D User Guideline of the Re-planning Tool

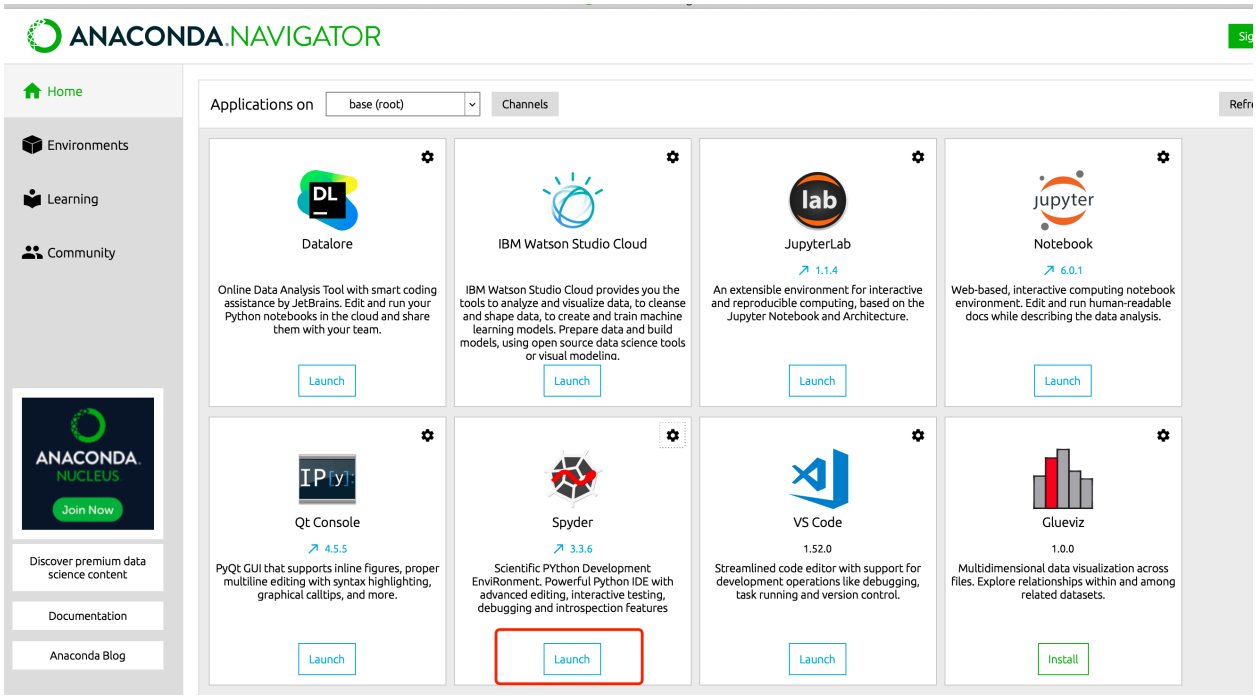
As described in Section 3.5, the replanning of the jobs follows specific heuristic rules. However, the company used to replan it manually in Microsoft excel files one to three times a week. Once a job's start date or CT needs to be changed, all the following jobs' plans must be modified accordingly. This manual process is cumbersome and time-consuming. A scheduler usually takes four to six hours to finish the replans. After understanding their rescheduling rules, we wrote a package via Python with an excel input interface. In this tool, a user does not need to know anything about coding but the initial jobs that need rescheduled. The remaining jobs will be rescheduled automatically. With the same number of jobs as manually replan, the tool can finish the replanning within five minutes. The following is the user guidance of the tool.

Task 1 Download and Install the Python Running App

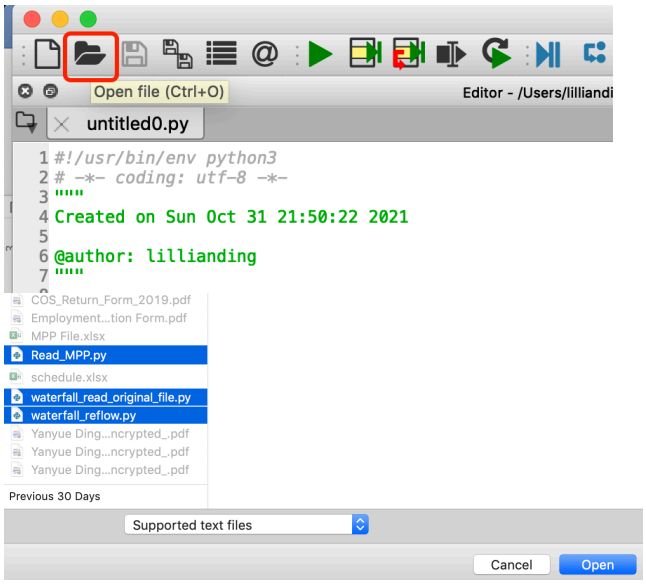
Step 1 Open the Python Running App. It can be Anaconda or Microsoft Visual Studio.

Step 2 If you use Anaconda, for example, search the App name "Anaconda Navigator" in your computer and open the App.

Step 3 When the Anaconda is open, click Launch in "Spyder" to open the running platform.



Step 4 Close all the opened code except for "untitled0.py". Open the code you want to run. Usually, it will be "Read_MPP.py", "waterfall_read_original_file.py", and "waterfall_reflow.py".



Task2 Transfer from Excel Tool to Python Tool

If it is your first time to use the scheduling tool encoded in Python Environment, you need to first transfer the current schedule in the Excel Tool and then do the reflow.

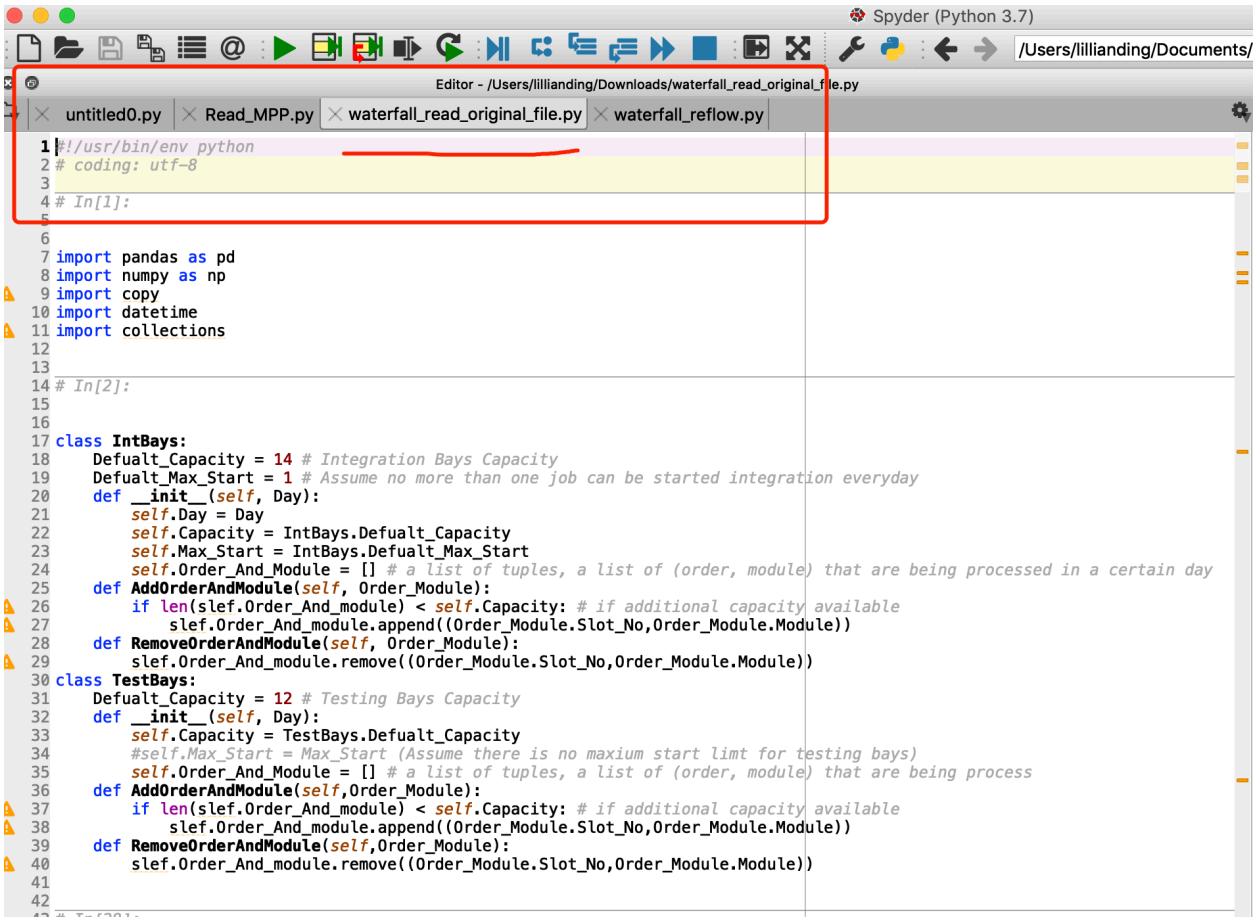
If it is not, please ignore this Section and go to Task 3.

Step 1 Download all the files from the Google Drive. Make sure all the files are saved in the same folder. In General, you can simply download the "Scheduling Tool and Input Files" Folder from the drive.

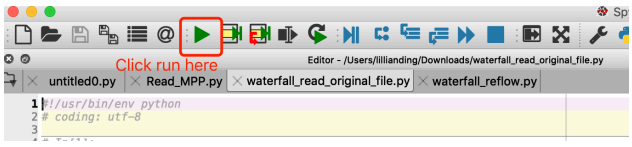
If you can not download this folder, open the folder, and download the files one by one. Do not omit any file and make sure all the files are saved in the same folder.

Step 2 Save the most current waterfall Excel that you want to do the reflow as "CMP AUS Waterfall.xlsx". Please make sure it is saved in xlsx format and the spaces in the file name is correct.

Step 3 Go to the Spyder and make sure you will be running the "waterfall_read_original_file.py" Python code.



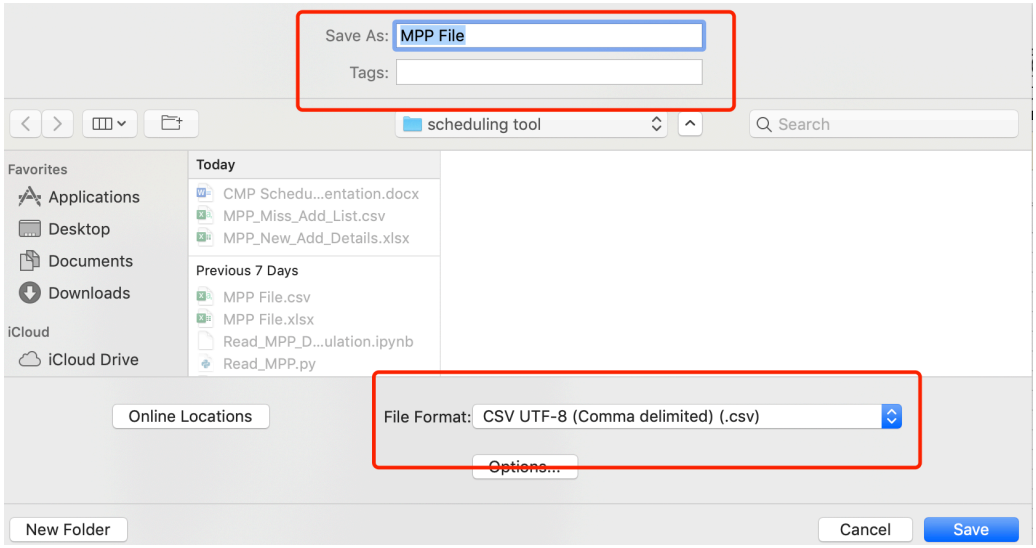
Step 4 Click "Run" in the Tab bar, your current waterfall in the Excel format will be transferred into a format that the new tool can understand. The transferred data will be saved in another excel named "Waterfall Input.xlsx" which can be the input file for the future reflows. (In Windows environment, make sure you closed the input excel before running the Python code.)



Task 3 Find the difference between current jobs with the newest MPP file

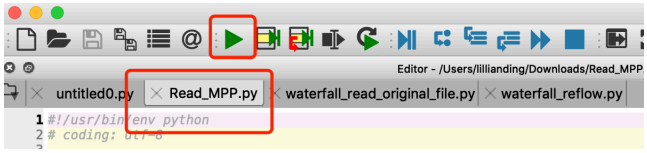
If you want to find if there are any miss-out or add-ins in your newest MPP file, go to Spyder and make sure you are running "Read_MPP.py" Python code. Note that the New tool will choose only "EOP Fiscal Quarter" include and after 2021Q4.

Step 1 Open your new MPP file and copy the data from the upper left to the lower right. Open a new excel file and paste the data into the new excel file. Save the new excel file as "MPP File.csv". Make sure "MPP File.csv" is saved in the same folder with the code.

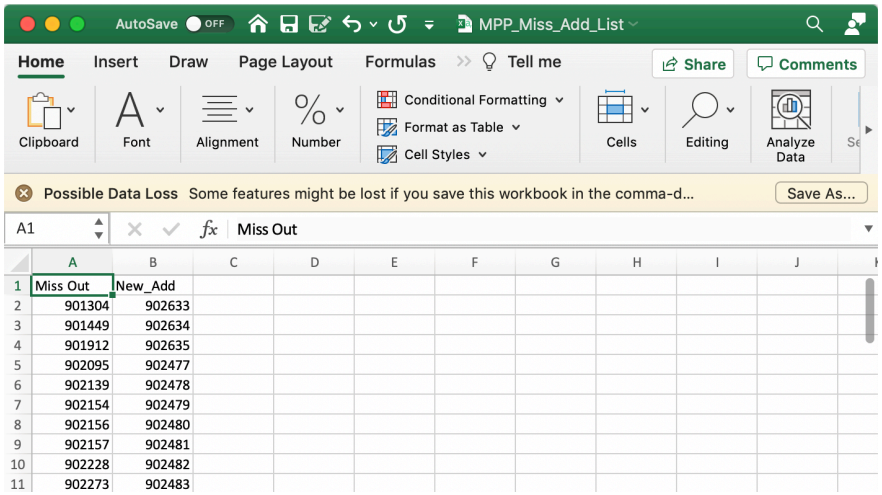


Step 2 Save your current schedule as "Waterfall Input.xlsx". It can be the output file you get in Task 2 if it is the first time you use the Python tool. It can also be the "Reflowed Output.xlsx" or another name of the reflowed schedule that you saved yesterday.

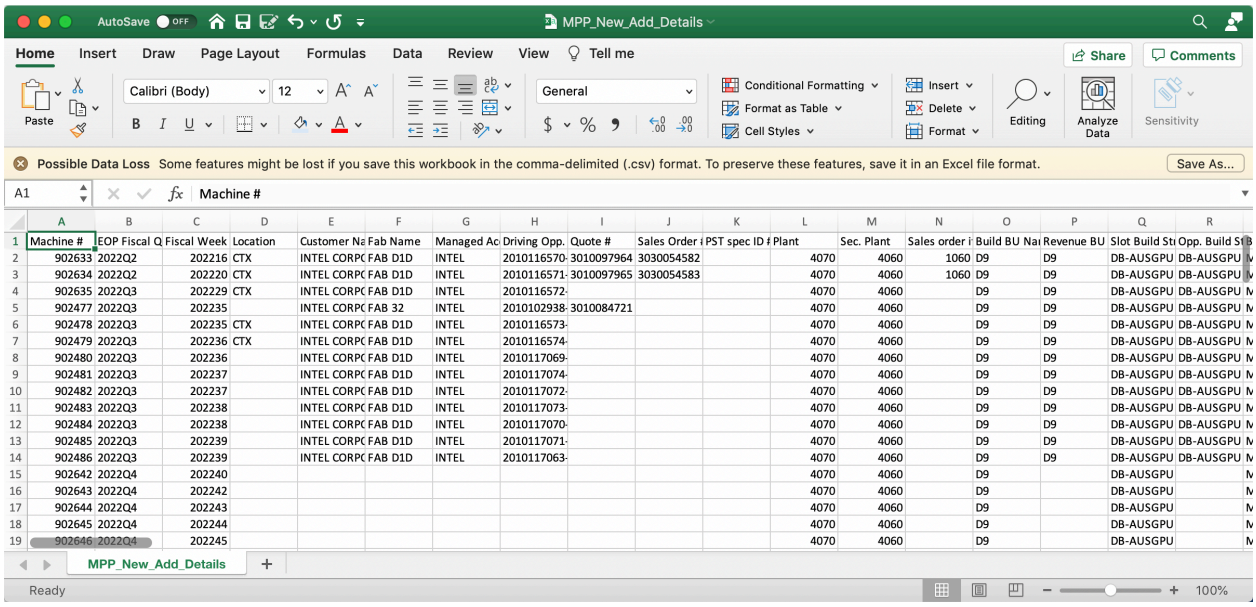
Step 3 Click "Run" in the Python Code, two files will be saved in the folder where you put all the code in.



There are two output files when the code is run. "MPP_Miss_Add_List.csv" records the list of "Slot #" that is newly added in the MPP file or should be deleted in the current schedule "Waterfall Input.xlsx" file.



Another output file "MPP_New_Add_Details.csv" drags the MPP data from the files for all the newly added jobs.

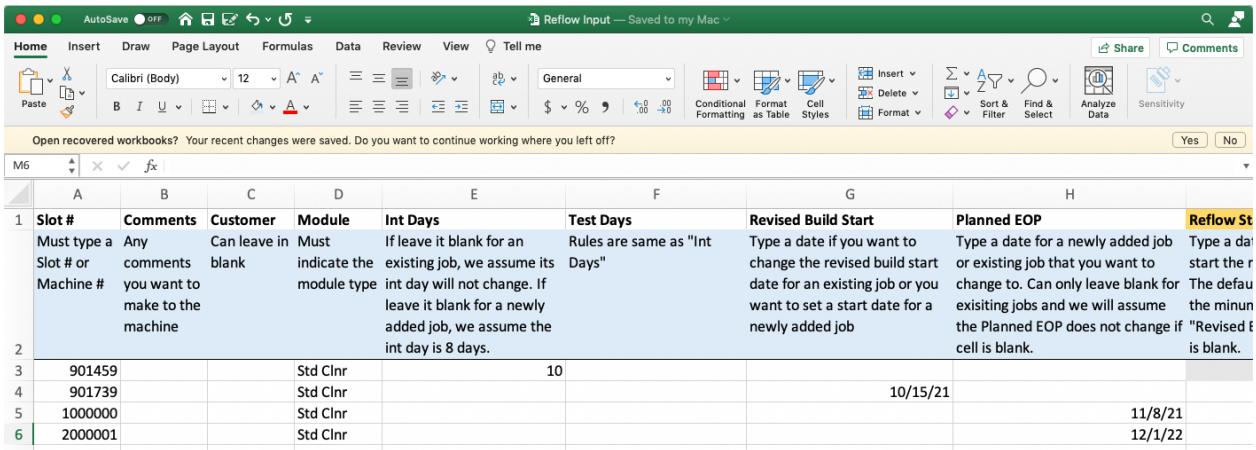


Machine #	EOP Fiscal Q	Fiscal Week	Location	Customer Na	Fab Name	Managed Ac	Driving Opp.	Quote #	Sales Order i	PST spec ID #	Plant	Sec. Plant	Sales order i	Build BU Na	Revenue BU	Slot Build Sti	Opp. Build St
902633	2022Q2	202216	CTX	INTEL CORPC	FAB D1D	INTEL	2010116570-	3010097964	3030054582		4070	4060	1060	D9	D9	DB-AUSGPU	DB-AUSGPU
902634	2022Q2	202220	CTX	INTEL CORPC	FAB D1D	INTEL	2010116571-	3010097965	3030054583		4070	4060	1060	D9	D9	DB-AUSGPU	DB-AUSGPU
902635	2022Q3	202229	CTX	INTEL CORPC	FAB D1D	INTEL	2010116572-				4070	4060		D9	D9	DB-AUSGPU	DB-AUSGPU
902477	2022Q3	202235		INTEL CORPC	FAB 32	INTEL	2010102938-	3010084721			4070	4060		D9	D9	DB-AUSGPU	DB-AUSGPU
902478	2022Q3	202235	CTX	INTEL CORPC	FAB D1D	INTEL	2010116573-				4070	4060		D9	D9	DB-AUSGPU	DB-AUSGPU
902479	2022Q3	202236	CTX	INTEL CORPC	FAB D1D	INTEL	2010116574-				4070	4060		D9	D9	DB-AUSGPU	DB-AUSGPU
902480	2022Q3	202236		INTEL CORPC	FAB D1D	INTEL	2010117069-				4070	4060		D9	D9	DB-AUSGPU	DB-AUSGPU
902481	2022Q3	202237		INTEL CORPC	FAB D1D	INTEL	2010117074-				4070	4060		D9	D9	DB-AUSGPU	DB-AUSGPU
902482	2022Q3	202237		INTEL CORPC	FAB D1D	INTEL	2010117072-				4070	4060		D9	D9	DB-AUSGPU	DB-AUSGPU
902483	2022Q3	202238		INTEL CORPC	FAB D1D	INTEL	2010117073-				4070	4060		D9	D9	DB-AUSGPU	DB-AUSGPU
902484	2022Q3	202238		INTEL CORPC	FAB D1D	INTEL	2010117070-				4070	4060		D9	D9	DB-AUSGPU	DB-AUSGPU
902485	2022Q3	202239		INTEL CORPC	FAB D1D	INTEL	2010117071-				4070	4060		D9	D9	DB-AUSGPU	DB-AUSGPU
902486	2022Q3	202239		INTEL CORPC	FAB D1D	INTEL	2010117063-				4070	4060		D9	D9	DB-AUSGPU	DB-AUSGPU
902642	2022Q4	202240									4070	4060		D9		DB-AUSGPU	
902643	2022Q4	202242									4070	4060		D9		DB-AUSGPU	
902644	2022Q4	202243									4070	4060		D9		DB-AUSGPU	
902645	2022Q4	202244									4070	4060		D9		DB-AUSGPU	
902646	2022Q4	202245									4070	4060		D9		DB-AUSGPU	

Task 4 Do the Reflow

Task 4.1 Reflow input platform

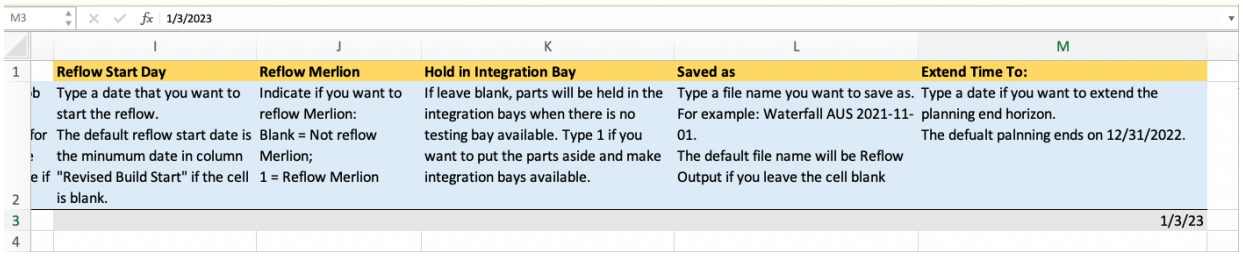
The reflow input platform is the excel file “Reflow Input.xlsx”. We can append the reflow information starting from row 3. Row 1 is the explanation for different columns.



Slot #	Comments	Customer	Module	Int Days	Test Days	Revised Build Start	Planned EOP	Reflow St
Must type a Slot # or Machine #	Any comments you want to make to the machine	Can leave in blank	Must indicate the module type	If leave it blank for an existing job, we assume its int day will not change. If leave it blank for a newly added job, we assume the int day is 8 days.	Rules are same as "Int Days"	Type a date if you want to change the revised build start date for an existing job or you want to set a start date for a newly added job	Type a date for a newly added job or existing job that you want to change to. Can only leave blank for exisiting jobs and we will assume the Planned EOP does not change if cell is blank.	Type a da start the r The defau the minun "Revised E is blank.
901459			Std Clnr	10				
901739			Std Clnr			10/15/21		
1000000			Std Clnr				11/8/21	
2000001			Std Clnr				12/1/22	

For all the jobs, you must fill out “Slot #” and “Module”. For new jobs, you must fill out “Planned EOP”. If you leave the cells blank for one job, the default value will be the value explained in row 1 colored with blue background.

There are two different input values: local input values for all jobs and global input values that indicate the reflow logic. Each job, a pair of “Slot #” and “Module”, has a set of input values in columns A to H. Global input values are recorded in Columns I to M in row 3 colored in grey.

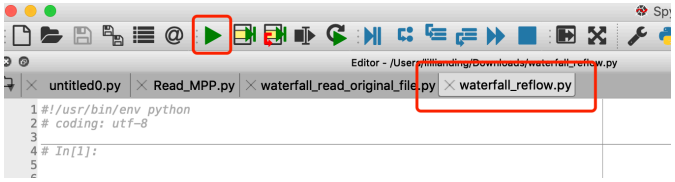


	I	J	K	L	M
1	**Reflow Start Day**	**Reflow Merlion**	**Hold in Integration Bay**	**Saved as**	**Extend Time To:**
2	Type a date that you want to start the reflow. The default reflow start date is the minumum date in column "Revised Build Start" if the cell is blank.	Indicate if you want to reflow Merlion: Blank = Not reflow Merlion; 1 = Reflow Merlion	If leave blank, parts will be held in the integration bays when there is no testing bay available. Type 1 if you want to put the parts aside and make integration bays available.	Type a file name you want to save as. For example: Waterfall AUS 2021-11-01. The default file name will be Reflow Output if you leave the cell blank	Type a date if you want to extend the planning end horizon. The defualt palnning ends on 12/31/2022.
3					1/3/23
4					

Task 4.2 Reflow Steps

Step 1 Save your most current schedule as "Waterfall Input.xlsx". This file might be the output from Task 2 if this is your first time to use the New Python Tool. It might also be the "Reflowed Output 10-30.xlsx" that you saved last time you do the reflow.

Step 2 Go to the Spyder platform and run the "waterfall_reflow.py" code.



Step 3 Open the Output file named "Reflowed Output.xlsx" or any other name you typed in the input file. Copy the data such as "Planned Build Start", "BD Start date", "Ship Revised Date" and "EOP DDD" from MPP file into the new jobs you added.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
1		Slot #	Comments	Customer	Module	Int	Final Test	IFT	BD	Hol,Wkd,Revise	ned Build S	ned Test S	Date Revise	Planned EO	ild Cycle Tir	D Start dat	hip Revise	EOP DDD
2	-1																	
3	-2																	
4	-3																	
5	-4																	
6	-5																	
7	-6	Slot #	Comments	Customer	Module	Int	Final Test	IFT	BD	Hol,Wkd,F	Planned Bu	nned Te	Comp Date	Planned EC	Build Cycle	BD Start da	Ship Revise	EOP DDD
8	0	901383		APPLIED M	NPI Clnr	167	15			2021-02-2	2021-06-2	2021-08-1	2021-10-0	2021-10-0	224	2020-01-0	2021-10-2	2021-10-1
9	1	901383		APPLIED M	NPI Pol	179	15			2021-03-0	2021-06-2	2021-09-0	2021-10-2	2021-10-0	235	2020-01-0	2021-10-2	2021-10-1
10	2	900916		Micron (Re	NPI Pol	33	18			2021-07-0	2021-05-1	2021-08-0	2021-09-0	2021-08-2	61	2020-01-0	2021-08-2	2021-08-2
11	3	901482		INTEL CORF	Std Clnr	13	7			2021-07-0	2021-07-1	2021-07-1	2021-07-2	2021-08-0	23	2020-01-0	2021-08-1	2021-08-1
12	4	901482		INTEL CORF	Std Pol	24	7			2021-07-0	2021-07-1	2021-08-0	2021-08-1	2021-08-0	36	2020-01-0	2021-08-1	2021-08-1
13	5	901621		Taiwan Sen	Std Clnr	13	7			2021-07-1	2021-07-0	2021-07-2	2021-08-0	2021-08-0	23	2020-01-0	2021-08-0	2021-08-0
14	6	901784		Taiwan Sen	Std Clnr	12	7			2021-07-1	2021-07-0	2021-07-2	2021-08-0	2021-08-0	22	2020-01-0	2021-08-0	2021-08-0
15	7	901787		INTEL SEMI	Std Clnr	14	6			2021-07-1	2021-07-1	2021-07-3	2021-08-1	2021-08-0	24	2020-01-0	2021-08-1	2021-08-1
16	8	901787		INTEL SEMI	Std Pol	17	7			2021-07-1	2021-07-1	2021-08-0	2021-08-1	2021-08-0	29	2020-01-0	2021-08-1	2021-08-1
17	9	901085		Taiwan Sen	Std Clnr	12	6			2021-07-2	2021-07-0	2021-08-0	2021-08-1	2021-08-0	21	2020-01-0	2021-08-1	2021-08-0
18	10	901084		Taiwan Sen	Std Clnr	13	6			2021-07-2	2021-07-0	2021-08-0	2021-08-1	2021-08-0	22	2020-01-0	2021-08-1	2021-08-1
399	391	902096		GLOBALFO	Std Clnr	8	7			2022-07-0	2021-12-2	2022-02-1	2022-02-2	2022-02-2	20	2020-01-0		
400	392	1000000			Std Clnr					2021-10-2	nan	2021-11-0	2021-11-1	2021-11-0	19	nan	nan	nan

Step 4 Delete the Miss-Out jobs in the output file. SAVE the excel, then you are all set!

Additional Notes

The New Tool will sort the reflow output by Column "Planned EOP", then the outcome might look different from the current waterfall in excel tool. This is because the "Planned EOP" is not strictly sorted. You can also choose not to sort the rows by adding a # in front of one line 849 in "waterfall_reflow_output.py"

From

```
# In[283]:


blank = [np.nan for i in range(17)]
Int_Total,Int_Clnr,Int_Pol,Test_Total,Test_Clnr,Test_Pol,Kit_Total,Start,NPI_Total = CalBaysCapacity(w1)
r1 = blank + ['# NPI in INT'] + [len(NPI_Total[t]) for t in Time_List_Index]
r2 = blank + ['# Total in INT'] + [len(Int_Total[t]) for t in Time_List_Index]
r3 = blank + ['# Pol in TEST'] + [len(Test_Pol[t]) for t in Time_List_Index]
r4 = blank + ['# Total in Test'] + [len(Test_Total[t]) for t in Time_List_Index]
r6 = blank + ['# Total in Kit'] + [len(Kit_Total[t]) for t in Time_List_Index]
r5 = w1.columns.tolist()
df = pd.DataFrame([r1,r2,r3,r4,r6,r5],index = [-1,-2,-3,-4,-5,-6],columns = r5)
w2 = w1.copy()
w3 = w2.reset_index(drop=True)
w3 = w3.sort_values(by=['Planned EOP'],ascending=True)
w4 = pd.concat([df,w3])
for index, item in w4.iterrows():
    for c in r5:
        if type(w4[c][index]) == datetime.datetime:
            w4[c][index] = str(w4[c][index])[:11]


# In[284]:


w = w4.style.applymap(highlight_cells)#\.set_table_style()


# In[285]:
```

To

```
# In[283]:


blank = [np.nan for i in range(17)]
Int_Total,Int_Clnr,Int_Pol,Test_Total,Test_Clnr,Test_Pol,Kit_Total,Start,NPI_Total = CalBaysCapacity(w1)
r1 = blank + ['# NPI in INT'] + [len(NPI_Total[t]) for t in Time_List_Index]
r2 = blank + ['# Total in INT'] + [len(Int_Total[t]) for t in Time_List_Index]
r3 = blank + ['# Pol in TEST'] + [len(Test_Pol[t]) for t in Time_List_Index]
r4 = blank + ['# Total in Test'] + [len(Test_Total[t]) for t in Time_List_Index]
r6 = blank + ['# Total in Kit'] + [len(Kit_Total[t]) for t in Time_List_Index]
r5 = w1.columns.tolist()
df = pd.DataFrame([r1,r2,r3,r4,r6,r5],index = [-1,-2,-3,-4,-5,-6],columns = r5)
w2 = w1.copy()
w3 = w2.reset_index(drop=True)
#w3 = w3.sort_values(by=['Planned EOP'],ascending=True)
w4 = pd.concat([df,w3])
for index, item in w4.iterrows():
    for c in r5:
        if type(w4[c][index]) == datetime.datetime:
            w4[c][index] = str(w4[c][index])[:11]


# In[284]:
```

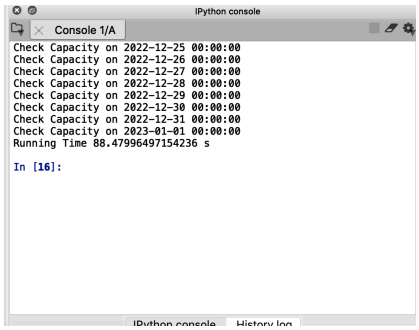
How to tell if a run is finished?

In the Console Task, if a new In [#]: appear, the run is finished.

Running

```
Check Capacity on 2022-12-25 00:00:00
Check Capacity on 2022-12-26 00:00:00
Check Capacity on 2022-12-27 00:00:00
Check Capacity on 2022-12-28 00:00:00
Check Capacity on 2022-12-29 00:00:00
Check Capacity on 2022-12-30 00:00:00
Check Capacity on 2022-12-31 00:00:00
Check Capacity on 2023-01-01 00:00:00
```

Finished



Want to add/modify the holiday?

Go to file “waterfall_reflow.py”, from line 662, in the “Holidays_And_Weekends” variable. If you want to delete 2022-11-25, delete “datetime.datetime(2022,11,25,0,0),” (Remember to delete the “,” in the code). The code will be:

```
# Generate Holiday and Weekends List
starttime = time.time()
Update_List = []
Pre_Time_Num = 17
pd.options.mode.chained_assignment = None  # default='warn'
Holidays_And_Weekends = pd.date_range(start = '2021-10-03', periods=80, freq='w').to_pydatetime().tolist()
Holidays_And_Weekends += [datetime.datetime(2021,11,25,0,0),datetime.datetime(2021,11,26,0,0),
                          datetime.datetime(2021,12,24,0,0),datetime.datetime(2021,12,25,0,0),
                          datetime.datetime(2021,12,31,0,0),datetime.datetime(2022,1,1,0,0),
                          datetime.datetime(2022,1,17,0,0),datetime.datetime(2022,2,21,0,0),
                          datetime.datetime(2022,5,30,0,0),datetime.datetime(2022,7,4,0,0),
                          datetime.datetime(2022,9,4,0,0),
                          datetime.datetime(2022,11,26,0,0),datetime.datetime(2022,12,24,0,0),
                          datetime.datetime(2022,12,25,0,0),datetime.datetime(2022,12,31,0,0),]
# Planning End Time
```

If you want to add a date, say 2023-12-25, append the date to the end of the code as:

```
# Generate Holiday and Weekends List
starttime = time.time()
Update_List = []
Pre_Time_Num = 17
pd.options.mode.chained_assignment = None  # default='warn'
Holidays_And_Weekends = pd.date_range(start = '2021-10-03', periods=80, freq='w').to_pydatetime().tolist()
Holidays_And_Weekends += [datetime.datetime(2021,11,25,0,0),datetime.datetime(2021,11,26,0,0),
                          datetime.datetime(2021,12,24,0,0),datetime.datetime(2021,12,25,0,0),
                          datetime.datetime(2021,12,31,0,0),datetime.datetime(2022,1,1,0,0),
                          datetime.datetime(2022,1,17,0,0),datetime.datetime(2022,2,21,0,0),
                          datetime.datetime(2022,5,30,0,0),datetime.datetime(2022,7,4,0,0),
                          datetime.datetime(2022,9,4,0,0),datetime.datetime(2022,11,25,0,0),
                          datetime.datetime(2022,11,26,0,0),datetime.datetime(2022,12,24,0,0),
                          datetime.datetime(2022,12,25,0,0),datetime.datetime(2022,12,31,0,0),
                          datetime.datetime(2023,12,25,0,0),]
# Planning End Time
```