Conference Call Project Documentation (20211210)

Summary of Main Processing Pipeline

	Task	Codes	Input	Output
1	**Download Raw Data**			
1.1	Download Thomson One's Conference Calls [L]	mouse_key_recorder.py automatic_download .py	-	O1: [yyyymmdd-yyyymmdd].pdf O2: [yyyymmdd-yyyymmdd].xls
2	**PDF Processing**			
2.1	Convert Conference Calls from .pdf to .txt [M]	pdftransfer.sh	O1	O3: [yyyymmdd-yyyymmdd].txt
2.2	Split Conference Call .txt files to separate out individual conference calls, and combine with report information from .xls files [M, L]	ParseCCpdf.jl	O3, O2	O4: [yyyymmdd-yyyymmdd].csv
3	**Firm Identification (Firm Name Matching)**			
3.1	Download Compustat datasets [L]	-	-	O5: ciqcompany. sas7bdat O6: ciqcountrygeo. sas7bdat O7: wrds_gvkey. sas7bdat
3.2	Download Hassan dataset [L]	-	-	O8: Hassanfile_raw_ updated20219030.csv
3.3	Process Compustat and Hassan datasets into usable and truncated .csv files. [L]	convert_sas7bdattocsv.py hassanfilecsv_viewable_ truncate.py	O5, O6, O7, O8	O9: ciqcompany.csv O10: ciqcountrygeo.csv O11: wrds_gvkey.csv O12: Hassanfile_raw_ updated20219030 _truncated.csv
3.4	Match titles in conference calls with firm names in Hassan and Compustat datasets, with both exact and	Based on the current flow of work, stage 3 is technically done after stage 4 since it used entryfiles_combined, though the original	O9, O10, O11, O12	O13: CC_List[yyyy].csv O14: CC_List_2020-2021.csv

	fuzzy matching [M, L]	codes were written as stage 4 after stage 3.		
1	Identify keywords for the whole CC data.	CC_identify_keywords.py	CC Data on the server at “CriCount/group{X}” where X = 1, 2, 3 ..50	XXX1: CC Data with relevant keywords at “/project/kh_mercury_1/CriCount/Full_Identified_Keywords/group{X}” where X = 1, 2, 3 ..50
2	Concatenating all these files into a single dataset.	concatenateOutputs.py	XXX1	XXX2: Full_Master_Keywords.csv
3	Filter only those entries which were not collected during the first run	optimizedGetNewEntries.py	XXX2	XXX3: Full_New_Not_Done.csv
4	Filter based on a more exact keyword identification algorithm (rather than just checking in, doing a holistic check by looking at the spaces around the keyword)	getCorrect.py	XXX3	XXX4: Amended_Correct_No_IR.csv
5	Filter based on the presence of a percentage (the words percent, per cent, percentage, %) and then order based on the sorting rule provided by Kilian.	Ordering And Filtering.ipynb	XXX4	XXX5: Filtered_Ordered_Amended_Correct_No_IR.csv
6	Perform the fuzzy matching between the Hassan/Compus	CCFuzzyMatch.ipynb	XXX5, Hassanfile_raw_updated2019030_viewable.csv, ????: ciqcompany_mergedwithgvkeyandcountry.csv	XXX6: manual_full_updated_conf_calls.xlsx

	tat and the CC datasets.			
4	**Keyword Identification**			
4.1	Make a folder structure with x groups (default: x = 50) [M, L]	mkdir.py dividefilesequallyintofolders.py	-	-
4.2	Make a list of keywords and template entry file [L]	-	-	O15: keyterms.txt O16: Entry mask.xlsx
4.3	Identify keywords in each conference call [M, L]	keyword_ident_1.py keyword_ident_1.sh	O15, O4	O17: FR5.csv
4.4	Extract all paragraphs from each conference call that contains a specific keyword [M, L]	keyword_ident_2.py keyword_ident_2.sh	O17	O18: TotalCircnew.xlsx
4.5	Cleans the identified matches and merges with gvkey dataset [L]	mergeclean.do	O18, O14	O19: cric1_newtotal.xlsx
4.6	Make a paragraph record file that splits the number of entries into groups of 500 [L]	make_paragraphrecord.py	O19	O20: paragraphrecord.xlsx
4.7	Bold the keywords and separate file into "entryfiles", each containing 500 entries. [M, L]	makeentryfiles.py makeentryfiles.sh	O19, O20, O15, O16	O21: [i].xlsx
4.8	Combine entry files [L]	combine_entryfilesjason.py	O21	O22: entryfiles_combined.xlsx

		combine_entryfilessixun.py combine_sixunandjasonentryfiles.py		
5	**Get Front Page Descriptions**			
5.1	Extract front page descriptions from conference calls [M, L]	extractdescriptioninfrontpage.py extractdescriptioninfrontpage.sh copyfiles.py copyfiles.sh	O21, O2, O3	O23: [yyyymmdd-yyyymmdd]_withfrontpagedesc.xlsx
5.2	Manually check through error cases and correct accordingly [L]	-	O23	O24: [yyyymmdd-yyyymmdd]_withfrontpagedesc.xlsx
5.3	Combine xls files [L]	combine_xlsfiles_withdescription.py	O24	O25: xlscombined_withfrontpagedescription.xlsx
5.4	Match and add front page descriptions to combined entry files [L]	-	O25, O22	O26: entryfiles_combined.xlsx (updated)

* M = Mercury, L = Local. [M] / [L] means this stage can be run on Mercury / locally (on your Booth Windows laptop) respectively. [M, L] means this stage can be run on both Mercury and your local laptop, where Mercury is preferred for large datasets and local is preferred for initial testing, debugging and small datasets.

1. Download Raw Data

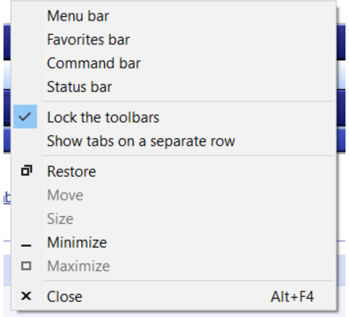
1.1 Download Thomson One's Conference Calls [L]

The goal is to download conference calls from Thomson One. This includes both the pdf files containing the actual calls, and xls files containing identifiers.

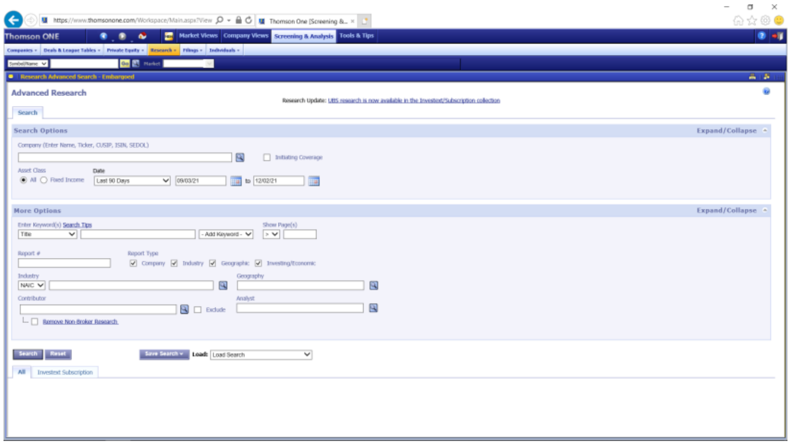
The main obstacles are that: (1) each page will only show 50 conference calls, (2) a maximum of 2,000 conference calls will be presented for every search, (3) web drivers are prohibited / web scraping doesn't work. The current solution is thus to write a python file that does auto-clicking, and this is automatic_download.py.

Before running the code:

- Open Internet Explorer (no other browers are allowed) to access Thomson One (proxy.uchicago.edu/login/thomsonone).
- Ensure that your browser settings are configured to enable the code to work. The goal is to hide away extraneous elements on the screen, so that no scrolling is needed to be able to click on all co-ordinates.
- The current code works for the Booth laptop, Lenono Thinkpad X1 Extreme (Windows 10) that is not connected to a HDMI screen. If it is connected to another screen, the current saved co-ordinates will likely be off. The screen resolution details (https://whatsmyscreensize.com/):
    - Screen Resolution
    - Width: 1920
    - Height: 1080
    - Device Pixel Ratio: 1.25
    - Display Dimensions (width x height): 16.0" x 9.0"
    - Screen Diagonal: 18.4" Screen
- The settings that worked for this set-up are:
    - Windows Taskbar: "Automatically hide the taskbar in desktop mode" is turned on
    - Internet Explorer settings: 125% zoom + the following



    - So that the screen looks like this:



Advanced Research

Research Update: UBS research is now available in the Investext/Subscription collection

Search Options

Company (Enter Name, Ticker, CUSIP, ISIN, SEDOL)

□ Initiating Coverage

Asset Class ● All ○ Fixed Income

Date Last 90 Days 09/03/21 to 12/02/21

More Options

Enter Keyword(s) Title - Add Keyword -

Show Page(s) >

Report #

Report Type ☑ Company ☑ Industry ☑ Geographic ☑ Investing/Economic

Industry NAIC

Geography

Contributor □ Exclude

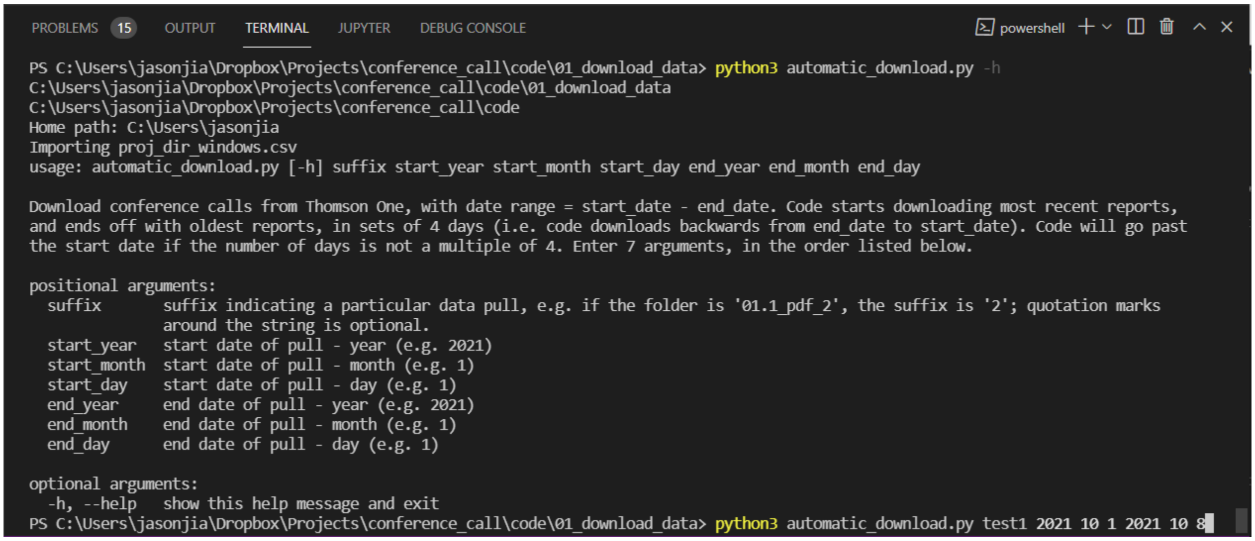
Analyst

Load: Load Search

- Create the necessary folders to store your output
  - Store the folders in: conference_call\output\01_download_cc
  - The root names are 01.1_pdf and 01.1_xls.
  - The suffix is used to separate different data pulls. Decide on a suffix.
  - Then create the two folders: "01.1_pdf_[suffix]" and "01.1_xls_[suffix]".

To run the code:

- Open a terminal.
- Cd to "conference_call\code\01_download_data".
- Run automatic_download.py, specifying the suffix, the start date (year, month, day) and the end date year, month, day):
  - E.g. python3 automatic_download.py test1 2021 10 1 2021 10 8



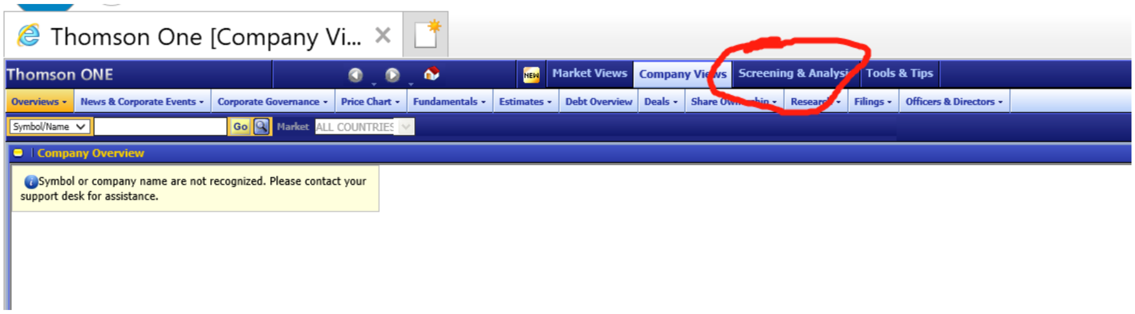
```
PS C:\Users\jasonjia\Dropbox\Projects\conference_call\code\01_download_data> python3 automatic_download.py -h
C:\Users\jasonjia\Dropbox\Projects\conference_call\code\01_download_data
C:\Users\jasonjia\Dropbox\Projects\conference_call\code
Home path: C:\Users\jasonjia
Importing proj_dir_windows.csv
usage: automatic_download.py [-h] suffix start_year start_month start_day end_year end_month end_day

Download conference calls from Thomson One, with date range = start_date - end_date. Code starts downloading most recent reports,
and ends off with oldest reports, in sets of 4 days (i.e. code downloads backwards from end_date to start_date). Code will go past
the start date if the number of days is not a multiple of 4. Enter 7 arguments, in the order listed below.

positional arguments:
  suffix       suffix indicating a particular data pull, e.g. if the folder is '01.1_pdf_2', the suffix is '2'; quotation marks
               around the string is optional.
  start_year   start date of pull - year (e.g. 2021)
  start_month  start date of pull - month (e.g. 1)
  start_day    start date of pull - day (e.g. 1)
  end_year     end date of pull - year (e.g. 2021)
  end_month    end date of pull - month (e.g. 1)
  end_day      end date of pull - day (e.g. 1)

optional arguments:
  -h, --help   show this help message and exit
PS C:\Users\jasonjia\Dropbox\Projects\conference_call\code\01_download_data> python3 automatic_download.py test1 2021 10 1 2021 10 8
```

When running the code:

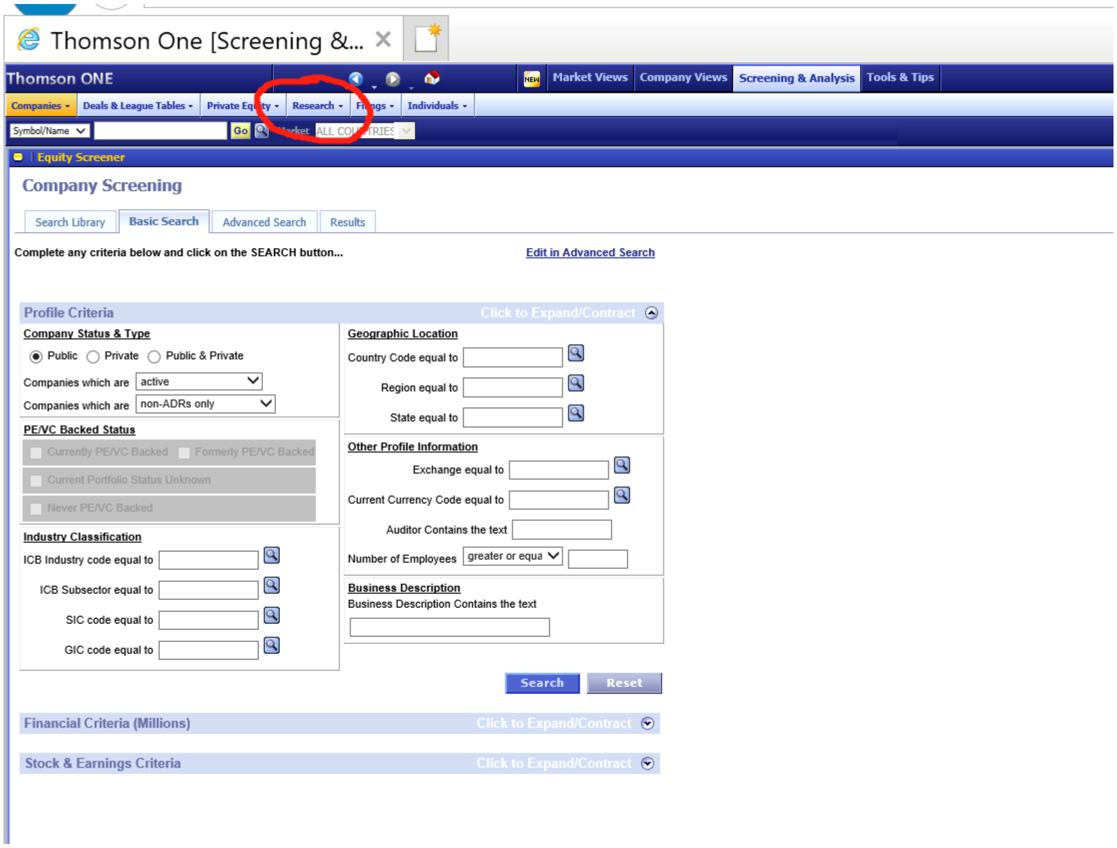
The main loop of the code tries to do the following:

<u>(a) Enter search details</u>

- Click on Screening & Analysis -> Research



Symbol or company name are not recognized. Please contact your support desk for assistance.



Complete any criteria below and click on the SEARCH button...

Profile Criteria

Company Status & Type

● Public ○ Private ○ Public & Private

Companies which are: active

Companies which are: non-ADRs only

PE/VC Backed Status

□ Currently PE/VC Backed □ Formerly PE/VC Backed

□ Current Portfolio Status Unknown

□ Never PE/VC Backed

Industry Classification

ICB Industry code equal to

ICB Subsector equal to

SIC code equal to

GIC code equal to

Geographic Location

Country Code equal to

Region equal to

State equal to

Other Profile Information

Exchange equal to

Current Currency Code equal to

Auditor Contains the text

Number of Employees: greater or equa

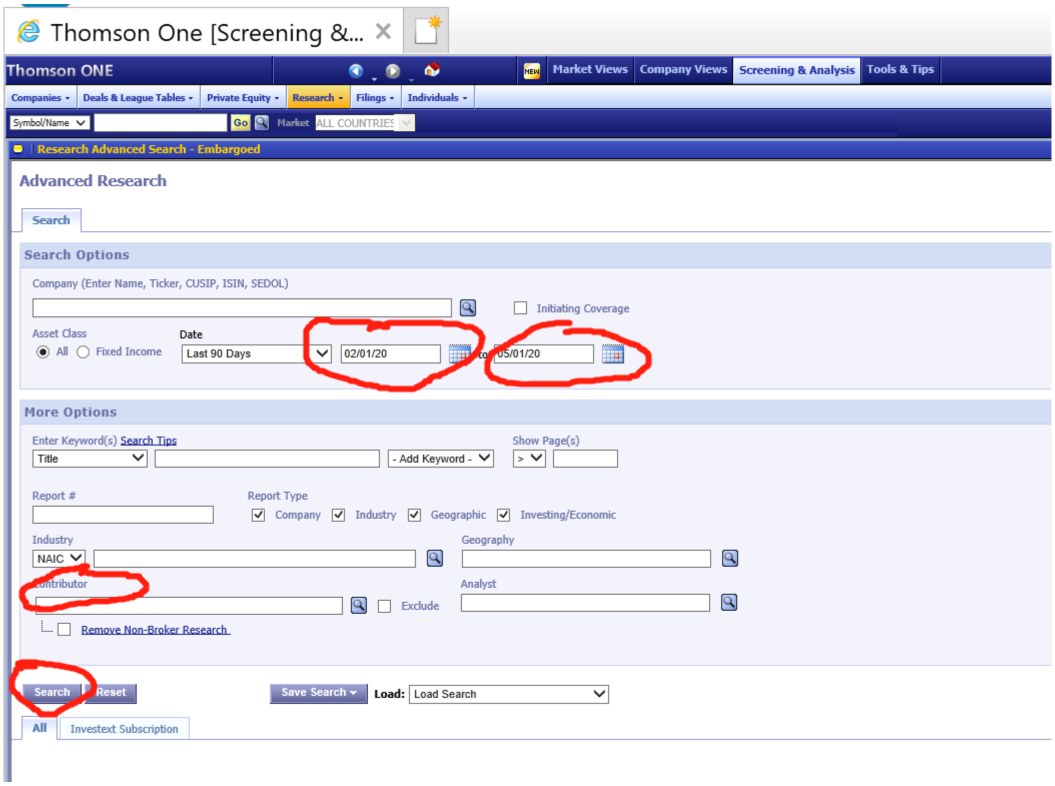
Business Description

Business Description Contains the text

Financial Criteria (Millions)

Stock & Earnings Criteria

- Click on the Contributor field
- Type Streetevents in the Contributor field
- Click “Refinitiv Streetevents” in the drop-down list
    - Used to be “Thomson Reuter Streetevents” and may be something else in the future. More importantly, there should only be one option containing Streetevents, and it should be the correct option.
- Click on the start date field
- Type the start date
- Click on the end date field
- Type the end date
    - A 4-day time interval is used to ensure that each search gives no more than 2,000 calls.
- Press enter, which is equivalent to clicking the search button.



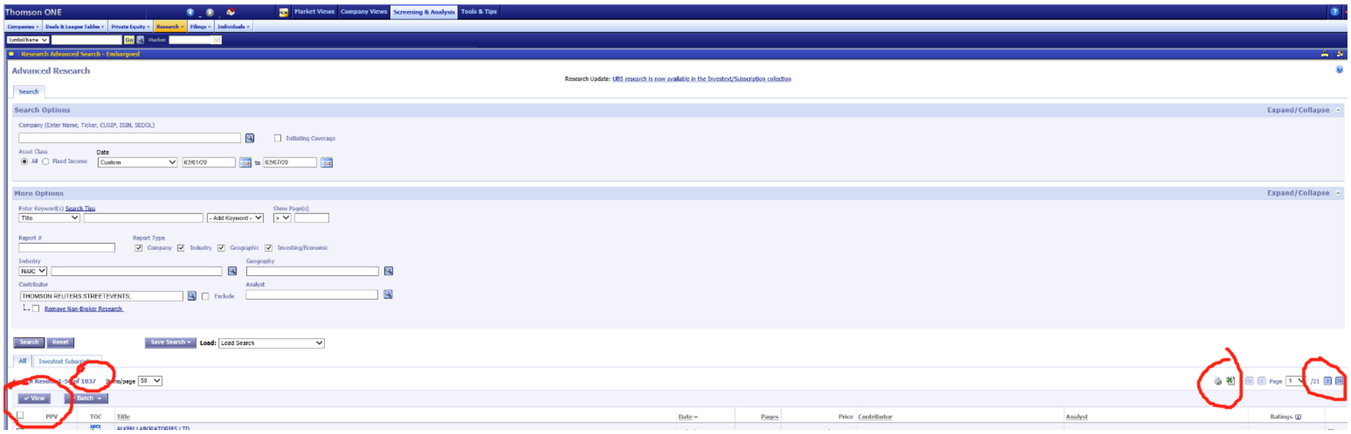
Asset Class: ● All ○ Fixed Income

Date: Last 90 Days 02/01/20 to 05/01/20

Report Type: ☑ Company ☑ Industry ☑ Geographic ☑ Investing/Economic

(b) Download the pdfs and xls files

- Copy the number of calls given by the search, save the number into a separate data set, and calculate the number of pages.
- Select all calls in one page
- Download the pdf file with all calls in that page into 01.1/pdf_[suffix]
- Download the xls file which contains information of each call into 01.1/xls_[suffix].
- To ensure that pdf and xls files are in pairs, check the existence of files in the folders.
- If there are some errors, the code will restart the process (by typing in the login websites, and automatically finishing the access steps.
- Uncheck all calls, click next page, repeat (b). If this is the last page of the date period, go back to (a) with date moving backwards.



Research Update: UBS research is now available in the Investext/Subscription collection

Date: Custom 02/01/20 to 02/07/20

Contributor: THOMSON REUTERS STREETEVENTS,

Search Results 1-50 of 1037 Items/page 50

Page 1 /21

ALKEM LABORATORIES LTD

Errors Handling

Errors can happen for many reasons, e.g. (1) automatic log off, (2) sudden network error, (3) system authentication error (log in failure), (4) change of file orders in the subsequent login, and (5) broken or corrupted files, and (6) unsuccessful download of files. The point is that some “manual coaxing” is necessary to help the code run smoothly from start to end.

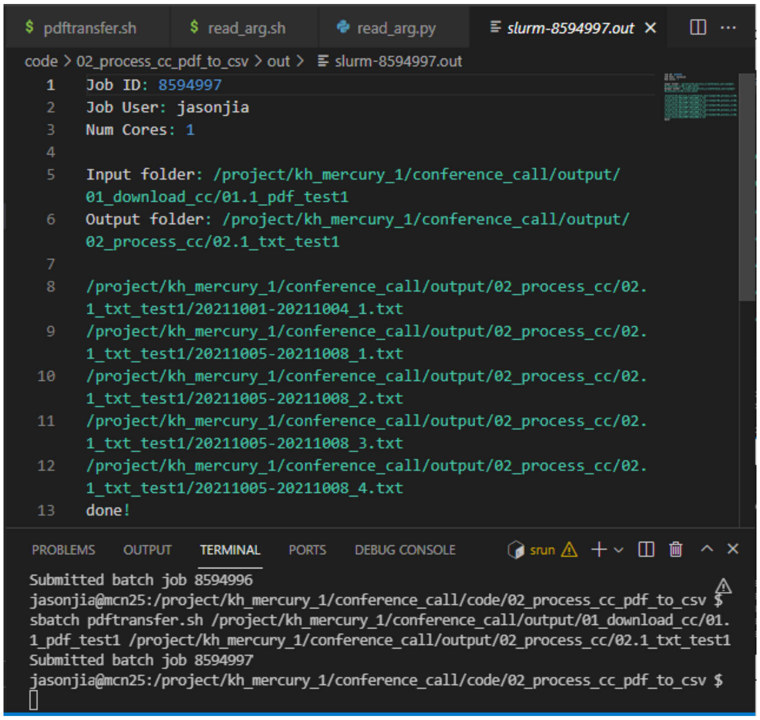
It is possible to try to account for all the errors, but from practical experience, the benefit of fewer errors, relative to the cost of more complex and harder-to-maintain code, diminishes quickly. We thus choose the following approach:

- If files fail to download, there is a time interval where the code will pause, for you to manually click to the correct state. Then, the code will try to save the file again.
- For all other errors, stop the code, get back into a workable state, and then rerun the code with a now truncated time frame.

2. PDF Processing

2.1 Convert Conference Calls from .pdf to .txt [M]

- Copy over the pdf and xls files to Mercury
- Run pdftransfer.sh on Mercury, specifying the full location of the input folder and the output folder
    - cd “conference_call/code/02_process_cc_pdf_to_csv”
    - sbatch pdftransfer.sh [input folder] [output folder]
- The important command is pdftotext, a Linux command that converts pdf to txt files. It also adds page and paragraph delimiters, which helps split the txt files by conference call later on.



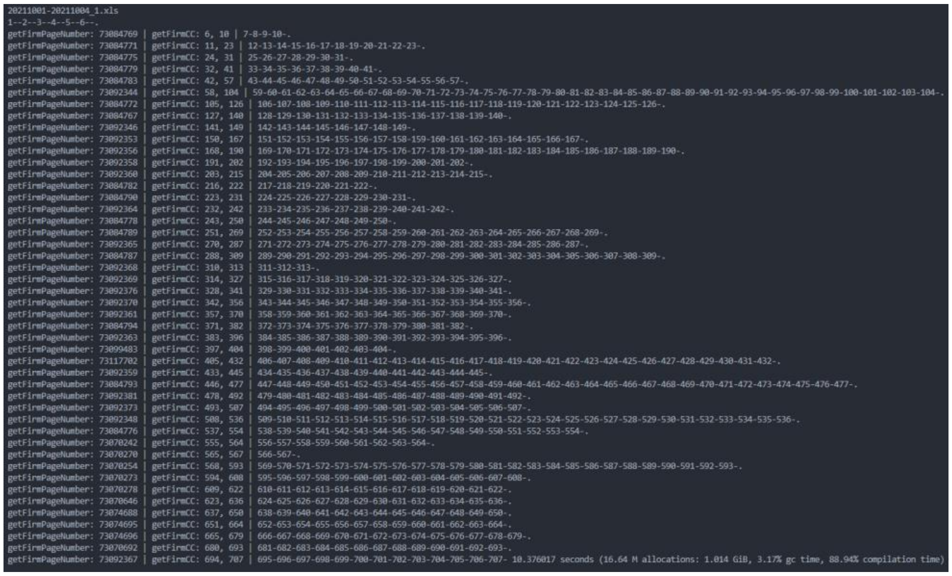
```
Job ID: 8594997
Job User: jasonjia
Num Cores: 1

Input folder: /project/kh_mercury_1/conference_call/output/
01_download_cc/01.1_pdf_test1
Output folder: /project/kh_mercury_1/conference_call/output/
02_process_cc/02.1_txt_test1

/project/kh_mercury_1/conference_call/output/02_process_cc/02.
1_txt_test1/20211001-20211004_1.txt
/project/kh_mercury_1/conference_call/output/02_process_cc/02.
1_txt_test1/20211005-20211008_1.txt
/project/kh_mercury_1/conference_call/output/02_process_cc/02.
1_txt_test1/20211005-20211008_2.txt
/project/kh_mercury_1/conference_call/output/02_process_cc/02.
1_txt_test1/20211005-20211008_3.txt
/project/kh_mercury_1/conference_call/output/02_process_cc/02.
1_txt_test1/20211005-20211008_4.txt
done!
```

```
Submitted batch job 8594996
jasonjia@mcn25:/project/kh_mercury_1/conference_call/code/02_process_cc_pdf_to_csv $
sbatch pdftransfer.sh /project/kh_mercury_1/conference_call/output/01_download_cc/01.
1_pdf_test1 /project/kh_mercury_1/conference_call/output/02_process_cc/02.1_txt_test1
Submitted batch job 8594997
jasonjia@mcn25:/project/kh_mercury_1/conference_call/code/02_process_cc_pdf_to_csv $
```

- The txt files will be in the output folder

2.2 Split Conference Call .txt files to separate out individual conference calls, and combine with report information from .xls files [M, L]

- Copy over the txt files to Dropbox
- Run ParseCCpdf.jl
- Print messages have been added so you can see how the pdf files are processed. An example is given below:



```
20211001-20211004_1.xls
1--2--3--4--5--6--.
getFirmPageNumber: 73084769 | getFirmCC: 6, 10 | 7-8-9-10-.
getFirmPageNumber: 73084771 | getFirmCC: 11, 23 | 12-13-14-15-16-17-18-19-20-21-22-23-.
getFirmPageNumber: 73084775 | getFirmCC: 24, 31 | 25-26-27-28-29-30-31-.
getFirmPageNumber: 73084779 | getFirmCC: 32, 41 | 33-34-35-36-37-38-39-40-41-.
getFirmPageNumber: 73084783 | getFirmCC: 42, 57 | 43-44-45-46-47-48-49-50-51-52-53-54-55-56-57-.
getFirmPageNumber: 73092344 | getFirmCC: 58, 104 | 59-60-61-62-63-64-65-66-67-68-69-70-71-72-73-74-75-76-77-78-79-80-81-82-83-84-85-86-87-88-89-90-91-92-93-94-95-96-97-98-99-100-101-102-103-104-.
getFirmPageNumber: 73084772 | getFirmCC: 105, 126 | 106-107-108-109-110-111-112-113-114-115-116-117-118-119-120-121-122-123-124-125-126-.
getFirmPageNumber: 73084767 | getFirmCC: 127, 140 | 128-129-130-131-132-133-134-135-136-137-138-139-140-.
getFirmPageNumber: 73092346 | getFirmCC: 141, 149 | 142-143-144-145-146-147-148-149-.
getFirmPageNumber: 73092353 | getFirmCC: 150, 167 | 151-152-153-154-155-156-157-158-159-160-161-162-163-164-165-166-167-.
getFirmPageNumber: 73092356 | getFirmCC: 168, 190 | 169-170-171-172-173-174-175-176-177-178-179-180-181-182-183-184-185-186-187-188-189-190-.
getFirmPageNumber: 73092358 | getFirmCC: 191, 202 | 192-193-194-195-196-197-198-199-200-201-202-.
getFirmPageNumber: 73092360 | getFirmCC: 203, 215 | 204-205-206-207-208-209-210-211-212-213-214-215-.
getFirmPageNumber: 73084782 | getFirmCC: 216, 222 | 217-218-219-220-221-222-.
getFirmPageNumber: 73084790 | getFirmCC: 223, 231 | 224-225-226-227-228-229-230-231-.
getFirmPageNumber: 73092364 | getFirmCC: 232, 242 | 233-234-235-236-237-238-239-240-241-242-.
getFirmPageNumber: 73084778 | getFirmCC: 243, 250 | 244-245-246-247-248-249-250-.
getFirmPageNumber: 73084789 | getFirmCC: 251, 269 | 252-253-254-255-256-257-258-259-260-261-262-263-264-265-266-267-268-269-.
getFirmPageNumber: 73092365 | getFirmCC: 270, 287 | 271-272-273-274-275-276-277-278-279-280-281-282-283-284-285-286-287-.
getFirmPageNumber: 73084787 | getFirmCC: 288, 309 | 289-290-291-292-293-294-295-296-297-298-299-300-301-302-303-304-305-306-307-308-309-.
getFirmPageNumber: 73092368 | getFirmCC: 310, 313 | 311-312-313-.
getFirmPageNumber: 73092369 | getFirmCC: 314, 327 | 315-316-317-318-319-320-321-322-323-324-325-326-327-.
getFirmPageNumber: 73092376 | getFirmCC: 328, 341 | 329-330-331-332-333-334-335-336-337-338-339-340-341-.
getFirmPageNumber: 73092370 | getFirmCC: 342, 356 | 343-344-345-346-347-348-349-350-351-352-353-354-355-356-.
getFirmPageNumber: 73092361 | getFirmCC: 357, 370 | 358-359-360-361-362-363-364-365-366-367-368-369-370-.
getFirmPageNumber: 73084794 | getFirmCC: 371, 382 | 372-373-374-375-376-377-378-379-380-381-382-.
getFirmPageNumber: 73092363 | getFirmCC: 383, 396 | 384-385-386-387-388-389-390-391-392-393-394-395-396-.
getFirmPageNumber: 73099483 | getFirmCC: 397, 404 | 398-399-400-401-402-403-404-.
getFirmPageNumber: 73117702 | getFirmCC: 405, 432 | 406-407-408-409-410-411-412-413-414-415-416-417-418-419-420-421-422-423-424-425-426-427-428-429-430-431-432-.
getFirmPageNumber: 73092359 | getFirmCC: 433, 445 | 434-435-436-437-438-439-440-441-442-443-444-445-.
getFirmPageNumber: 73084793 | getFirmCC: 446, 477 | 447-448-449-450-451-452-453-454-455-456-457-458-459-460-461-462-463-464-465-466-467-468-469-470-471-472-473-474-475-476-477-.
getFirmPageNumber: 73092381 | getFirmCC: 478, 492 | 479-480-481-482-483-484-485-486-487-488-489-490-491-492-.
getFirmPageNumber: 73092373 | getFirmCC: 493, 507 | 494-495-496-497-498-499-500-501-502-503-504-505-506-507-.
getFirmPageNumber: 73092348 | getFirmCC: 508, 536 | 509-510-511-512-513-514-515-516-517-518-519-520-521-522-523-524-525-526-527-528-529-530-531-532-533-534-535-536-.
getFirmPageNumber: 73084776 | getFirmCC: 537, 554 | 538-539-540-541-542-543-544-545-546-547-548-549-550-551-552-553-554-.
getFirmPageNumber: 73070242 | getFirmCC: 555, 564 | 556-557-558-559-560-561-562-563-564-.
getFirmPageNumber: 73070270 | getFirmCC: 565, 567 | 566-567-.
getFirmPageNumber: 73070254 | getFirmCC: 568, 593 | 569-570-571-572-573-574-575-576-577-578-579-580-581-582-583-584-585-586-587-588-589-590-591-592-593-.
getFirmPageNumber: 73070273 | getFirmCC: 594, 608 | 595-596-597-598-599-600-601-602-603-604-605-606-607-608-.
getFirmPageNumber: 73070278 | getFirmCC: 609, 622 | 610-611-612-613-614-615-616-617-618-619-620-621-622-.
getFirmPageNumber: 73070646 | getFirmCC: 623, 636 | 624-625-626-627-628-629-630-631-632-633-634-635-636-.
getFirmPageNumber: 73074688 | getFirmCC: 637, 650 | 638-639-640-641-642-643-644-645-646-647-648-649-650-.
getFirmPageNumber: 73074695 | getFirmCC: 651, 664 | 652-653-654-655-656-657-658-659-660-661-662-663-664-.
getFirmPageNumber: 73074696 | getFirmCC: 665, 679 | 666-667-668-669-670-671-672-673-674-675-676-677-678-679-.
getFirmPageNumber: 73070692 | getFirmCC: 680, 693 | 681-682-683-684-685-686-687-688-689-690-691-692-693-.
getFirmPageNumber: 73092367 | getFirmCC: 694, 707 | 695-696-697-698-699-700-701-702-703-704-705-706-707- 10.376017 seconds (16.64 M allocations: 1.014 GiB, 3.17% gc time, 88.94% compilation time)
```

- The output will be csv files that can be thought of as .xls files combined with the conference call text. This serves as the “primary database” containing the following variables: Title (firm name), Subtitle (firm name, date, and whether final/primary transcripts), Date, Pages (the number of pages of the call), Analyst (analysts who collect these transcripts, different analysts have slightly different forms of transcripts.), Report (Unique report number), Call (Raw call transcripts).
- The code:
    - Start from the delimiters contained in the .txt file to identify pages
    - Use title and pages in the information file to locate the beginning and end of each conference call.
    - Generate a new variable in the information file to store the raw call scripts.

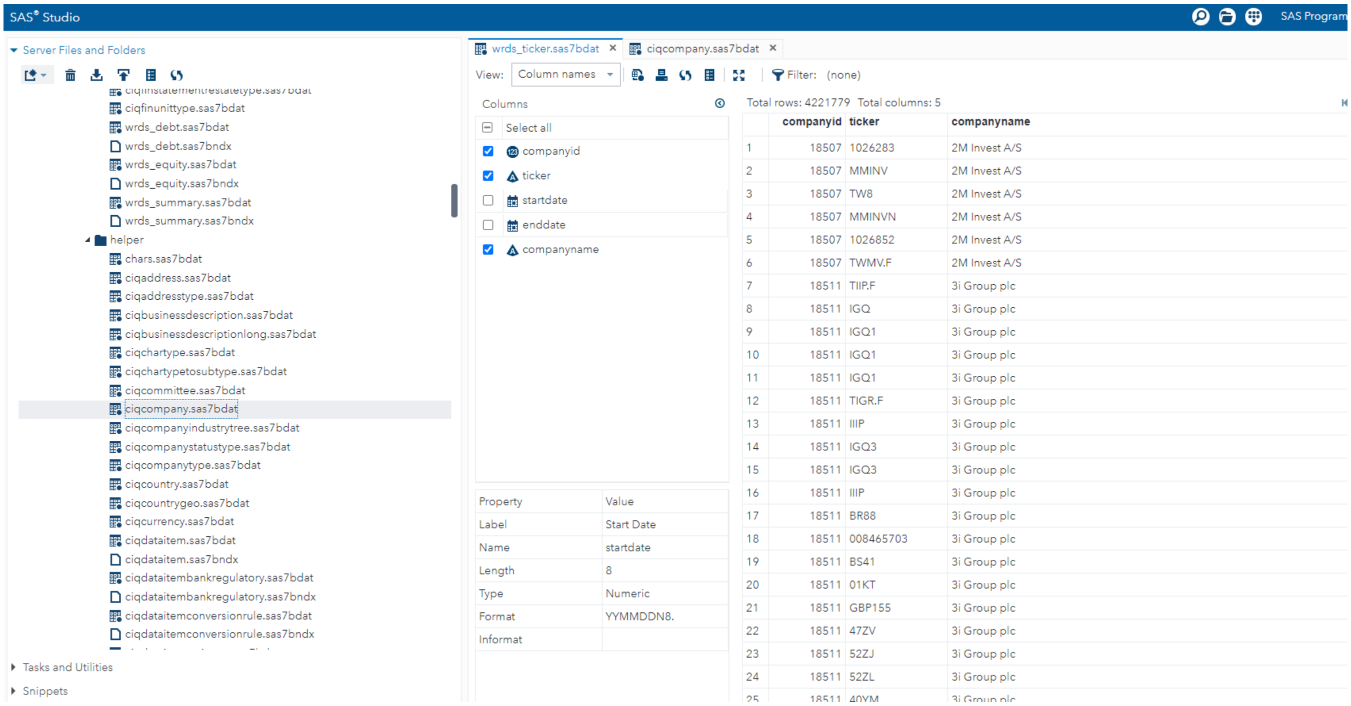
3 Firm Identification (Firm Name Matching)

3.1 Download Compustat datasets [L]

- This step aims to match firms in conference calls to gvkeys, a unique firm identifier, as well as country information. In the xls (and csv) files, the 'title' variable gives the firm name associated with a particular conference call.
- Gvkeys are found in the Compustat – Capital IQ datasets. Access the database using Wharton Research Data Services (WRDS).

Steps:

- Register for an account and wait for approval by the IT team: https://wrds-www.wharton.upenn.edu/register/
- Sign into the SAS-studio web application: https://wrds-www.wharton.upenn.edu/pages/data/sasstudio-wrds/
- On the left, there is a folder directory, titled "Server Files and Folder".
- Capital-IQ auxiliary files are located at Files -> wrds/capitaliq/sasdata/helper.



ciqfinunittype.sas7bdat
wrds_debt.sas7bdat
wrds_debt.sas7bndx
wrds_equity.sas7bdat
wrds_equity.sas7bndx
wrds_summary.sas7bdat
wrds_summary.sas7bndx
helper
chars.sas7bdat
ciqaddress.sas7bdat
ciqaddresstype.sas7bdat
ciqbusinessdescription.sas7bdat
ciqbusinessdescriptionlong.sas7bdat
ciqchartype.sas7bdat
ciqchartypetosubtype.sas7bdat
ciqcommittee.sas7bdat
ciqcompany.sas7bdat
ciqcompanyindustrytree.sas7bdat
ciqcompanystatustype.sas7bdat
ciqcompanytype.sas7bdat
ciqcountry.sas7bdat
ciqcountrygeo.sas7bdat
ciqcurrency.sas7bdat
ciqdataitem.sas7bdat
ciqdataitem.sas7bndx
ciqdataitembankregulatory.sas7bdat
ciqdataitembankregulatory.sas7bndx
ciqdataitemconversionrule.sas7bdat
ciqdataitemconversionrule.sas7bndx

Columns

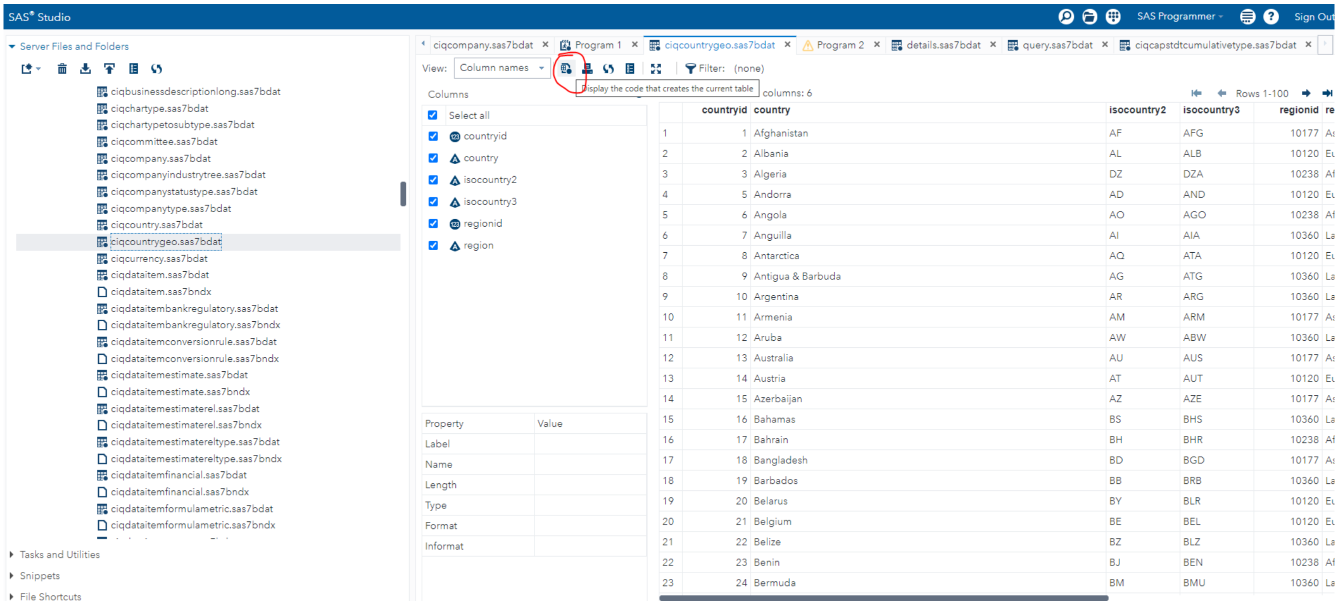
- ⊟ Select all
- ☑ companyid
- ☑ ticker
- ☐ startdate
- ☐ enddate
- ☑ companyname

Property	Value
Label	Start Date
Name	startdate
Length	8
Type	Numeric
Format	YYMMDDN8.
Informat	

Total rows: 4221779 Total columns: 5

	companyid	ticker	companyname
1	18507	1026283	2M Invest A/S
2	18507	MMINV	2M Invest A/S
3	18507	TW8	2M Invest A/S
4	18507	MMINVN	2M Invest A/S
5	18507	1026852	2M Invest A/S
6	18507	TWMV.F	2M Invest A/S
7	18511	TIIP.F	3i Group plc
8	18511	IGQ	3i Group plc
9	18511	IGQ1	3i Group plc
10	18511	IGQ1	3i Group plc
11	18511	IGQ1	3i Group plc
12	18511	TIGR.F	3i Group plc
13	18511	IIIP	3i Group plc
14	18511	IGQ3	3i Group plc
15	18511	IGQ3	3i Group plc
16	18511	IIIP	3i Group plc
17	18511	BR88	3i Group plc
18	18511	008465703	3i Group plc
19	18511	BS41	3i Group plc
20	18511	01KT	3i Group plc
21	18511	GBP155	3i Group plc
22	18511	47ZV	3i Group plc
23	18511	52ZJ	3i Group plc
24	18511	52ZL	3i Group plc
25	18511	40YM	3i Group plc

- Open the desired table.
- Select the desired columns.
- The best way to download data is to create Query (right mouse button on a table ⇒ new ⇒ Query).
- Downloading Query's result is a little bit tricky, since you can only print the result. The result is located in user's temporary folder.
- Click the button to "display the code that creates the current table".



ciqbusinessdescriptionlong.sas7bdat
ciqchartype.sas7bdat
ciqchartypetosubtype.sas7bdat
ciqcommittee.sas7bdat
ciqcompany.sas7bdat
ciqcompanyindustrytree.sas7bdat
ciqcompanystatustype.sas7bdat
ciqcompanytype.sas7bdat
ciqcountry.sas7bdat
ciqcountrygeo.sas7bdat
ciqcurrency.sas7bdat
ciqdataitem.sas7bdat
ciqdataitem.sas7bndx
ciqdataitembankregulatory.sas7bdat
ciqdataitembankregulatory.sas7bndx
ciqdataitemconversionrule.sas7bdat
ciqdataitemconversionrule.sas7bndx
ciqdataitemestimate.sas7bdat
ciqdataitemestimate.sas7bndx
ciqdataitemestimaterel.sas7bdat
ciqdataitemestimaterel.sas7bndx
ciqdataitemestimatereltype.sas7bdat
ciqdataitemestimatereltype.sas7bndx
ciqdataitemfinancial.sas7bdat
ciqdataitemfinancial.sas7bndx
ciqdataitemformulametric.sas7bdat
ciqdataitemformulametric.sas7bndx

Columns

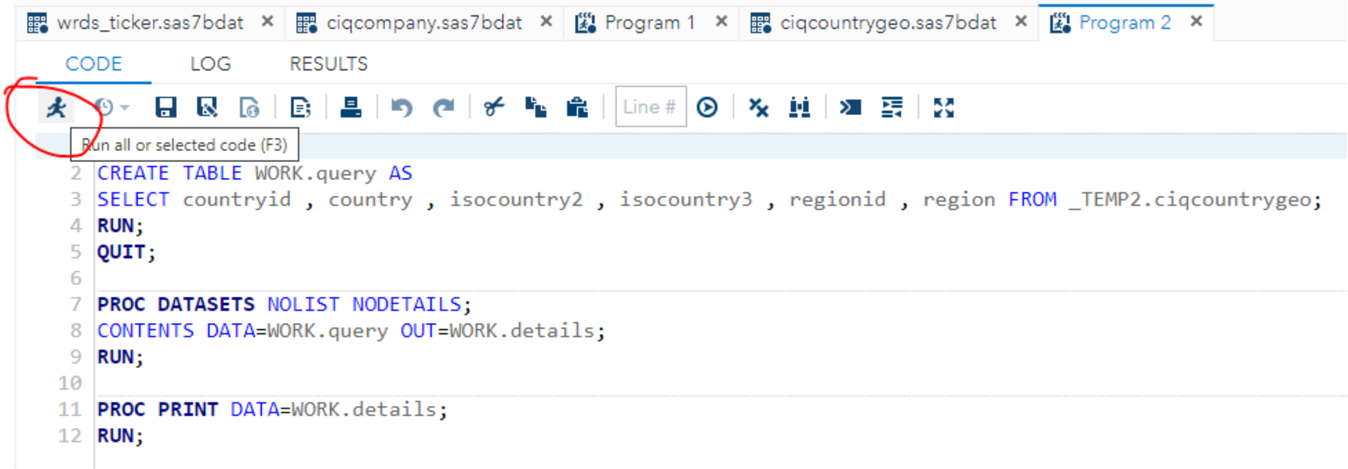
☑ Select all
☑ countryid
☑ country
☑ isocountry2
☑ isocountry3
☑ regionid
☑ region

Property	Value
Label	
Name	
Length	
Type	
Format	
Informat	

columns: 6 Rows 1-100

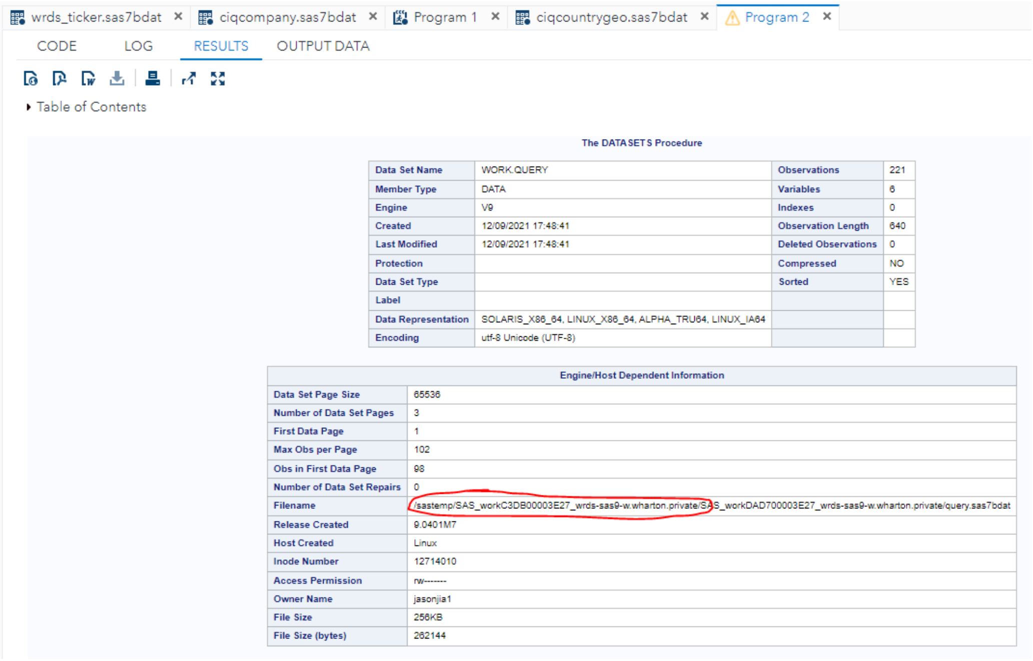
	countryid	country	isocountry2	isocountry3	regionid	re
1	1	Afghanistan	AF	AFG	10177	As
2	2	Albania	AL	ALB	10120	Eu
3	3	Algeria	DZ	DZA	10238	Af
4	5	Andorra	AD	AND	10120	Eu
5	6	Angola	AO	AGO	10238	Af
6	7	Anguilla	AI	AIA	10360	La
7	8	Antarctica	AQ	ATA	10120	Eu
8	9	Antigua & Barbuda	AG	ATG	10360	La
9	10	Argentina	AR	ARG	10360	La
10	11	Armenia	AM	ARM	10177	As
11	12	Aruba	AW	ABW	10360	La
12	13	Australia	AU	AUS	10177	As
13	14	Austria	AT	AUT	10120	Eu
14	15	Azerbaijan	AZ	AZE	10177	As
15	16	Bahamas	BS	BHS	10360	La
16	17	Bahrain	BH	BHR	10238	Af
17	18	Bangladesh	BD	BGD	10177	As
18	19	Barbados	BB	BRB	10360	La
19	20	Belarus	BY	BLR	10120	Eu
20	21	Belgium	BE	BEL	10120	Eu
21	22	Belize	BZ	BLZ	10360	La
22	23	Benin	BJ	BEN	10238	Af
23	24	Bermuda	BM	BMU	10360	La

- 3) run the code



```
CREATE TABLE WORK.query AS
SELECT countryid , country , isocountry2 , isocountry3 , regionid , region FROM _TEMP2.ciqcountrygeo;
RUN;
QUIT;

PROC DATASETS NOLIST NODETAILS;
CONTENTS DATA=WORK.query OUT=WORK.details;
RUN;

PROC PRINT DATA=WORK.details;
RUN;
```

- 4) Result window → Engine/Host Dependent Information → filename shows your temporary folder.

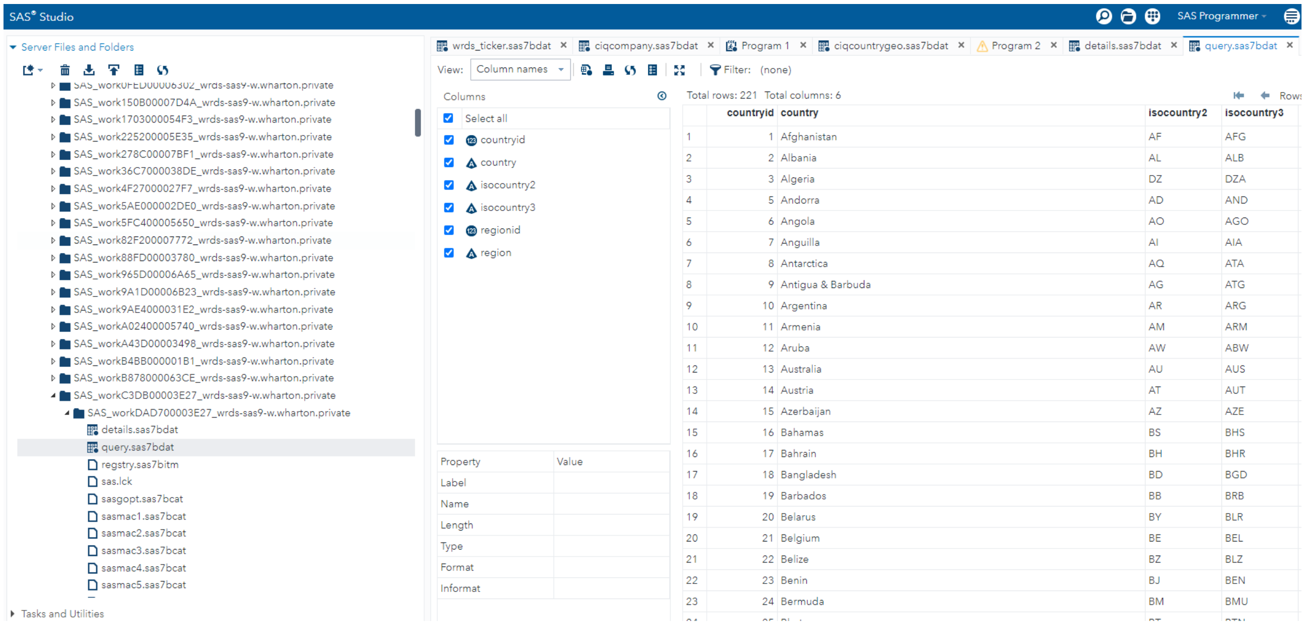


The DATASETS Procedure

Data Set Name	WORK.QUERY	Observations	221
Member Type	DATA	Variables	6
Engine	V9	Indexes	0
Created	12/09/2021 17:48:41	Observation Length	640
Last Modified	12/09/2021 17:48:41	Deleted Observations	0
Protection		Compressed	NO
Data Set Type		Sorted	YES
Label			
Data Representation	SOLARIS_X86_64, LINUX_X86_64, ALPHA_TRU64, LINUX_IA64		
Encoding	utf-8 Unicode (UTF-8)		

Engine/Host Dependent Information	
Data Set Page Size	65536
Number of Data Set Pages	3
First Data Page	1
Max Obs per Page	102
Obs in First Data Page	98
Number of Data Set Repairs	0
Filename	/sastemp/SAS_workC3DB00003E27_wrds-sas9-w.wharton.private/SAS_workDAD700003E27_wrds-sas9-w.wharton.private/query.sas7bdat
Release Created	9.0401M7
Host Created	Linux
Inode Number	12714010
Access Permission	rw-------
Owner Name	jasonjia1
File Size	256KB
File Size (bytes)	262144

- Go to the temporary folder and open query.sas7bdat to confirm this is the dataset you want to download.



SAS_work0FED00006302_wrds-sas9-w.wharton.private
SAS_work150B00007D4A_wrds-sas9-w.wharton.private
SAS_work1703000054F3_wrds-sas9-w.wharton.private
SAS_work225200005E35_wrds-sas9-w.wharton.private
SAS_work278C00007BF1_wrds-sas9-w.wharton.private
SAS_work36C7000038DE_wrds-sas9-w.wharton.private
SAS_work4F27000027F7_wrds-sas9-w.wharton.private
SAS_work5AE000002DE0_wrds-sas9-w.wharton.private
SAS_work5FC400005650_wrds-sas9-w.wharton.private
SAS_work82F200007772_wrds-sas9-w.wharton.private
SAS_work88FD00003780_wrds-sas9-w.wharton.private
SAS_work965D00006A65_wrds-sas9-w.wharton.private
SAS_work9A1D00006B23_wrds-sas9-w.wharton.private
SAS_work9AE4000031E2_wrds-sas9-w.wharton.private
SAS_workA02400005740_wrds-sas9-w.wharton.private
SAS_workA43D00003498_wrds-sas9-w.wharton.private
SAS_workB4BB000001B1_wrds-sas9-w.wharton.private
SAS_workB878000063CE_wrds-sas9-w.wharton.private
SAS_workC3DB00003E27_wrds-sas9-w.wharton.private
SAS_workDAD700003E27_wrds-sas9-w.wharton.private
details.sas7bdat
query.sas7bdat
regstry.sas7bitm
sas.lck
sasgopt.sas7bcat
sasmac1.sas7bcat
sasmac2.sas7bcat
sasmac3.sas7bcat
sasmac4.sas7bcat
sasmac5.sas7bcat

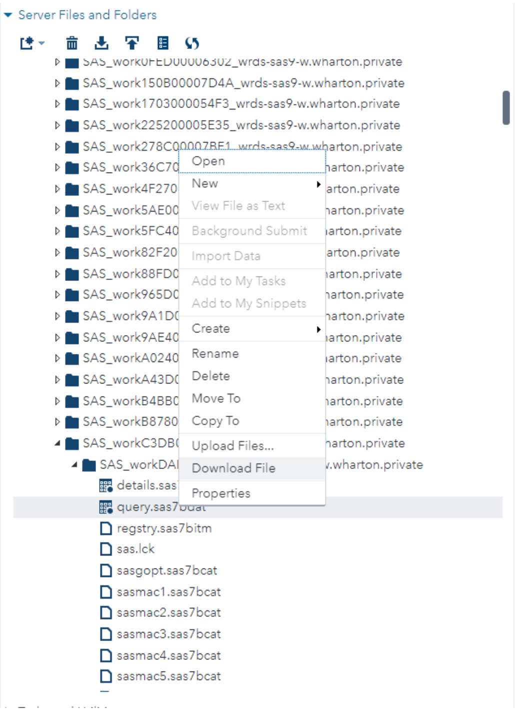
☑ Select all
☑ countryid
☑ country
☑ isocountry2
☑ isocountry3
☑ regionid
☑ region

Property	Value
Label	
Name	
Length	
Type	
Format	
Informat	

Total rows: 221 Total columns: 6

	countryid	country	isocountry2	isocountry3
1	1	Afghanistan	AF	AFG
2	2	Albania	AL	ALB
3	3	Algeria	DZ	DZA
4	5	Andorra	AD	AND
5	6	Angola	AO	AGO
6	7	Anguilla	AI	AIA
7	8	Antarctica	AQ	ATA
8	9	Antigua & Barbuda	AG	ATG
9	10	Argentina	AR	ARG
10	11	Armenia	AM	ARM
11	12	Aruba	AW	ABW
12	13	Australia	AU	AUS
13	14	Austria	AT	AUT
14	15	Azerbaijan	AZ	AZE
15	16	Bahamas	BS	BHS
16	17	Bahrain	BH	BHR
17	18	Bangladesh	BD	BGD
18	19	Barbados	BB	BRB
19	20	Belarus	BY	BLR
20	21	Belgium	BE	BEL
21	22	Belize	BZ	BLZ
22	23	Benin	BJ	BEN
23	24	Bermuda	BM	BMU

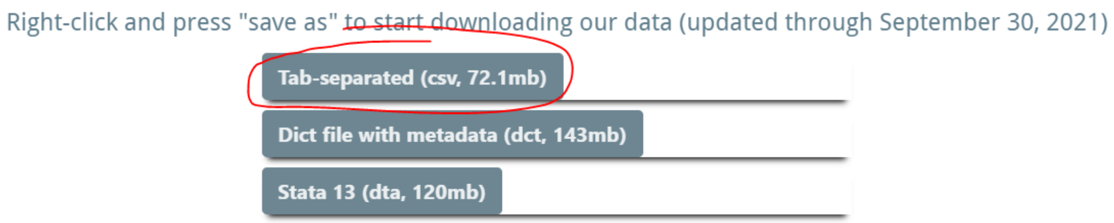
- Right click on query.sas7bdat and click download file.



- We used the following tables with specific columns from Compustat:
  - 1. CIQCOMPANY: companyid, companyname, tickersymbol
  - , countryid and other columns.
  - Total: 24,511,757 obs. Public Companies: 66,256. Private Companies: 16,544,322. Public Investment Firms: 1,987 and Private Investment Firms: 203,407
  - 2. WRDS GVKEY: companyid, gvkey (115,357 observations).
  - 3. CIQCOUNTRY: countryid, countryname: countryid, countryname (221 countries).

3.2 Download Hassan dataset [L]

- Gvkeys are also found in Hassan's Firm-Level Political Risk dataset.
- Go to https://www.firmlevelrisk.com/download and download the tab-separated file in csv.
- This data set was also used because it also uses conference calls, but have already matched firm names to gvkeys, which would help in our firm name matching. Note that the dataset is updated over time.



Right-click and press "save as" to start downloading our data (updated through September 30, 2021)

Tab-separated (csv, 72.1mb)
Dict file with metadata (dct, 143mb)
Stata 13 (dta, 120mb)

3.3 Process Compustat and Hassan datasets into usable and truncated .csv files. [L]

- Go to conference_call\code\03_firm_identification
- Run the codes convert_compustat.py on terminal and process_hassan.py on terminal.

3.4 Match titles in conference calls with firm names in Hassan and Compustat datasets, with both exact and fuzzy matching [M, L]

[to be added]

4 Keyword Identification

4.1 Make a folder structure with x groups (default: x = 50) [M, L]

- Run mkdir.py, specifying the output folder that will contain x groups.
- The output will be x empty sub-folders in the output folder, named group1, …, groupx.

4.2 Make a list of keywords and template entry file [L]

- This is done manually. The existing version of keywords and template entry file are found in \conference_call\output\04_keyword_identification\04.2_reference_files as keterms.txt and Entrymask.xlsx.
- Changes to keyterms are recorded in changelog.txt. You can also consider adding suffixes to record different sets of keywords.

4.3 Identify keywords in each conference call [M, L]

- The csvs will now be copied over and divided equally into 50 (or x) groups, to enable parallelization on Mercury.
- Run dividefilesequallyintofolders.py.
- Then, identify keywords in each conference call.
- Run keyword_ident_1.py.
- The output will be

4.4 Extract all paragraphs from each conference call that contains a specific keyword [M, L]

- -

4.5 Cleans the identified matches and merges with gvkey dataset [L]

- -

4.6 Make a paragraph record file that splits the number of entries into groups of 500 [L]

- Run make_paragraphrecord.py and inputting the number of entries.

4.7 Bold the keywords and separate file into “entryfiles”, each containing 500 entries. [M, L]

- If doing locally, run makeentryfiles.py; if doing on Mercury, run makeentryfiles.sh

4.8 Combine entry files [L]

- Run combine_entryfiles.py.