CASE STUDY – MEGA EXECUTOR

-AMOL BHOYAR

AGENDA

1	Problem Statement
2	Mega Executor : In Brief
3	App Feature List
4	UI Screens and User Instructions
5	I/p and o/p files
6	Innovation Impact

Problem Statement

As part of data architecture team, we have to do a lot of DB validations. These validation types include across different data bases (eg. Snowflake vs Impala), different client data sets (Client1 vs Client2), different environments (UAT vs PROD) etc. Majority of time in these validations goes into logging in to different interfaces, executing queries 1 by 1 and then collating your findings in a shareable format using screenshots, manual exports etc.

Mega Executor automates this manual part by taking care of executions and also makes sure output file is "Ready to Share" formatted with all the information the end user will need in a simple Excel workbook!

The App offers -

Generalised Execution Tool

The app is designed as a generalized tool. There is no limitation set on which database is supported on tool. All users need is the .exe file and the appropriate 32 bit ODBC Driver and they are good to go! (No need of Python installed as well)

Customisable Output

You can customise the result file as per your need by grouping related queries in separate sheets, displaying results one below the other side by side. Your Choice!

Choice of Input

You can provide input excel as a standard input. However, the app is capable of reading the input directly from Google Sheet. Just share your sheet with app's mail id and you are done!

Keep track of the progress

The app is designed to show you the status at each point for individual query (success/fail) and for overall progress. The app will tell you once your result file is formatted and ready!

Mega Executor : In Brief

- Automation of cross database / environment query execution & result logging
- Reduced time consumption

Input File

Sheet Name, Database type, Connection String, Dataset, Query and Comment (optional)

Additional Parameters (optional)

Starting Point , Input Source check, Formatting check

App creates data base connection

App executes query

App generates formatted query results

Database Type, Dataset, Query, Comment, Time stamp, *Execution info & Data OR Error

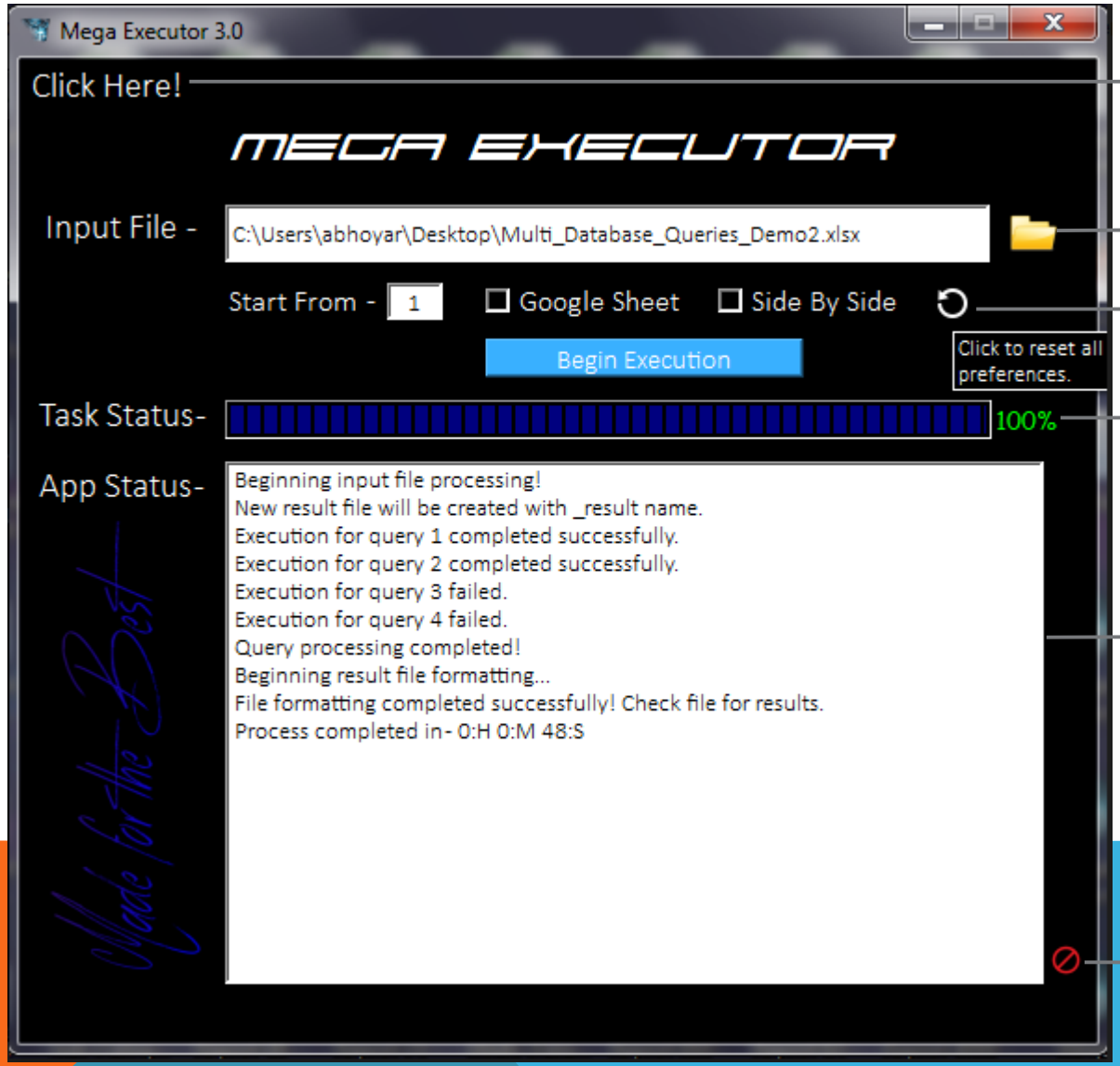
*If query is successful, output will have count of rows returned, execution time (calculated within app) as Execution info and data returned else the error in query

App Feature List

- Any Database platform support , Ex Impala, Snowflake
- Multiple Input sources supported
  - **Excel | GOOGLE SHEETS**
- Output Formatting :
  - **Stacked | Side By Side**
  - **Append | New Sheet**
- Logging :
  - **Query Execution Time | Number of result rows**
- Other user preferences :
  - **"Click Here!" menu**
  - **Custom Start point**
  - **Tolerance for special characters (\n and \t)**
  - **Terminate process**

UI SCREENS AND USER INSTRUCTIONS

UI Main Screen



Click Here Menu

File Explorer Button

Add on Preferences

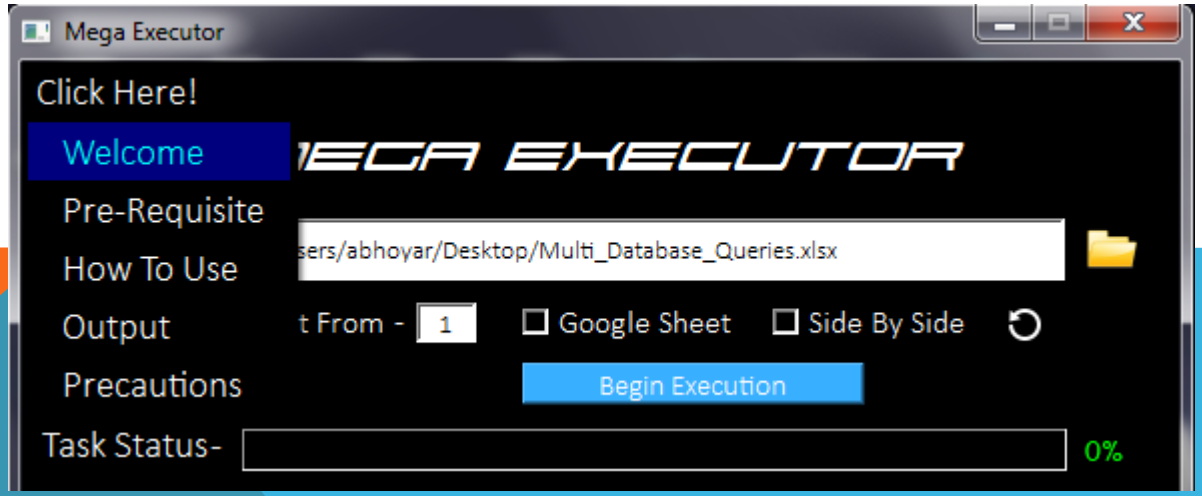
Overall Progress Bar

Individual Query/Process Status

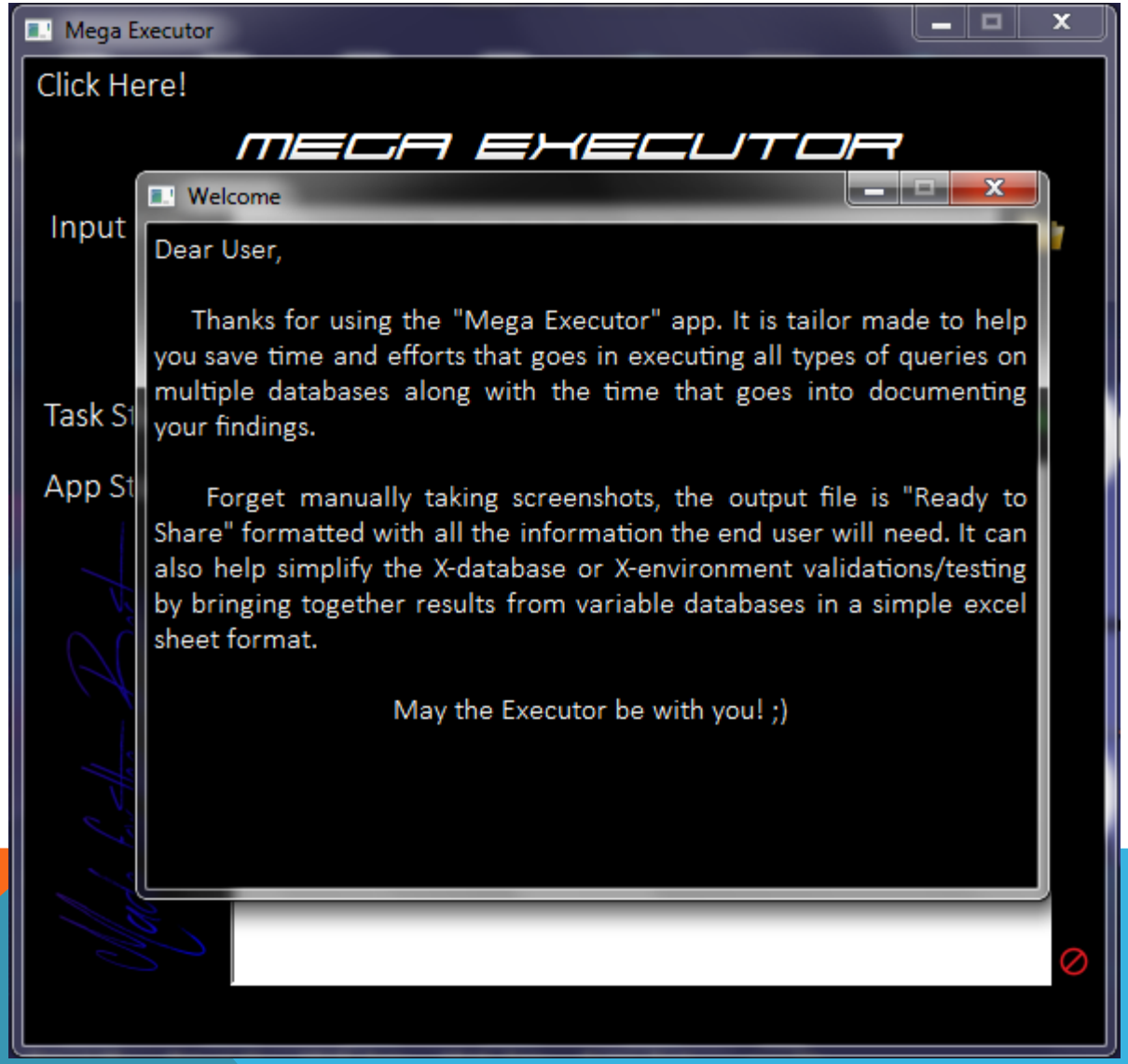
Terminate Process

User Instructions

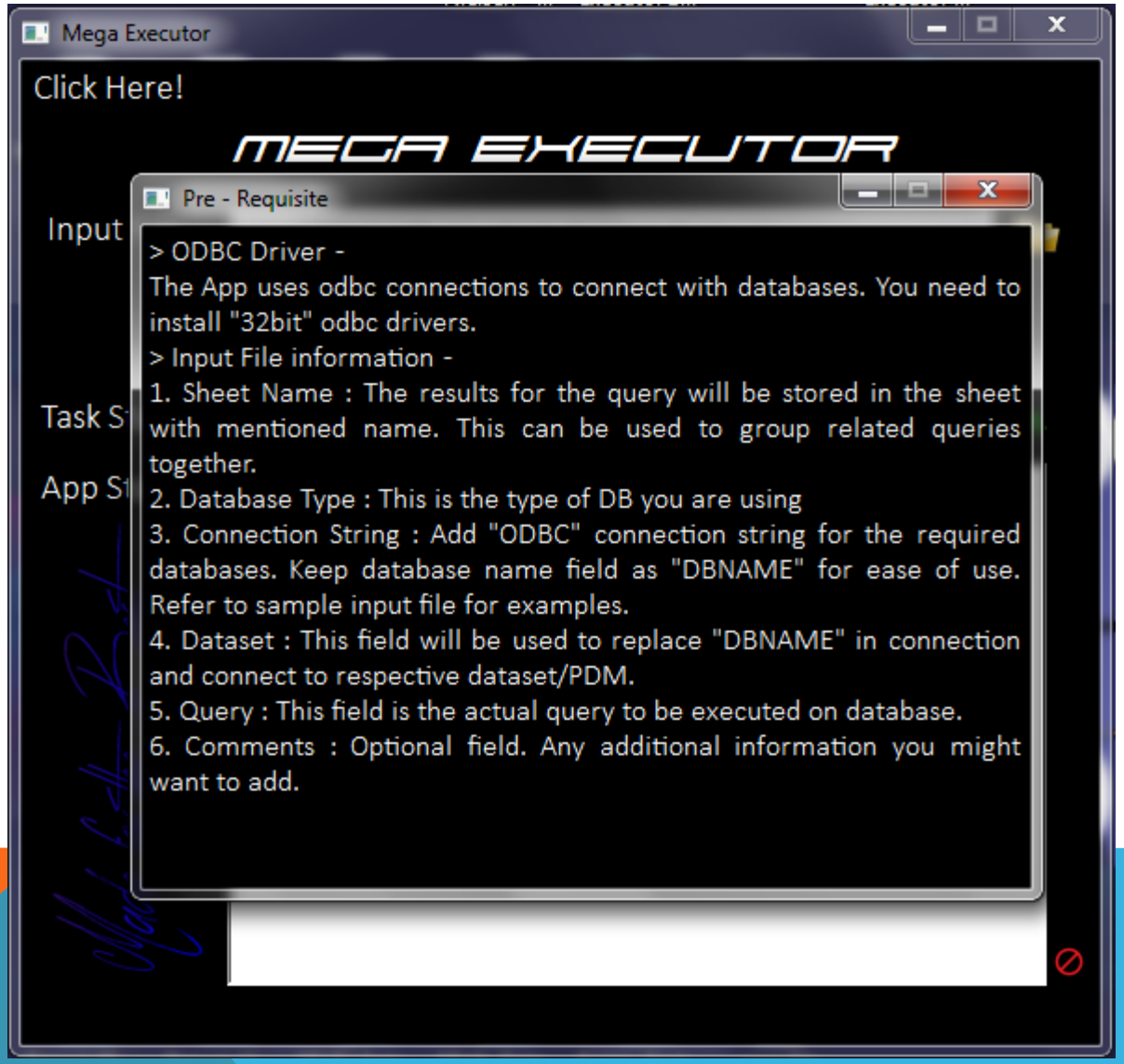
- All the required app details are already present in app, so that user doesn't have to go through any additional documentation
- These screens contain all the details as how to use the app, details on i/p and o/p, precautions to be taken and a welcome note as well!
- They can be accessed on clicking "Click Here" menu on main screen
- No need for updating/maintaining different "how to" type of documents with updating versions. The app is one stop shop for everything related to it !



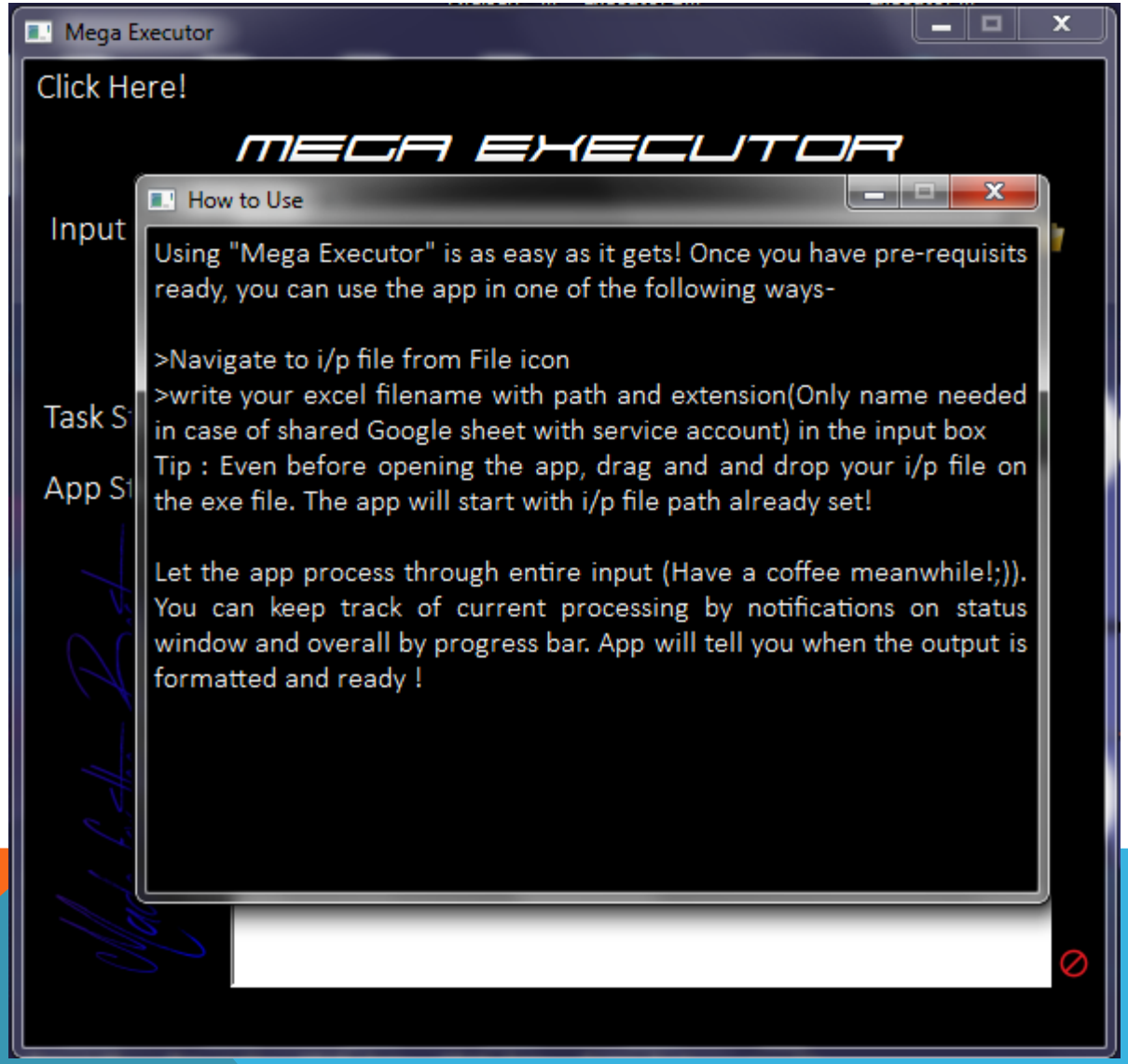
Welcome Note



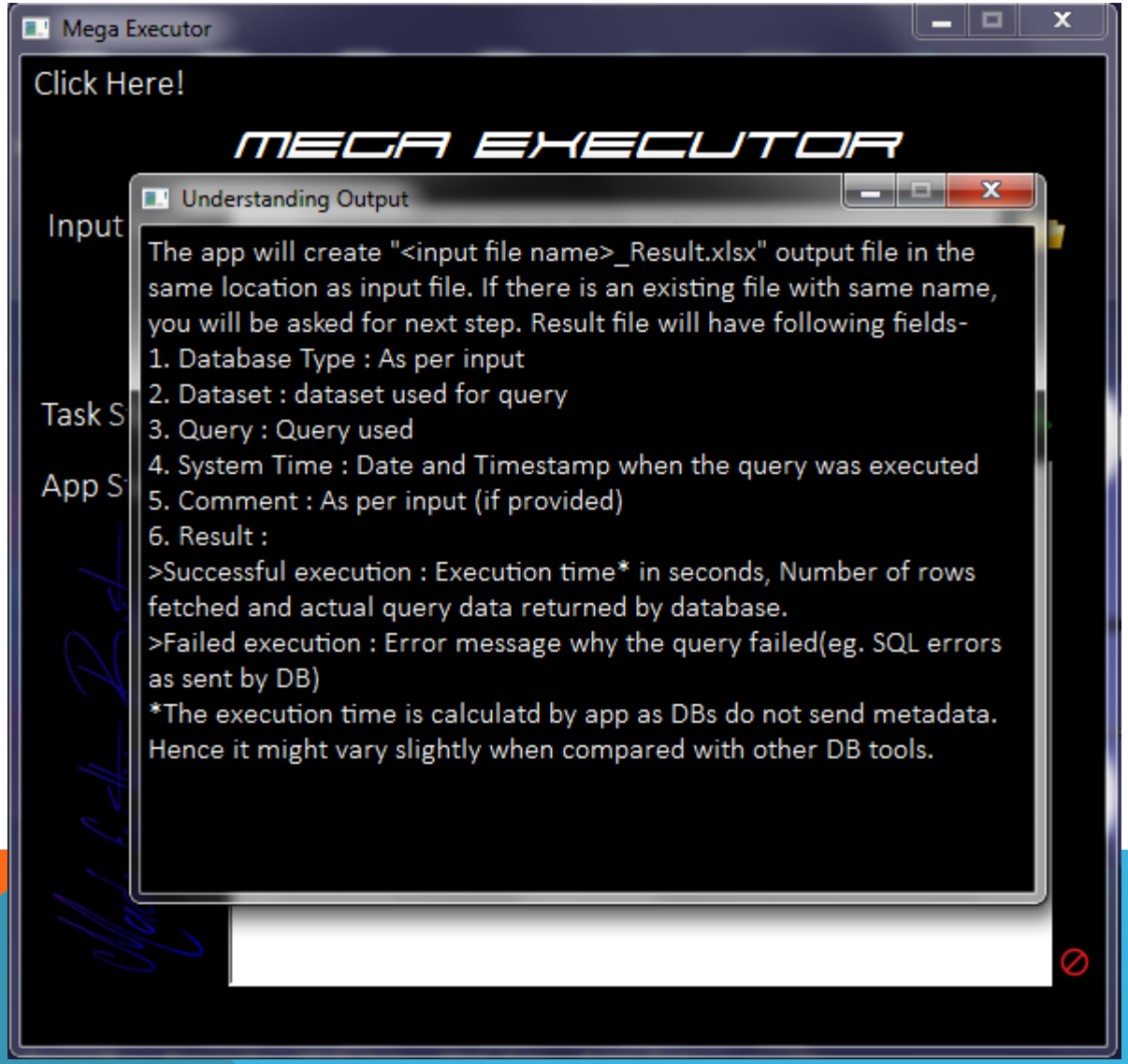
Pre-Requisite



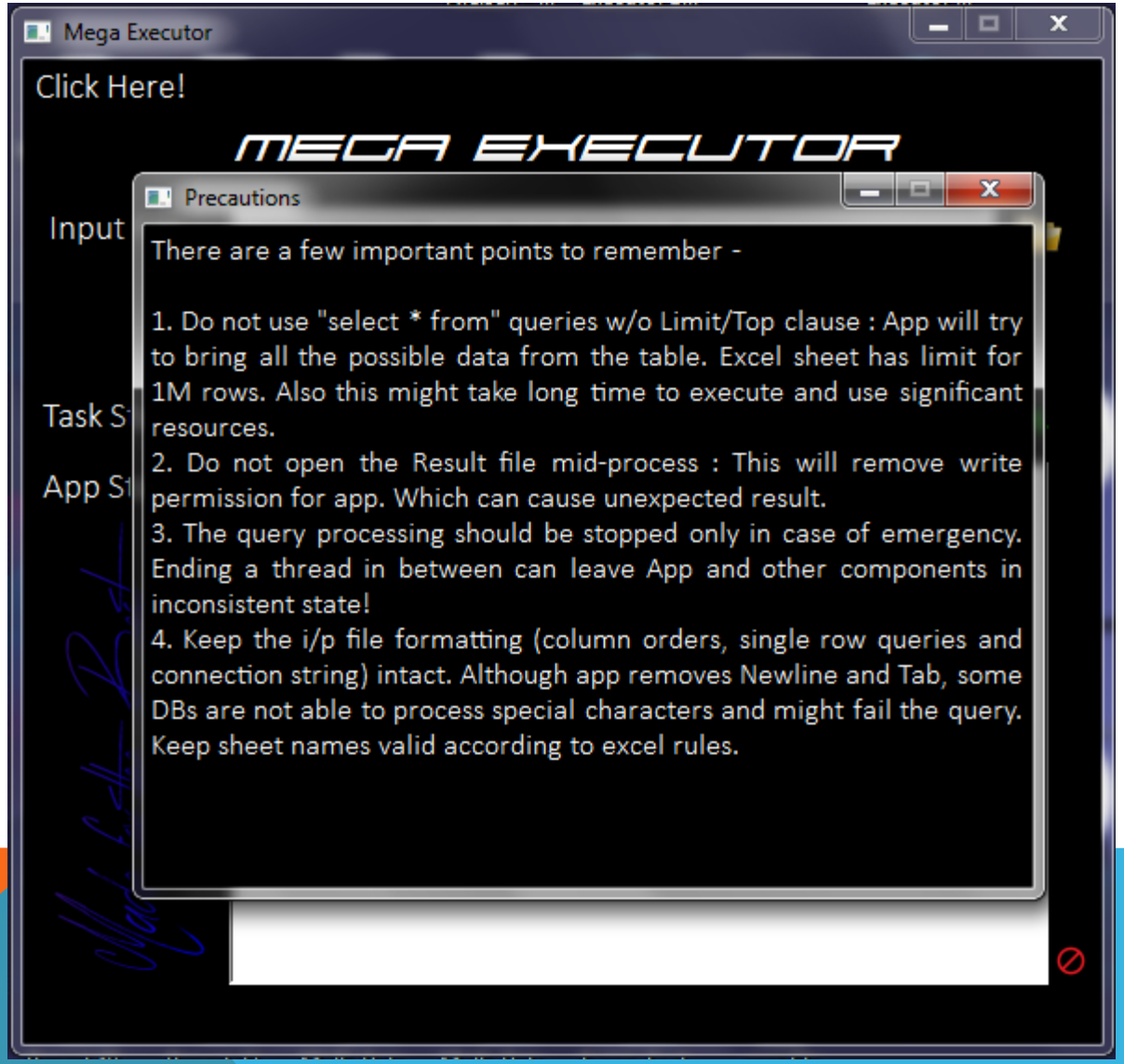
How to Use


Mega Executor
Click Here!
MEGA EXECUTOR
Input
Task S
App St
How to Use
Using "Mega Executor" is as easy as it gets! Once you have pre-requisits ready, you can use the app in one of the following ways-
>Navigate to i/p file from File icon
>write your excel filename with path and extension(Only name needed in case of shared Google sheet with service account) in the input box
Tip : Even before opening the app, drag and and drop your i/p file on the exe file. The app will start with i/p file path already set!
Let the app process through entire input (Have a coffee meanwhile!;)). You can keep track of current processing by notifications on status window and overall by progress bar. App will tell you when the output is formatted and ready !

Output


Mega Executor
Click Here!
MEGA EXECUTOR
Input
Task S
App S
Understanding Output
The app will create "<input file name>_Result.xlsx" output file in the same location as input file. If there is an existing file with same name, you will be asked for next step. Result file will have following fields-
1. Database Type : As per input
2. Dataset : dataset used for query
3. Query : Query used
4. System Time : Date and Timestamp when the query was executed
5. Comment : As per input (if provided)
6. Result :
>Successful execution : Execution time* in seconds, Number of rows fetched and actual query data returned by database.
>Failed execution : Error message why the query failed(eg. SQL errors as sent by DB)
*The execution time is calculatd by app as DBs do not send metadata. Hence it might vary slightly when compared with other DB tools.

Precautions



I/P AND O/P FILES

Input File

- The Input file is simple excel file/Google Sheet containing fields as shown below –

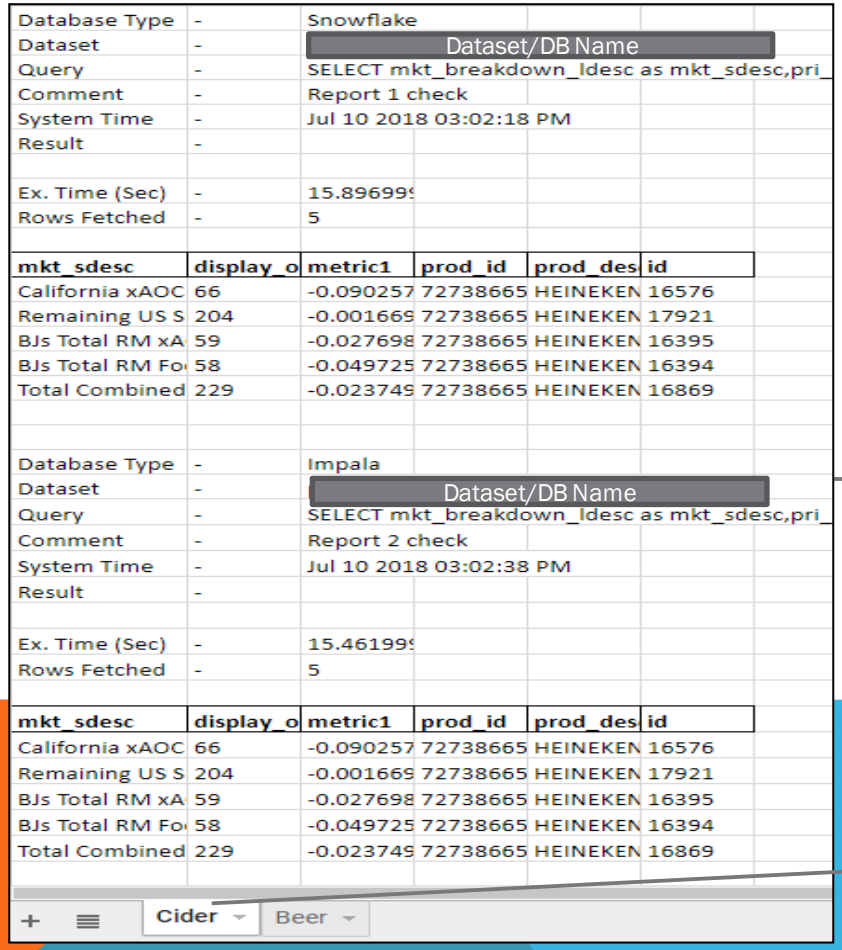
A	B	C	D	E	F	G
Sheet Name	Database Type	Connection String	Datasets	Queries	Comments	
Cider	Impala	Driver=Cloudera ODBC Driver for In		select	Query to check top 5 products	
Beer	Impala	Driver=Cloudera ODBC Driver for In		select	Impala Beer dataset check query	
Beer	Snowflake	Driver=SnowflakeDSIIDriver;server		select		
Cider	Impala	Driver=Cloudera ODBC Driver for In		SELECT	Impala app report 22 Query	
Beer	Impala	Driver=Cloudera ODBC Driver for In		select		
Beer	Snowflake	Driver=SnowflakeDSIIDriver;server		select	Snowflake Beer dataset check query	

Dataset/ DB Name

Queries

Output file (Default) -

- Considering previous file as i/p and no other instructions following is how the output will be generated (file name will be “<i/p file name>_result.xlsx”) –



Database Type	-	Snowflake
Dataset	-	Dataset/DB Name
Query	-	SELECT mkt_breakdown_ldesc as mkt_sdesc,pri_
Comment	-	Report 1 check
System Time	-	Jul 10 2018 03:02:18 PM
Result	-	
Ex. Time (Sec)	-	15.89699
Rows Fetched	-	5

mkt_sdesc	display_o	metric1	prod_id	prod_des	id
California xAOC	66	-0.090257	72738665	HEINEKEN	16576
Remaining US S	204	-0.001669	72738665	HEINEKEN	17921
BJs Total RM xA	59	-0.027698	72738665	HEINEKEN	16395
BJs Total RM Fo	58	-0.049725	72738665	HEINEKEN	16394
Total Combined	229	-0.023749	72738665	HEINEKEN	16869

Database Type	-	Impala
Dataset	-	Dataset/DB Name
Query	-	SELECT mkt_breakdown_ldesc as mkt_sdesc,pri_
Comment	-	Report 2 check
System Time	-	Jul 10 2018 03:02:38 PM
Result	-	
Ex. Time (Sec)	-	15.46199
Rows Fetched	-	5

mkt_sdesc	display_o	metric1	prod_id	prod_des	id
California xAOC	66	-0.090257	72738665	HEINEKEN	16576
Remaining US S	204	-0.001669	72738665	HEINEKEN	17921
BJs Total RM xA	59	-0.027698	72738665	HEINEKEN	16395
BJs Total RM Fo	58	-0.049725	72738665	HEINEKEN	16394
Total Combined	229	-0.023749	72738665	HEINEKEN	16869

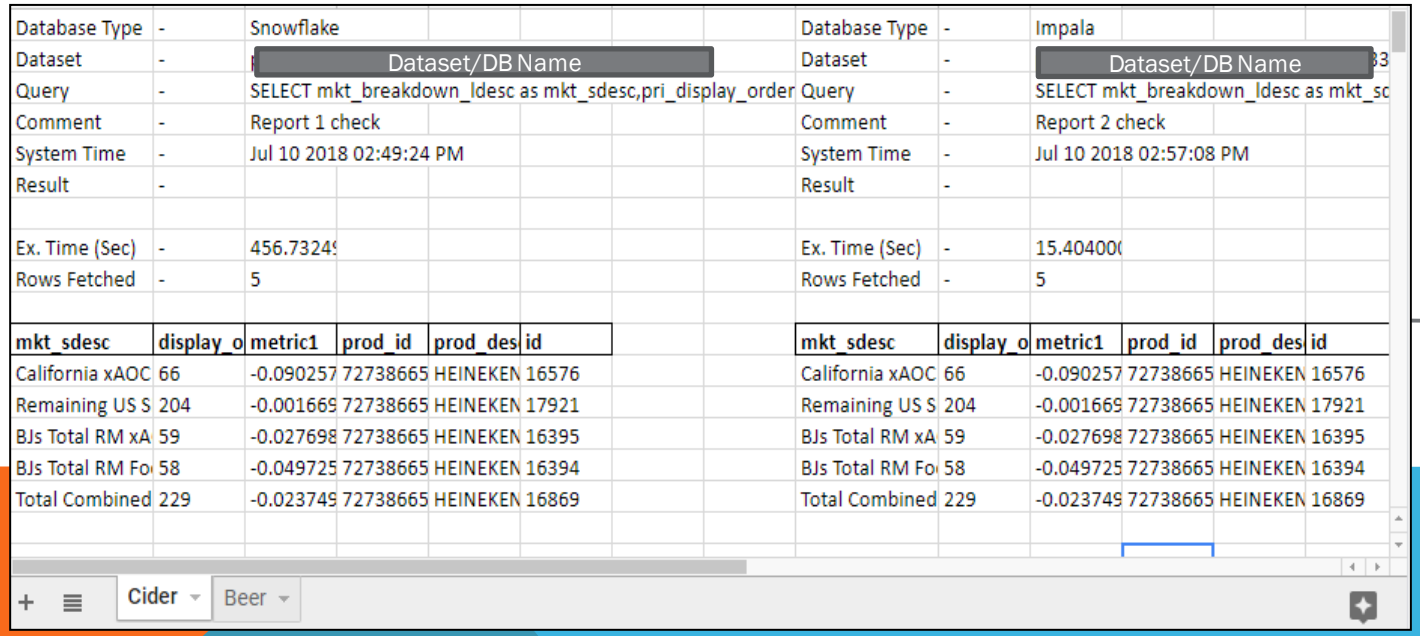
Cider Beer

Queries clubbed by sheet name and displayed one below the other

Sheet Names as provided in input

Output file (Side by side enabled) -

- For same input file if “<i/p file name>_result.xlsx” file already exists, app will ask if user wants to append results to it (It will not overwrite, to prevent accidental data loss)
- Considering user selects not to append results to existing file “<i/p file name>_result_<date-time stamp>.xlsx” file will be generated -



Database Type	-	Snowflake				Database Type	-	Impala			
Dataset	-	[redacted] Dataset/DB Name				Dataset	-	[redacted] Dataset/DB Name			
Query	-	SELECT mkt_breakdown_ldesc as mkt_sdesc,pri_display_order				Query	-	SELECT mkt_breakdown_ldesc as mkt_sc			
Comment	-	Report 1 check				Comment	-	Report 2 check			
System Time	-	Jul 10 2018 02:49:24 PM				System Time	-	Jul 10 2018 02:57:08 PM			
Result	-					Result	-				
Ex. Time (Sec)	-	456.7324[illegible]				Ex. Time (Sec)	-	15.40400[illegible]			
Rows Fetched	-	5				Rows Fetched	-	5			
mkt_sdesc	display_o	metric1	prod_id	prod_des	id	mkt_sdesc	display_o	metric1	prod_id	prod_des	id
California xAOC	66	-0.090257	72738665	HEINEKEN	16576	California xAOC	66	-0.090257	72738665	HEINEKEN	16576
Remaining US S	204	-0.001669	72738665	HEINEKEN	17921	Remaining US S	204	-0.001669	72738665	HEINEKEN	17921
BJs Total RM xA	59	-0.027698	72738665	HEINEKEN	16395	BJs Total RM xA	59	-0.027698	72738665	HEINEKEN	16395
BJs Total RM Fo	58	-0.049725	72738665	HEINEKEN	16394	BJs Total RM Fo	58	-0.049725	72738665	HEINEKEN	16394
Total Combined	229	-0.023749	72738665	HEINEKEN	16869	Total Combined	229	-0.023749	72738665	HEINEKEN	16869

Cider | Beer

Queries clubbed by sheet name and displayed side by side

Innovation Impact

- No manual intervention needed once i/p is provided
- The turnaround time for cross DB data validations came down to 1-2 hrs (consists of actual execution and input query creation time) from a full day
- The turnaround time for cross client dataset validations came down to 20-30 mins (consists of actual execution and input query creation time) from 2-3 hrs
- Used by multiple teams in org

Python Libraries Used

- Pandas – For input/output processing
- Pyodbc – For querying to database
- Openpyxl – For writing output files
- PyQt5 Framework – UI design and Multi-threading
- Gspread – For processing google sheets
- Pyinstaller – For converting code to .exe file

Thank You for reading!

For more details reach out to -

amolbhoyar29@gmail.com