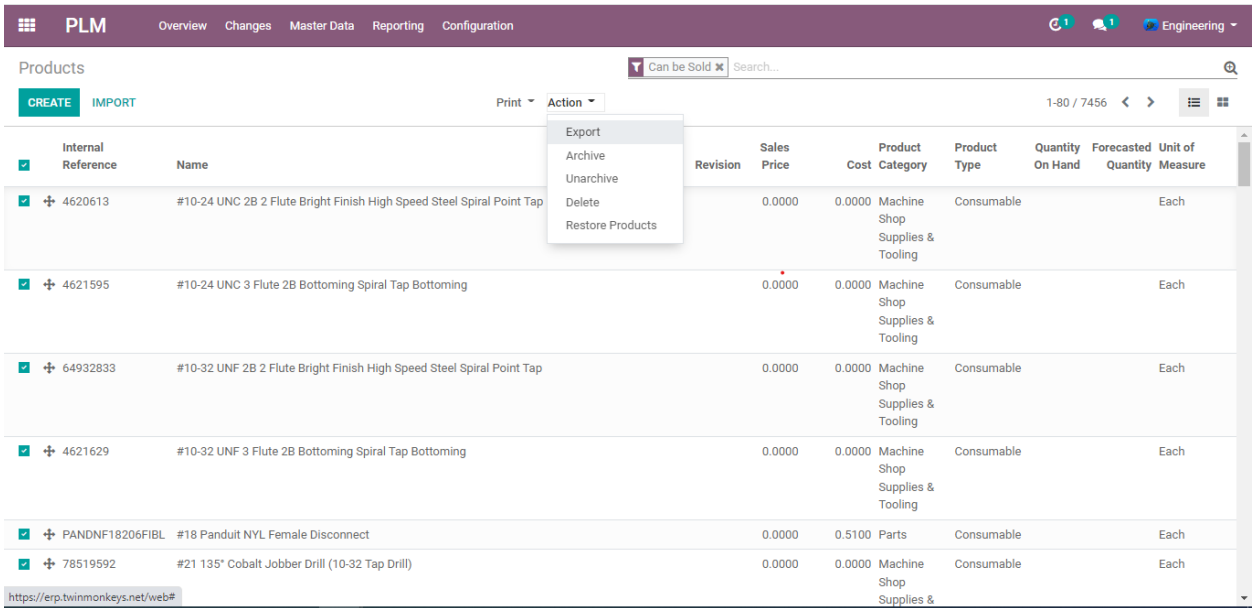
All that has to happen to prepare the program is to download it. It's pretty big, so it may take a few minutes. Remember that it will spit out new files, and if it's saved in a folder that has a lot of other files in it the new files may be hard to find. It may take a few tries and some googling to get windows to run it- I'm not paying for the certification that Windows's anti-malware software looks for.

Preparing the Data

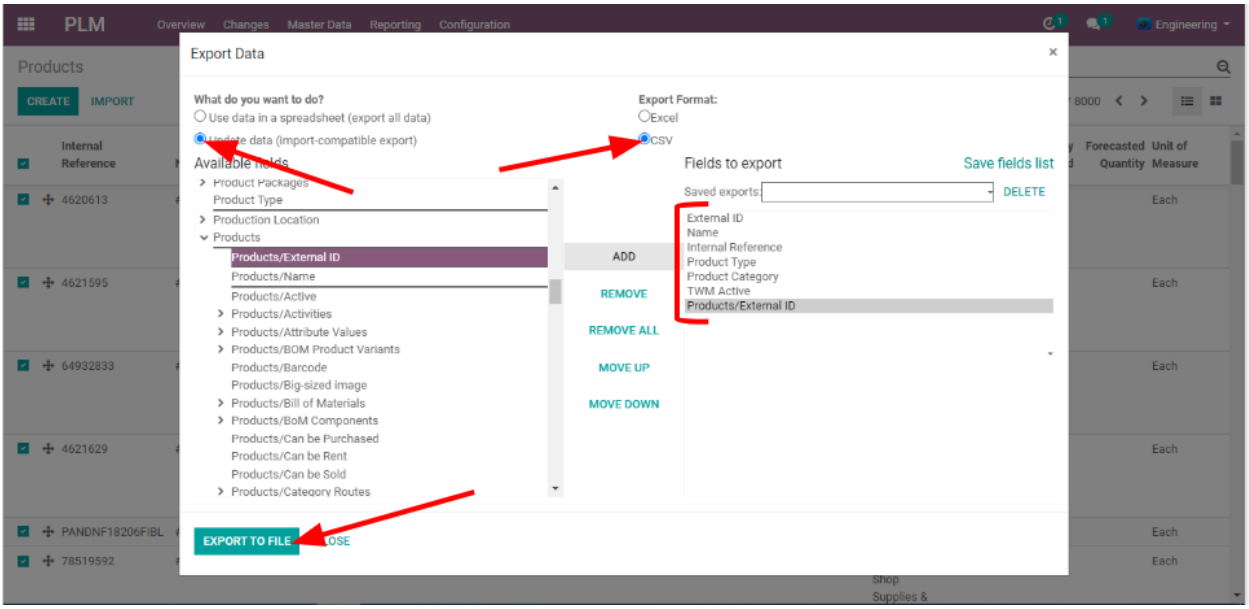
The next step is to gather what the program needs to calculate what parts are obsolete. There are seven spreadsheets that the program looks for as it runs. The first contains every part in the database, the second contains all the active parts, the third contains all parts that are to be excluded, the fourth all the recent Manufacturing Orders, the fifth all the Reordering Rules in the database, and the sixth and seventh are Sales Orders and Production Orders respectively.

All Parts

To create the first spreadsheet, select every part in the database and then export them.



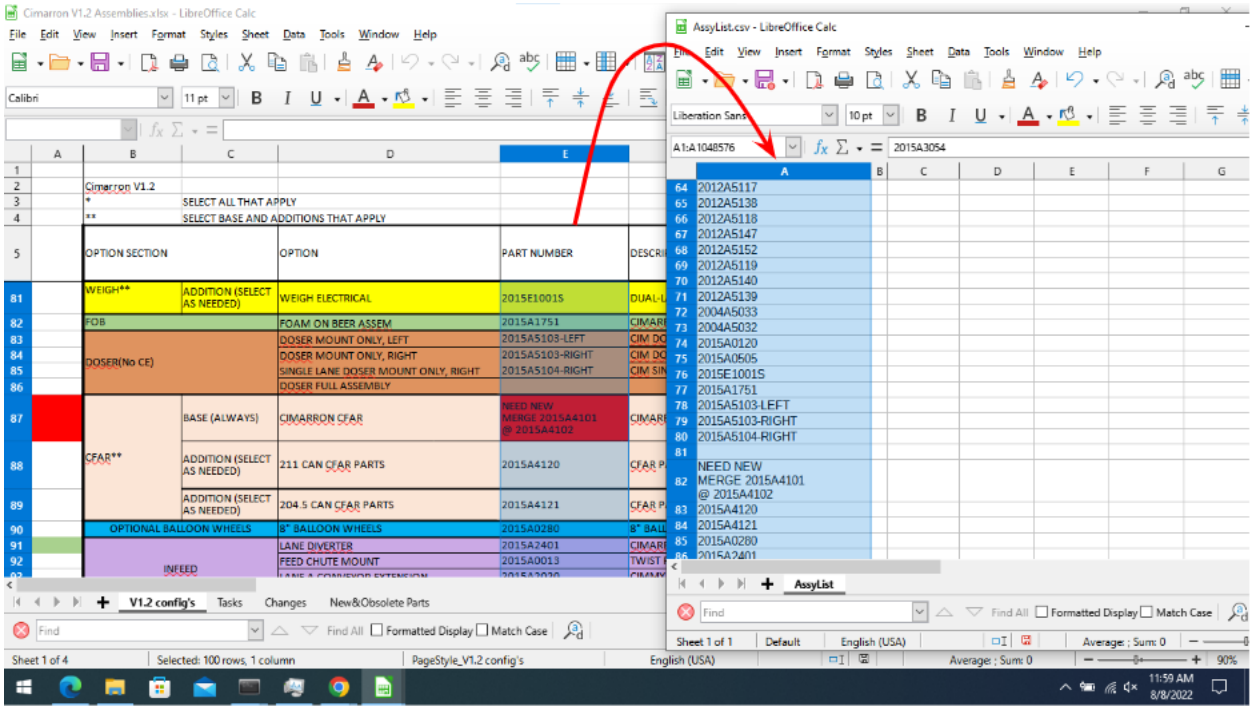
Make sure to select the Update Data and CSV options within the export page. The export fields are very important; if the program doesn't detect the seven it needs it won't accept the file. As shown in the following image, the necessary fields are (in no particular order): External ID, Internal Reference, Name, Product Type, Product Category, TWM Active, and Products/External ID. Other fields can be added if desired and the program will still run, but no action will be performed on or with any other provided data.



Once done, clicking export to file will save a file with a name something like “product.template.csv”. This file will need to be selected later, so either remember the name or rename it to something more useful.

The Star Lord BOM

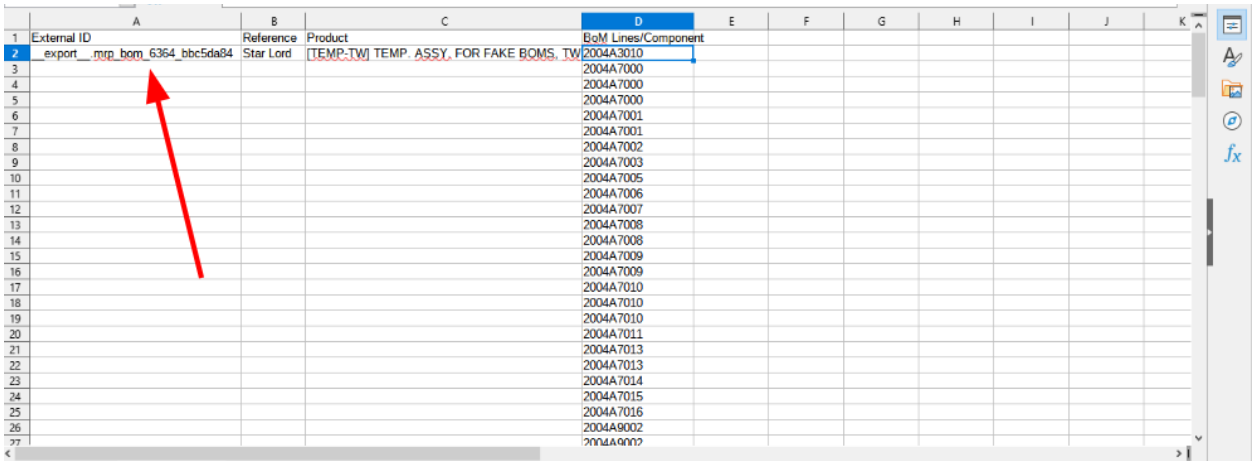
The second file the program needs contains every part that is currently not obsolete. While there is more than one way to generate such a list, the following one is recommended by engineering. First, go through the configuration spreadsheet (tmserver -> MonkeyBusiness -> Engineering -> Configurations) for each active machine and copy each assembly number into a separate spreadsheet. Make sure to only get the assembly numbers, and not other things like titles or multipliers. Duplicates throughout this BoM are fine (and most of the program as well), and definitely preferable to missing an assembly.



After this, go through every quotation template and put every assembly number in the templates into the spreadsheet. This can get tedious, which is what the SO_reader widget is for. To use the SO_reader, copy-paste all the text in each quotation template into the command window that is opened with the widget. Once all the text is in, press enter twice and it will print a list of all

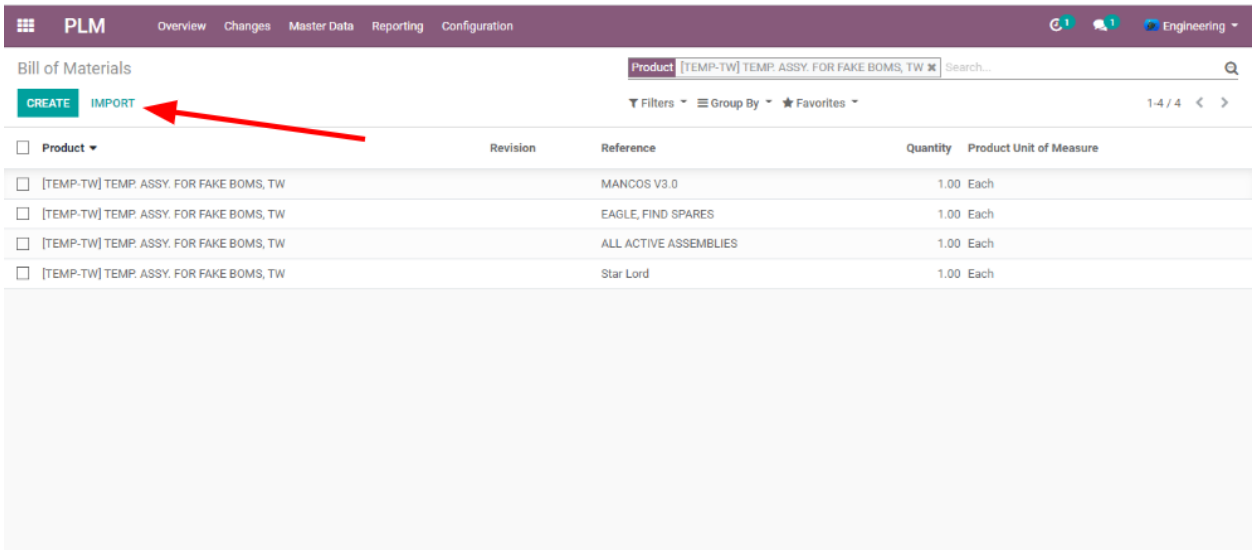
the text that's within brackets from the text that was provided. Then just copy-paste that list into the spreadsheet.

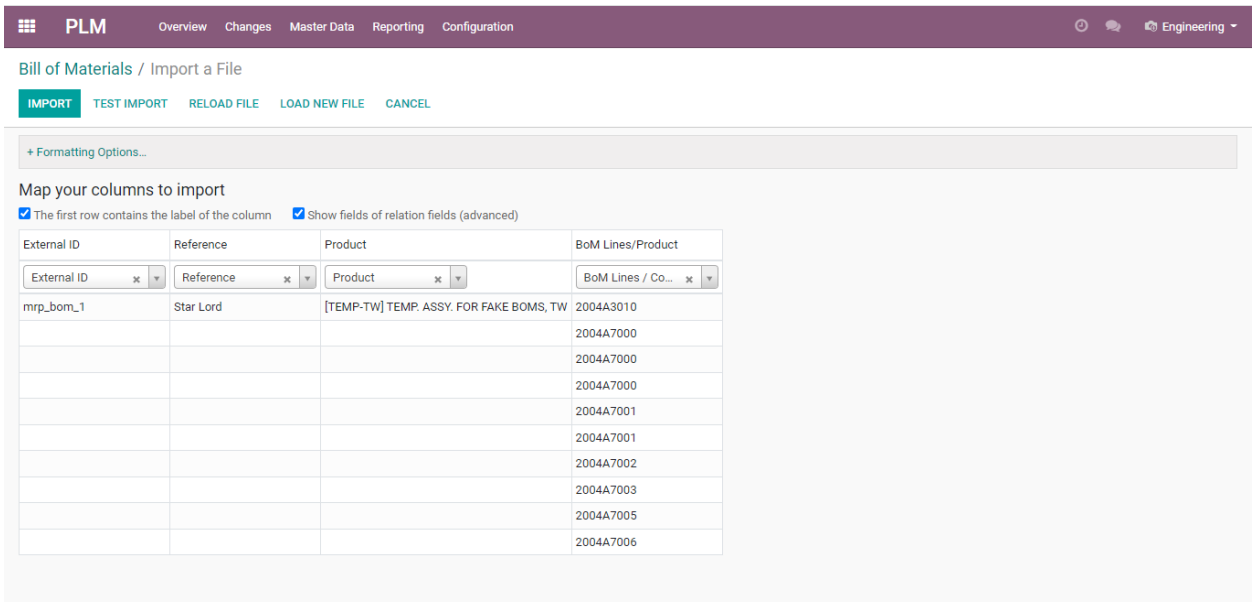
Once the spreadsheet has every current assembly on it, it needs to be imported into Odoo to create a BOM with every current assembly (i.e. the Star Lord BOM). To do this, format it the way the following image has it- with a column that is titled 'External ID', and another that is 'BoM Lines/ Component'. What is in the 'Product Column' affects where the BoM is saved, and the 'Reference' is optional but recommended to make the new BoM easier to find. If the following external id is already in the database, then the components will be added to that BoM. If it isn't, a new BoM will be created. If unsure of what some action will do, the test database is a great resource here.



	A	B	C	D
1	External ID	Reference	Product	BoM Lines/Component
2	__export__.mrp_bom_6364_bbc5da84	Star Lord	[TEMP-TW] TEMP, ASSY, FOR FAKE BOMS, TW	2004A3010
3				2004A7000
4				2004A7000
5				2004A7000
6				2004A7001
7				2004A7001
8				2004A7002
9				2004A7003
10				2004A7005
11				2004A7006
12				2004A7007
13				2004A7008
14				2004A7008
15				2004A7009
16				2004A7009
17				2004A7010
18				2004A7010
19				2004A7010
20				2004A7011
21				2004A7013
22				2004A7013
23				2004A7014
24				2004A7015
25				2004A7016
26				2004A9002

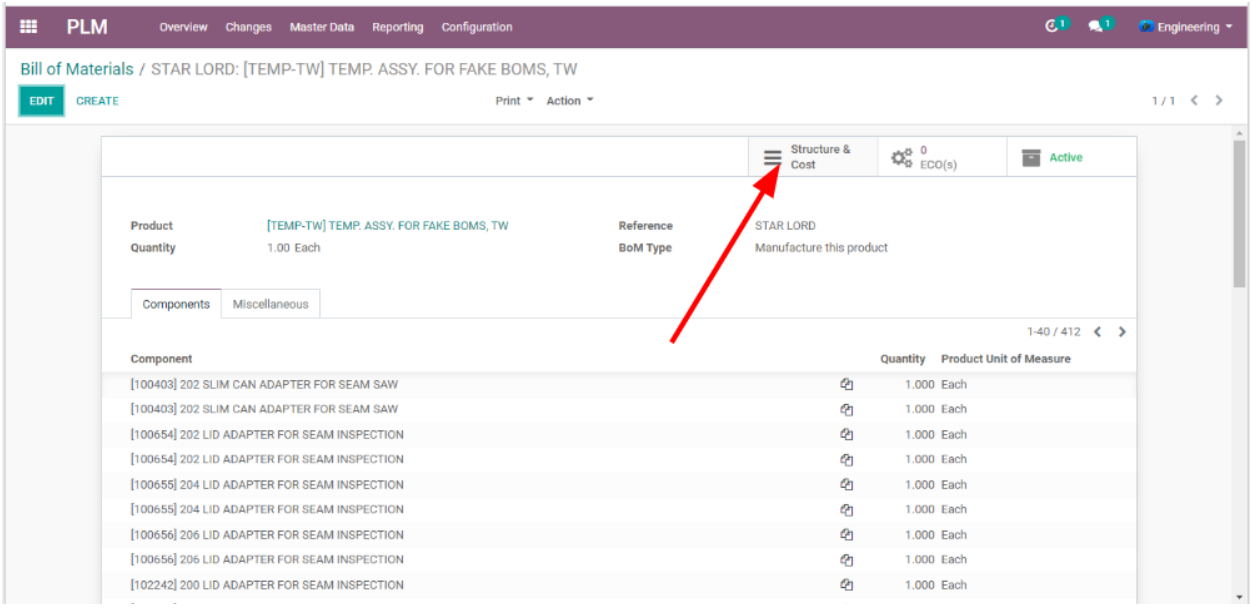
Select where the new BoM should be located, or where the one to be updated is, and click upload, select the file, and click import, making sure that the correct headers are loaded into Odoo. The following images (and previous image) are for updating the BoM with the reference 'Star Lord'.



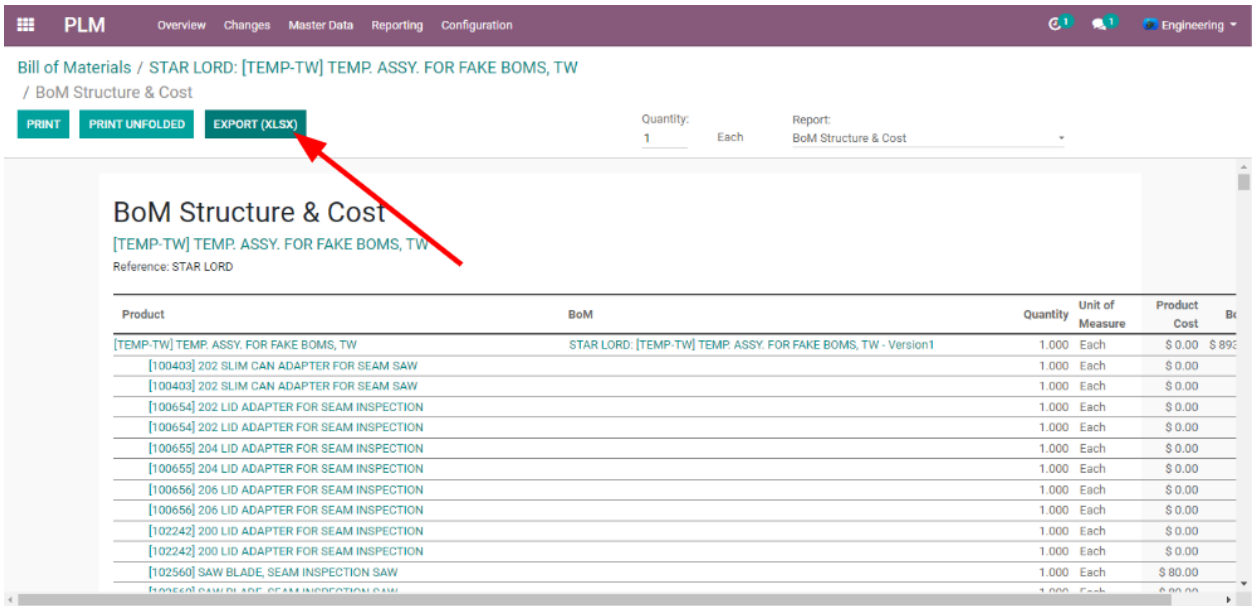


If it isn't certain that every assembly in the spreadsheet is a viable product in Odoo, the 'TEST IMPORT' button will load faster and give similar error information.

Once this BOM has been loaded, it needs to be exported with all the part numbers in it. Open the BOM, then click on the Structure & Cost button.



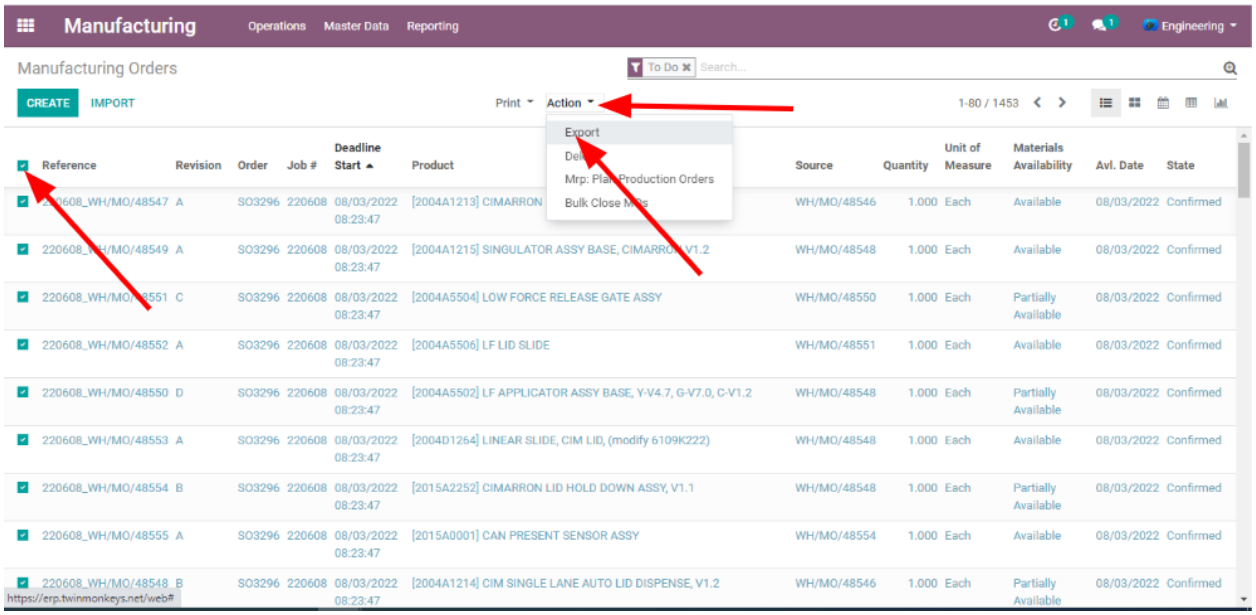
Next click on the export button.



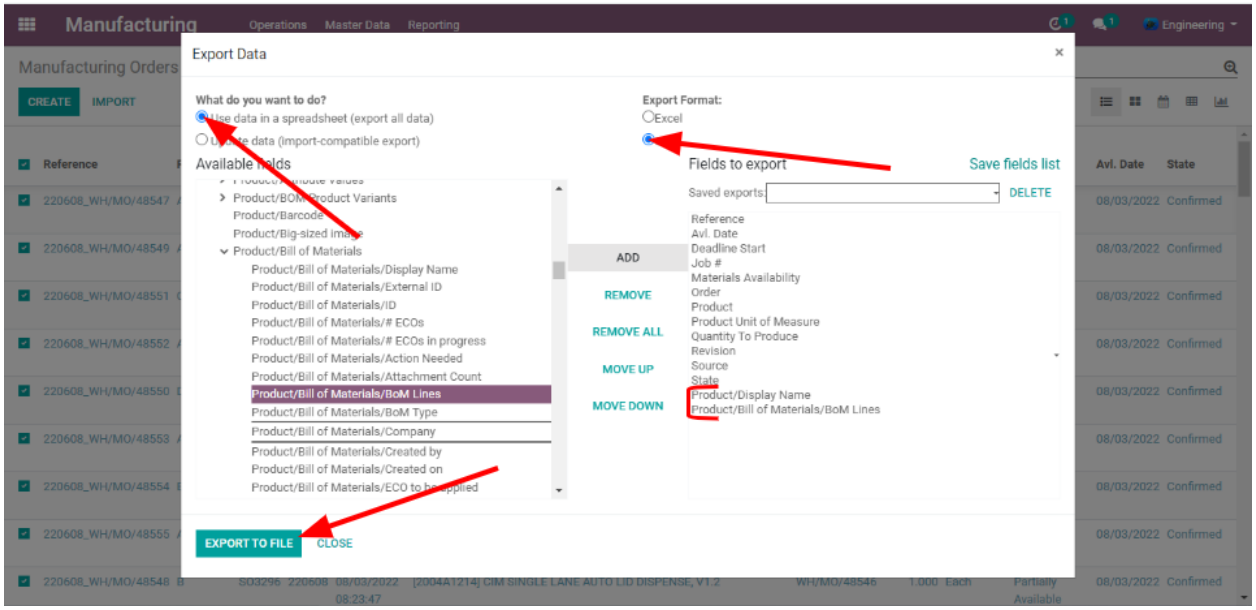
This will save an excel spreadsheet to the computer. It will probably be named something close to “bom_structure_and_cost.xlsx”. Make sure to open the file in Excel, select ‘Save As”, select .csv, and save the file before running the program since the program can’t read Excel files.

Manufacturing Orders

To load the manufacturing orders (MO's) into the program, select every Manufacturing Order with the checkbox on the top left. Then Action -> Export.



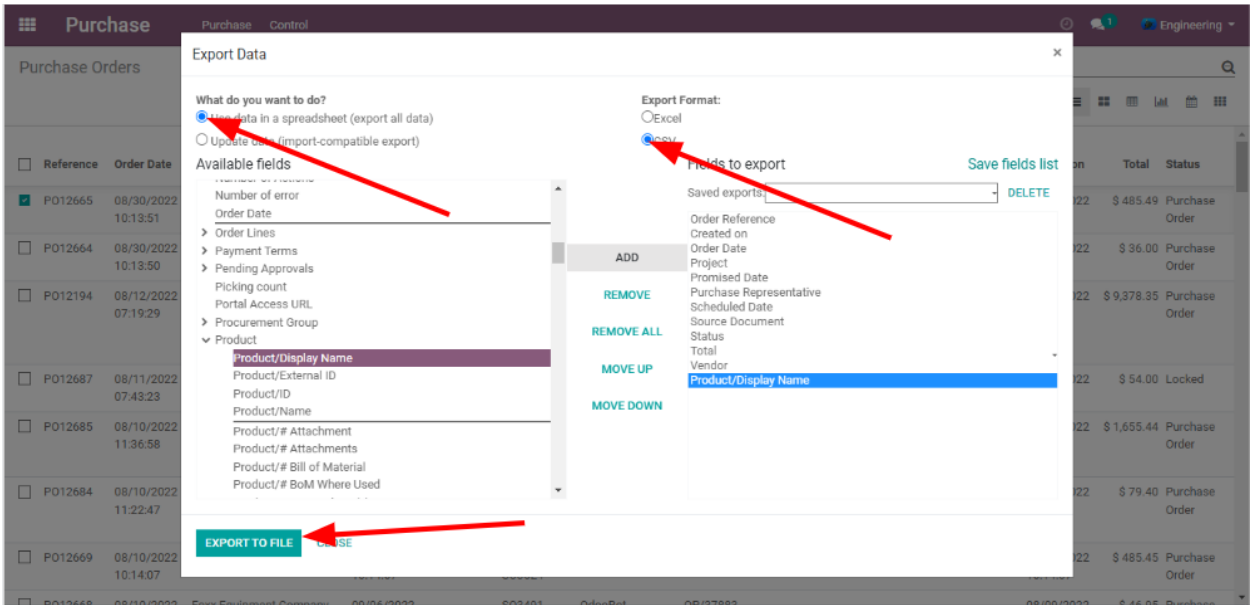
Make sure that "Products/Display Name" as well as the "Product/Bill of Materials/BOM Lines" is selected in the export fields. This export, unlike the first, has to have "Use data…" selected under the "What do you want to do" part. Click "Export to File", and it should be ready. It may be worth removing some unnecessary fields as many fields can cause the export to take a while.



This method exports all saved MO's. The user may want to filter out the ones they don't want using Odoo or Excel.

Sales Orders and Purchase Orders

Go into the Sales/Purchase part of Odoo, and select everything -> action -> export like the rest. Use data, CSV, and make sure to add Product/Display Name to the output for purchase, Order Lines/Product/Display Name for sales. Similar to the Manufacturing Orders, the user will probably have to do some sorting in Odoo and or Excel.



Excluded Parts

Some parts and assemblies usually end up getting excluded from the final obsolete list despite the fact that they weren't/aren't on the Star Lord BOM. To format this spreadsheet, put all the part numbers/assembly numbers (just the numbers, not the part descriptions) in the furthest left column. The other columns can have anything in them, so part names can go next to the Internal IDs without messing up the program. Save it as a .csv, using the "Save As" option in excel. Similar to the Star Lord BoM, duplicates don't really hurt anything. Blank space also doesn't hurt, so if it's helpful to space things out a little- no harm done.

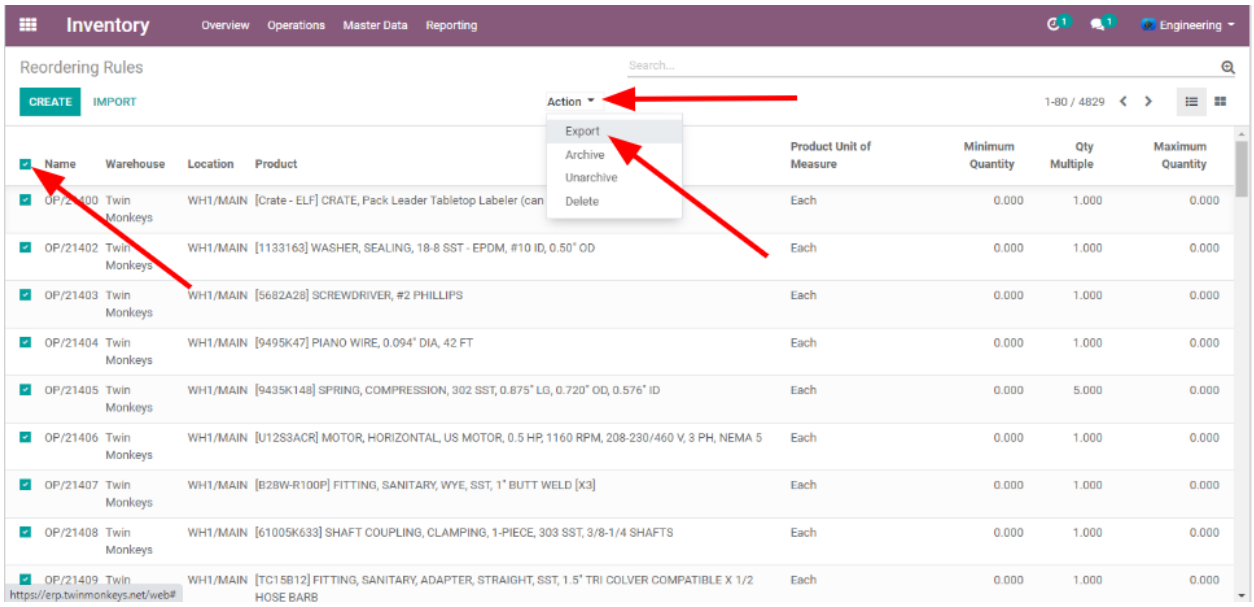
A quick note on the file- it is possible that some part numbers that only contain numbers will be seen, in excel's eyes, as an integer. This is fine, unless the part number starts with a zero. In that case the zero will be dropped and the .exe will not find the new part number. To avoid this try to get excel to format all cells (or at least the cells that will contain part numbers) to be text/strings/something that isn't numeric and doesn't change the format. If it does change it isn't the end of the world- the program lists out the parts from the excluded file that it didn't find and those can be edited manually in the excluded file to match what Odoo has on another iteration

Hopefully nobody ever needs this information, but if the list being copied in is in the format [PART NUMBER] PART NAME (the part name is attached to the part number, which is in brackets) the program can handle that and will extract the part number from the string. It can be mixed with regular part numbers as well- the entire column doesn't have to be either format.

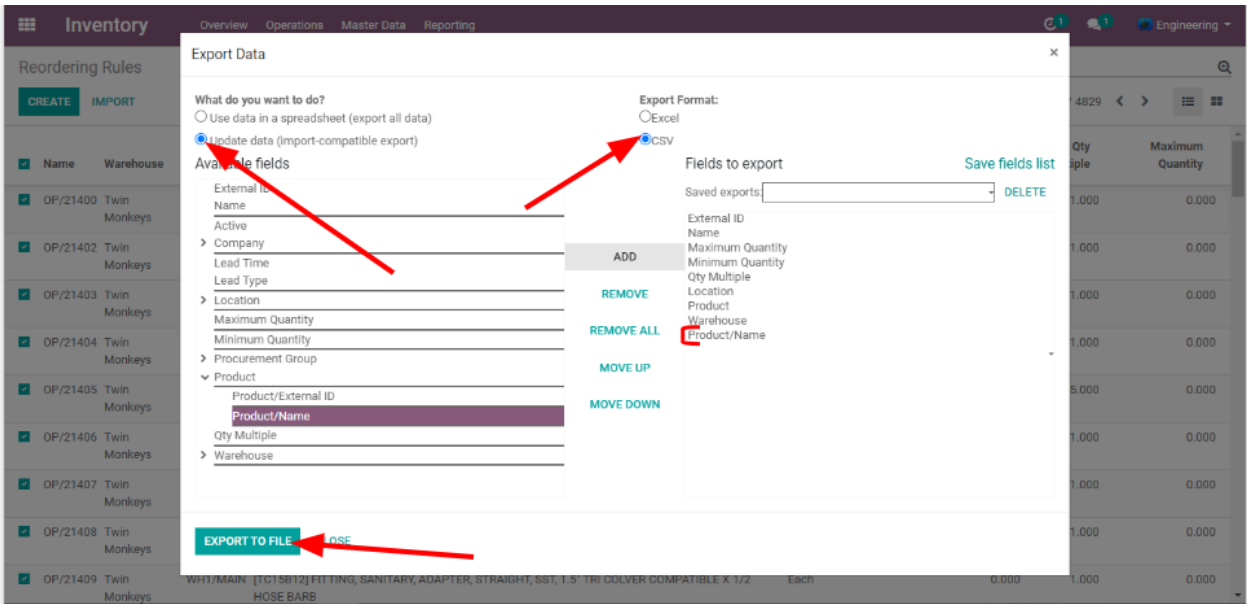
Reordering Rules

These are optional. If the file isn't included in the folder, the program will still run, but it won't give output to update the reordering rules.

Go into Reordering Rules, click the top left checkbox to select every reordering rule, then Action -> Export.



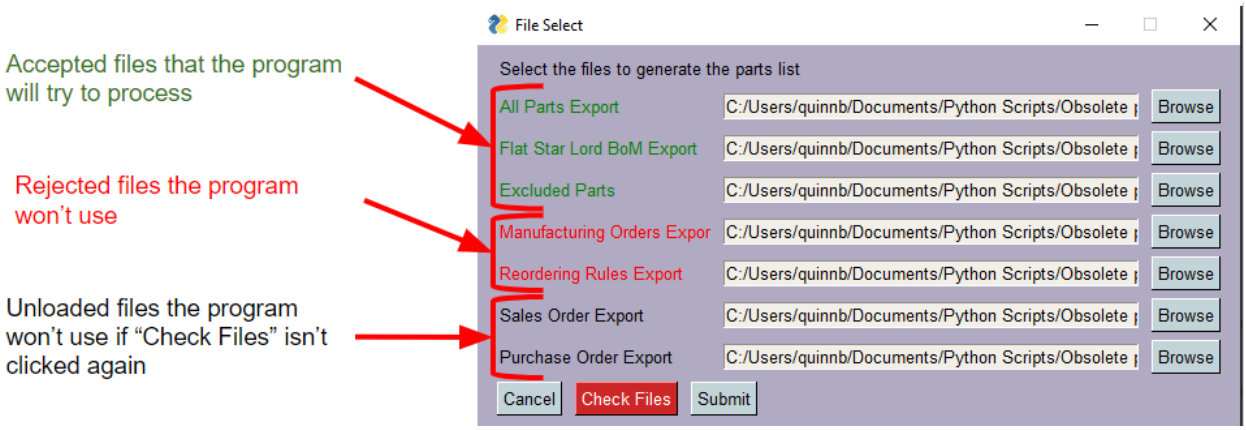
Select “Update data” and “CSV”, and make sure to add “Product/Name” to the export fields before exporting it to a file.



Note: The Product/Name category here will export the same External IDs for parts/assemblies that the Products/External ID field will have in the All parts export.

Using the Program / Output

When the program launches, it will need a few seconds before it does anything. Once it starts, a menu prompting the user to select files for the program to process will appear. As files are selected, if the “check files” button is pressed, the text at the title of the file will turn either green or red. Green means that the file was accepted, red is rejected. Especially in the case of the second file (the Star Lord BoM), the user can select a file that won’t work and the program will most likely not catch it. It is important to double check that the selected files are correct, as the program can’t execute successfully without the proper files selected. Or even worse, the program will execute without error but the output will be incorrect. If a file isn’t green when “Submit” is clicked, it means that the program hasn’t seen that file and won’t use it as it executes.

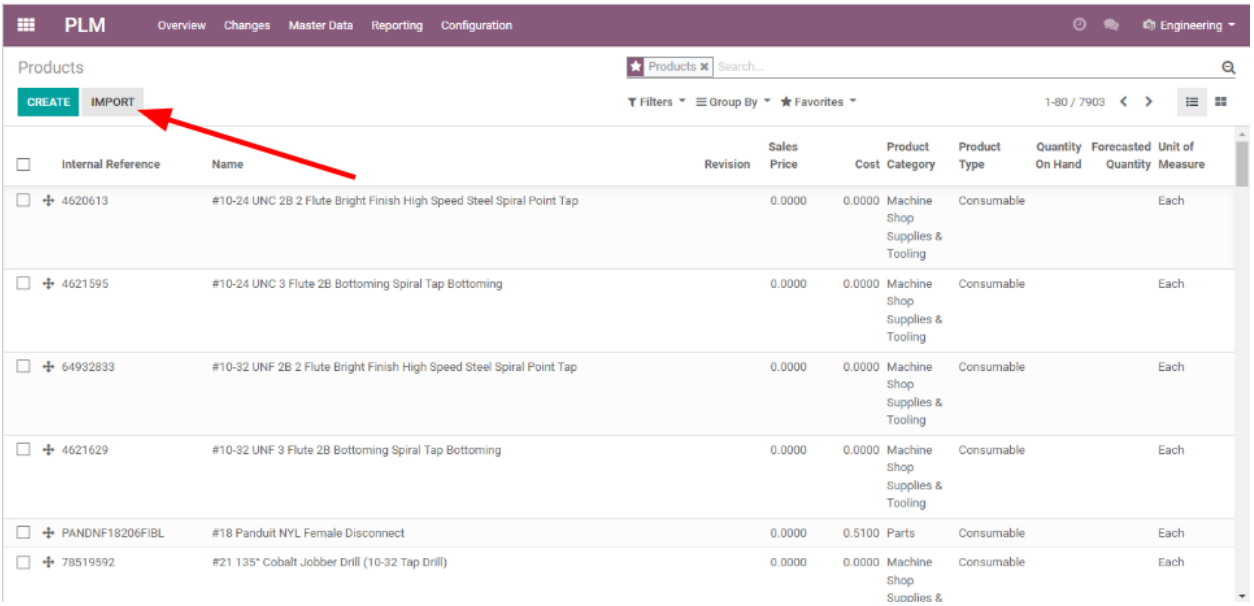


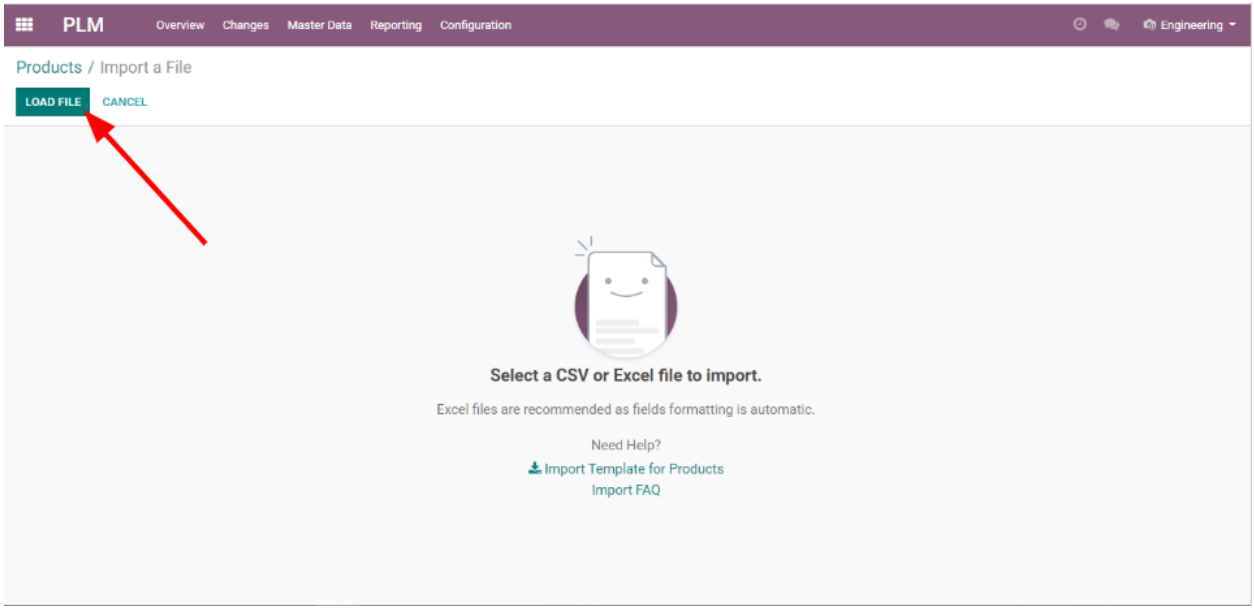
The "Submit" button will not appear until both the All Parts and Stary Lord BOM fields have valid inputs in them.

Obsolete_update.csv

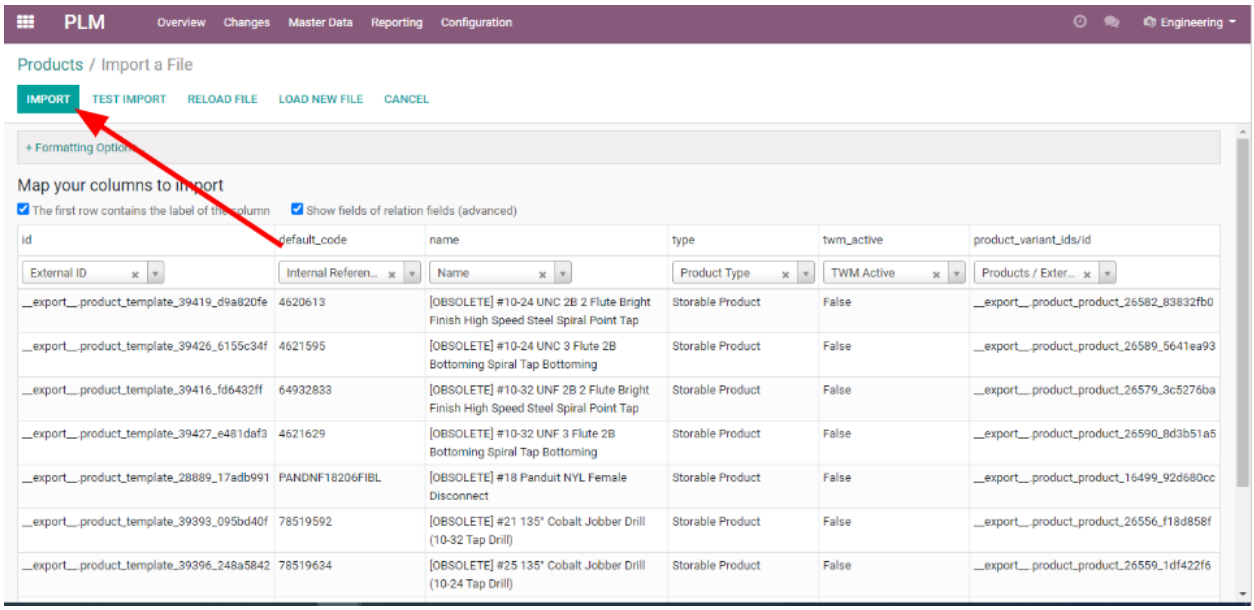
The goal of the program- if this file is imported into Odoo it will update the obsolete parts. This update includes changing the name to have [OBSOLETE] at the start of the name, changing the part to not active if it wasn't already, and making it a Storable Product if it wasn't already. Any other fields that were selected in the All Parts export will appear here unchanged.

To update Odoo, go into products and click the import button, then the Load File button.





Select the Obsolete_update.csv from the computer. Then select import to update the database. Similar to importing the Star Lord BOM, the test import button can be helpful to find out if all the information is valid without changing anything in Odoo.



Questionable.csv

Contains all parts that were marked obsolete and were on MO’s, SO’s, PO’s, or weren’t Storable Products. The goal here is that the user can copy the parts from the questionable file to the

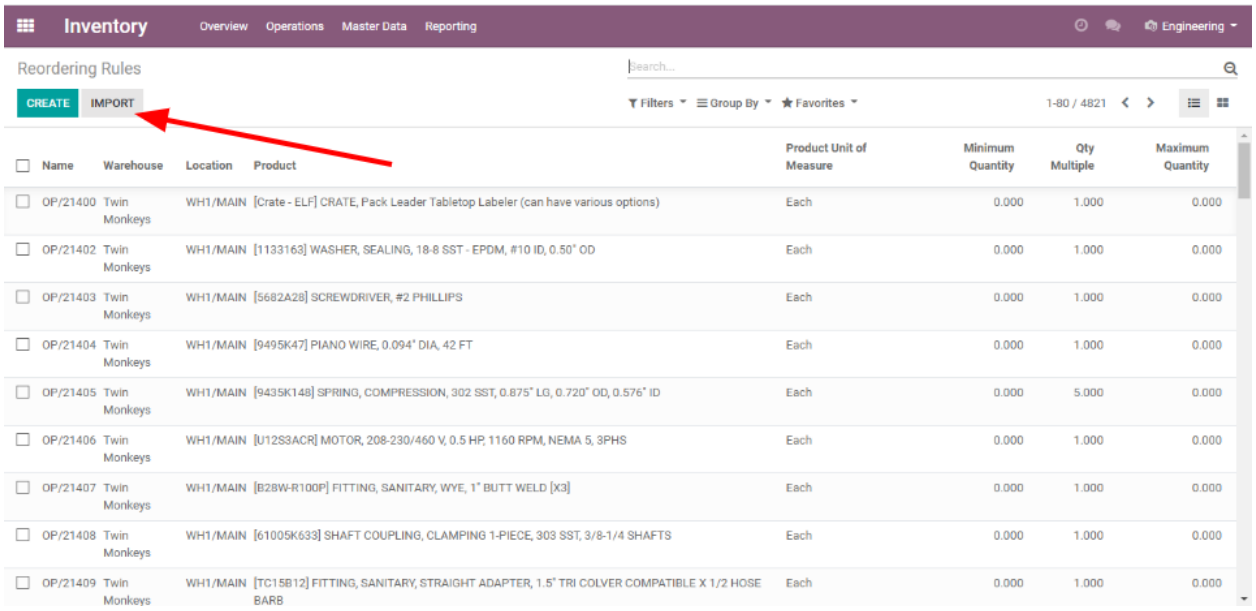
excluded file and run the program again without having to change anything on the Star Lord BOM.
This file, depending on what program reads it, may have the same problem with the computer reading part numbers as integers as was seen in the Excluded parts file.

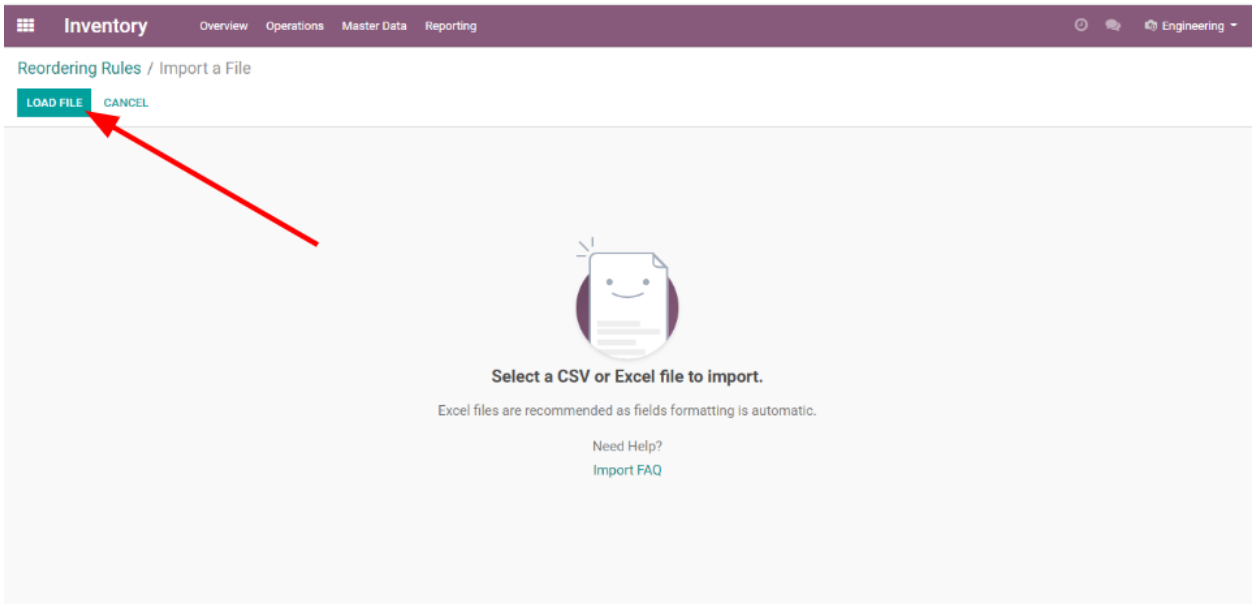
Reordering Rules

The program spits out two files that are relevant to reordering rules. This is because it isn't easy (to my knowledge) to both update reordering rules and create new rules in the same import. So the following process should be done for each of the two files (Updated Reordering Rules.csv and New Reordering Rules.csv).

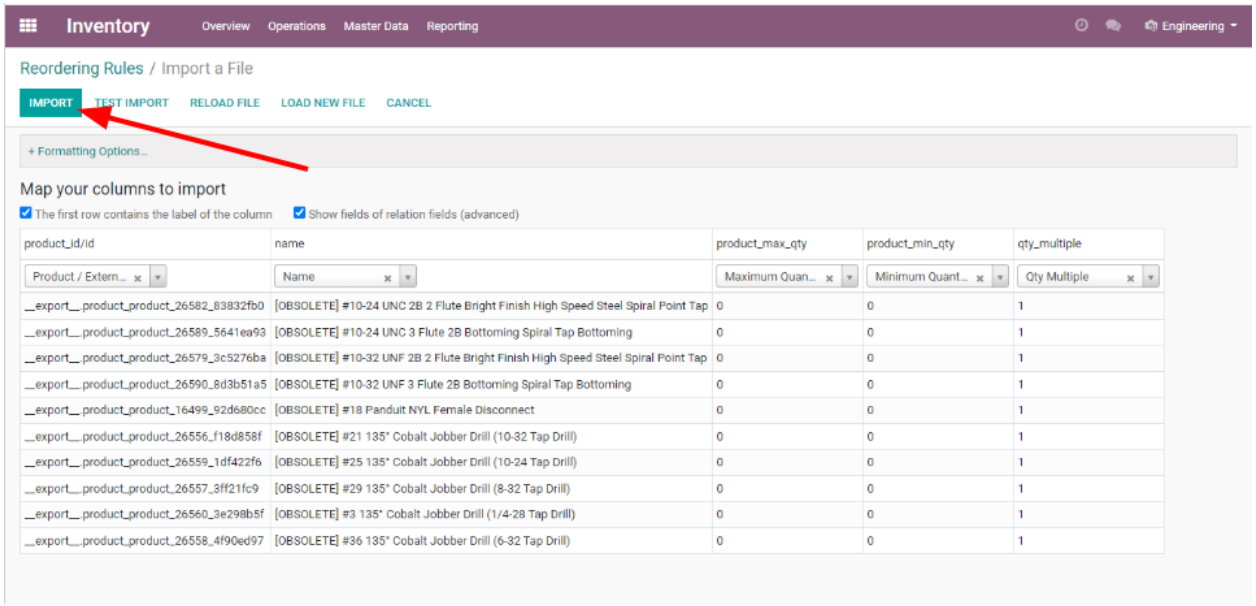
A brief reminder about reordering rules- a part/assembly can have more than one reordering rule under it. So even if the rules are already set under a part, "updating" said rules will only create new ones as opposed to changing the old if the reordering rules external ID isn't specified. Hence the reason the reordering rules in Odoo have their own set of external IDs, which are separate from the Product IDs.

In Reordering Rules, select import and then load file.





Select one of the two files, and press open. Then press import again and that file is done. Again, the test import may be useful.



product_id/id	name	product_max_qty	product_min_qty	qty_multiple
Product / Extern...	Name	Maximum Quan...	Minimum Quant...	Qty Multiple
__export__.product_product_26582_83832fb0	[OBSOLETE] #10-24 UNC 2B 2 Flute Bright Finish High Speed Steel Spiral Point Tap	0	0	1
__export__.product_product_26589_5641ea93	[OBSOLETE] #10-24 UNC 3 Flute 2B Bottoming Spiral Tap Bottoming	0	0	1
__export__.product_product_26579_3c5276ba	[OBSOLETE] #10-32 UNF 2B 2 Flute Bright Finish High Speed Steel Spiral Point Tap	0	0	1
__export__.product_product_26590_8d3b51a5	[OBSOLETE] #10-32 UNF 3 Flute 2B Bottoming Spiral Tap Bottoming	0	0	1
__export__.product_product_16499_92d680cc	[OBSOLETE] #18 Panduit NYL Female Disconnect	0	0	1
__export__.product_product_26556_f18d858f	[OBSOLETE] #21 135° Cobalt Jobber Drill (10-32 Tap Drill)	0	0	1
__export__.product_product_26559_1df422f6	[OBSOLETE] #25 135° Cobalt Jobber Drill (10-24 Tap Drill)	0	0	1
__export__.product_product_26557_3ff21fc9	[OBSOLETE] #29 135° Cobalt Jobber Drill (8-32 Tap Drill)	0	0	1
__export__.product_product_26560_3e298b5f	[OBSOLETE] #3 135° Cobalt Jobber Drill (1/4-28 Tap Drill)	0	0	1
__export__.product_product_26558_4f90ed97	[OBSOLETE] #36 135° Cobalt Jobber Drill (6-32 Tap Drill)	0	0	1