openmainframeChallenge3.2_simplified_answer

This is Simple solution to OpenMainframe Advanced Challenge3.2 using only ISPF and Mainframe COBOL (non-Zowe solution)

The details of this project can be found at https://github.com/openmainframeproject/cobol-programming-course . A most elegant solution using Typescript to automate json to csv conversion + the Zowe API Visual Studio plugin to access the IBM sponsored mainframe - by the code master/challenger here https://medium.com/@jessielaine.punongbayan/solution-covid-19-reports-cobol-challenge-6c509579e3fe

Zowe does a very nice job bringing the VS IDE experience to z/OS mainfraime COBOL. IBM made free mainframe access availble for the 'COBOL Programming with VSCode' challenge. The access included a Zowe interface for working through the program with VSCode. Very nice however *only availble on a limited basis.*

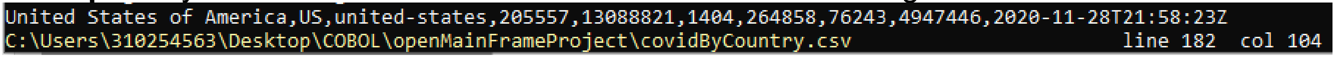
If like me, you are renting $50/mo access from a commercial mainframe provider, the Zowe interface is generally not available. No problem! Here I show a simple solution to this challenge using only ISPF and COBOL from a standard mainframe access provider.

Step 1: As instructed, Use the API: https://api.covid19api.com/summary. To extract the JSON file.

Step 2: Convert JSON to (comma delimited) CSV format using the conversion method of your choice. (HINT: There are several online converters available)

Step 3: Using your favorite cmd line editor, scroll through the CSV and determine the max record length in characters. This is an important detail for the next step.

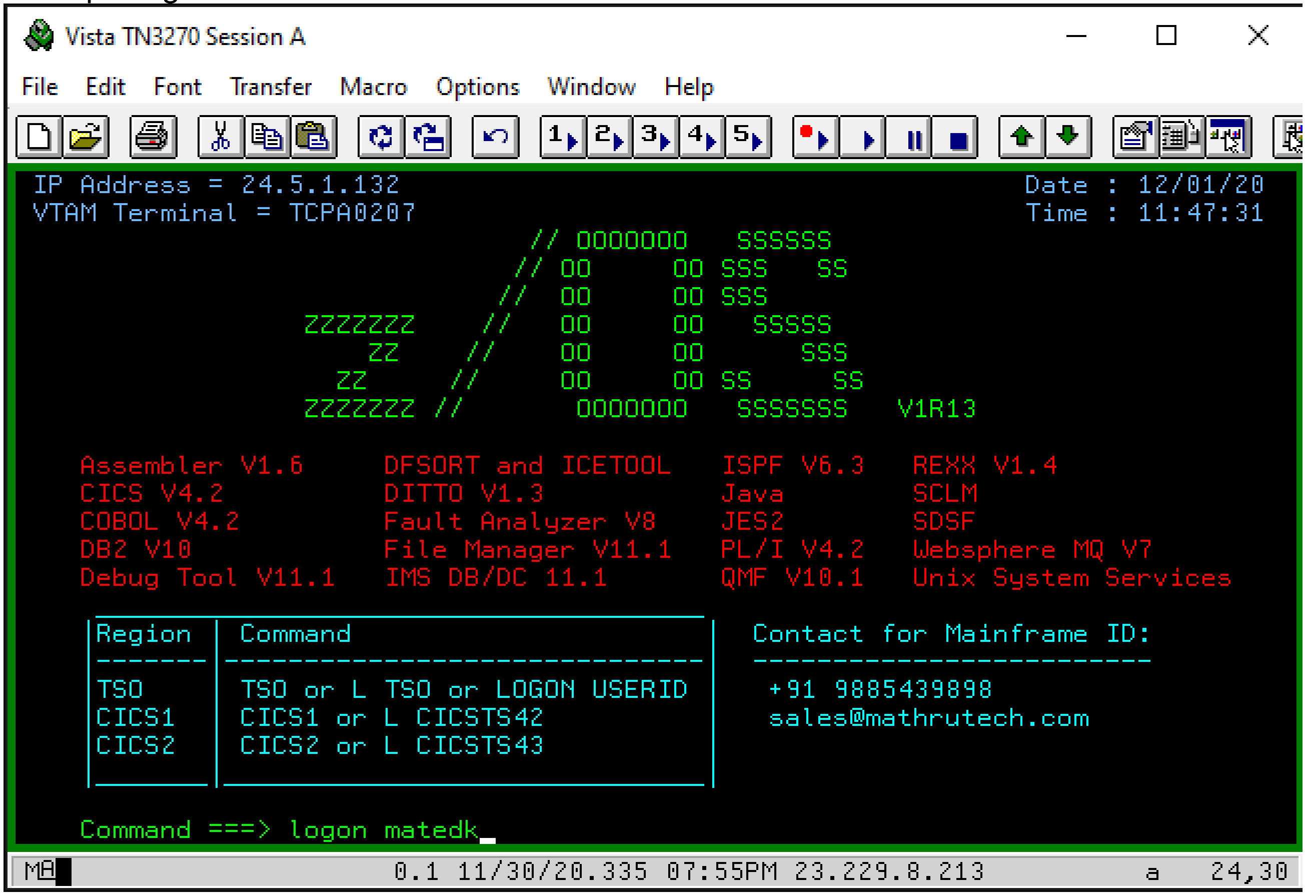
Example: My CSV shows United States of America with the longest record at 104 characters.



```
United States of America,US,united-states,205557,13088821,1404,264858,76243,4947446,2020-11-28T21:58:23Z
C:\Users\310254563\Desktop\COBOL\openMainFrameProject\covidByCountry.csv              line 182  col 104
```

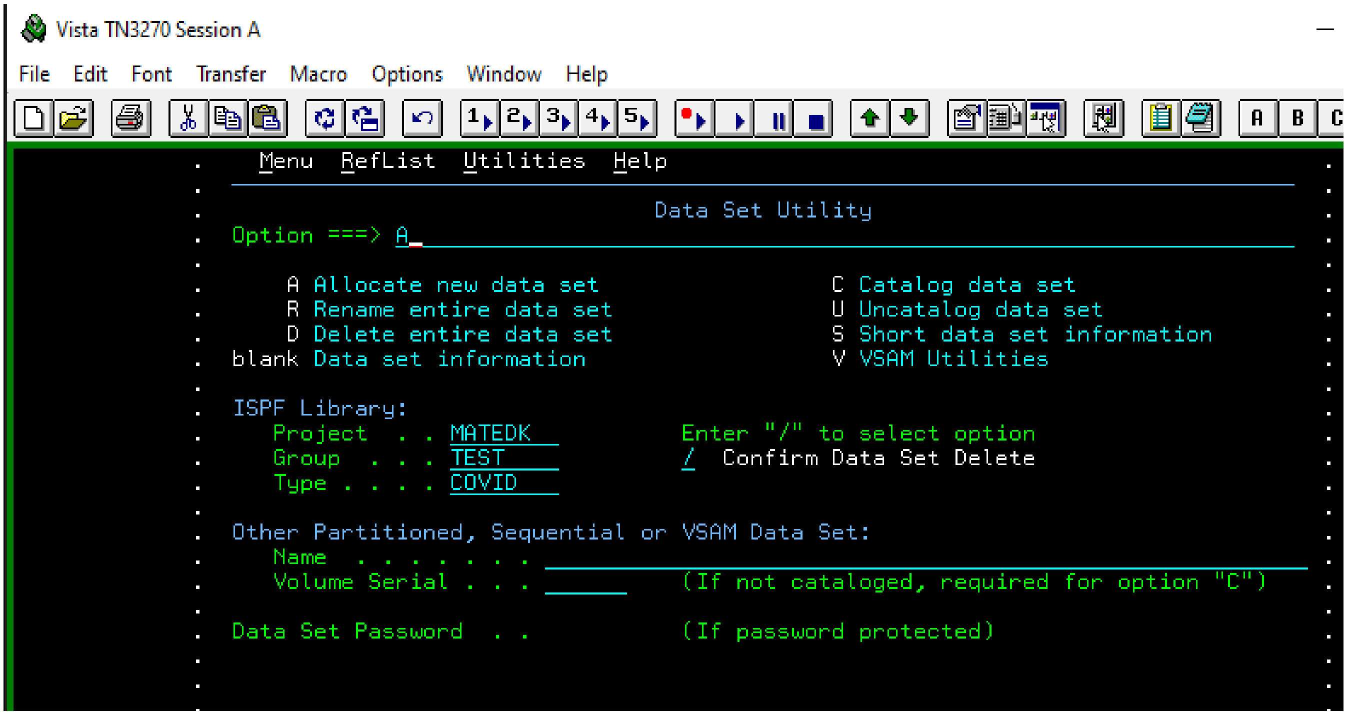
Step 4: Logon to the Mainframe.

Example logon screen.



```
IP Address = 24.5.1.132                                       Date : 12/01/20
VTAM Terminal = TCPA0207                                      Time : 11:47:31
                               // OOOOOOO   SSSSSS
                              // OO     OO SSS   SS
                             //  OO     OO SSS
                 ZZZZZZZ    //   OO     OO   SSSSS
                     ZZ    //    OO     OO      SSS
                   ZZ     //     OO     OO SS     SS
                 ZZZZZZZ //       OOOOOOO   SSSSSSS   V1R13

   Assembler V1.6     DFSORT and ICETOOL   ISPF V6.3   REXX V1.4
   CICS V4.2          DITTO V1.3           Java        SCLM
   COBOL V4.2         Fault Analyzer V8    JES2        SDSF
   DB2 V10            File Manager V11.1   PL/I V4.2   Websphere MQ V7
   Debug Tool V11.1   IMS DB/DC 11.1       QMF V10.1   Unix System Services

   |Region |Command                       |  Contact for Mainframe ID:
   |-------|------------------------------|  -------------------------
   |TSO    | TSO or L TSO or LOGON USERID |   +91 9885439898
   |CICS1  | CICS1 or L CICSTS42          |   sales@mathrutech.com
   |CICS2  | CICS2 or L CICSTS43          |
   |       |                              |
   |_______|______________________________|

   Command ===> logon matedk_
 MA                0.1 11/30/20.335 07:55PM 23.229.8.213         a    24,30
```

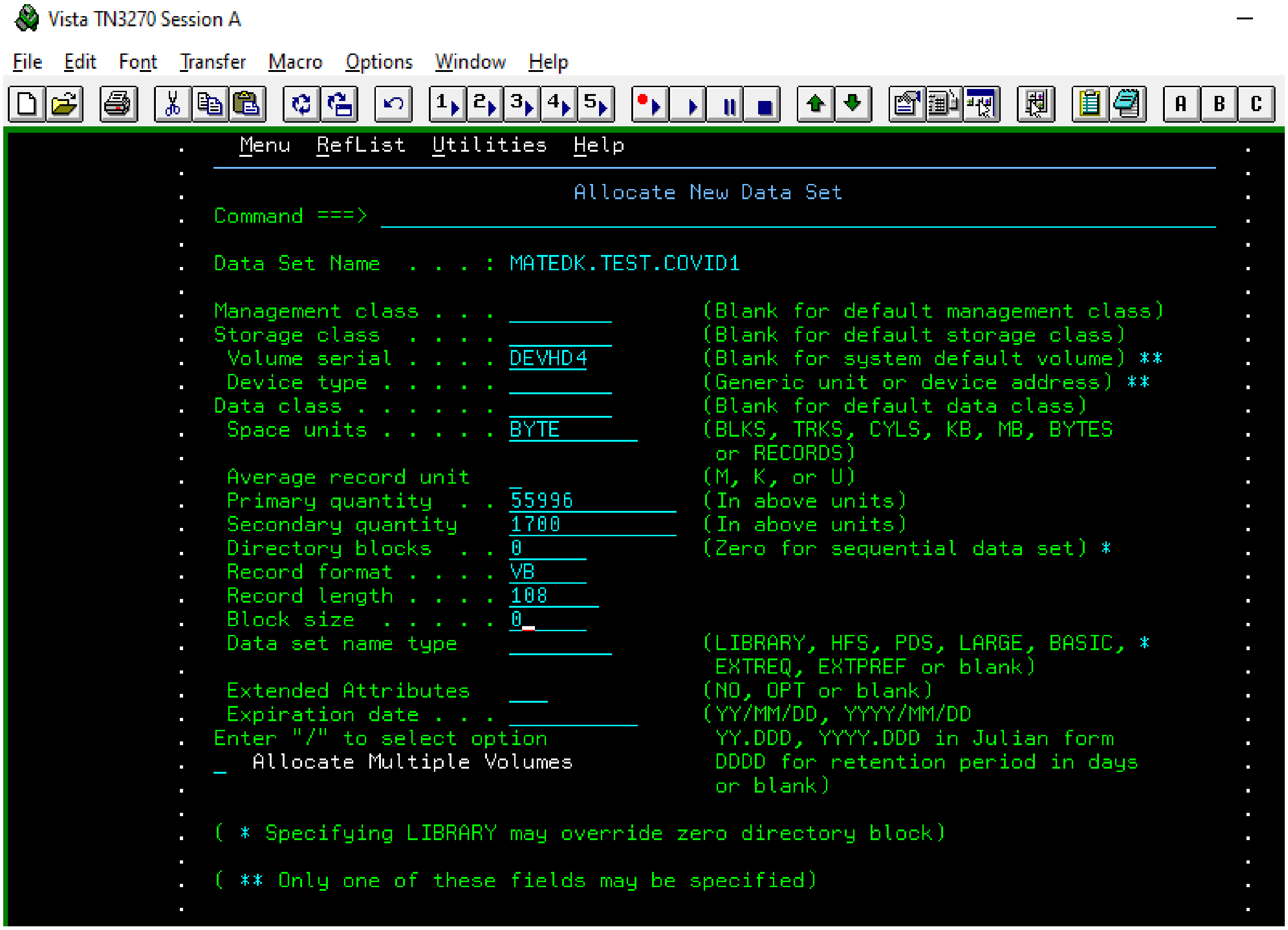
Step 5: Use the ISPF Data Set Utility to create a PDS as seen below, which will receive the CVS File.



```
   Menu  RefList  Utilities  Help
 ------------------------------------------------------------------------------
                             Data Set Utility
 Option ===> A_
 ------------------------------------------------------------------------------
     A Allocate new data set                   C Catalog data set
     R Rename entire data set                  U Uncatalog data set
     D Delete entire data set                  S Short data set information
 blank Data set information                    V VSAM Utilities

 ISPF Library:
    Project  . . MATEDK           Enter "/" to select option
    Group  . . . TEST             /  Confirm Data Set Delete
    Type . . . . COVID

 Other Partitioned, Sequential or VSAM Data Set:
    Name  . . . . . . .
    Volume Serial . . .           (If not cataloged, required for option "C")

 Data Set Password  . .           (If password protected)
```

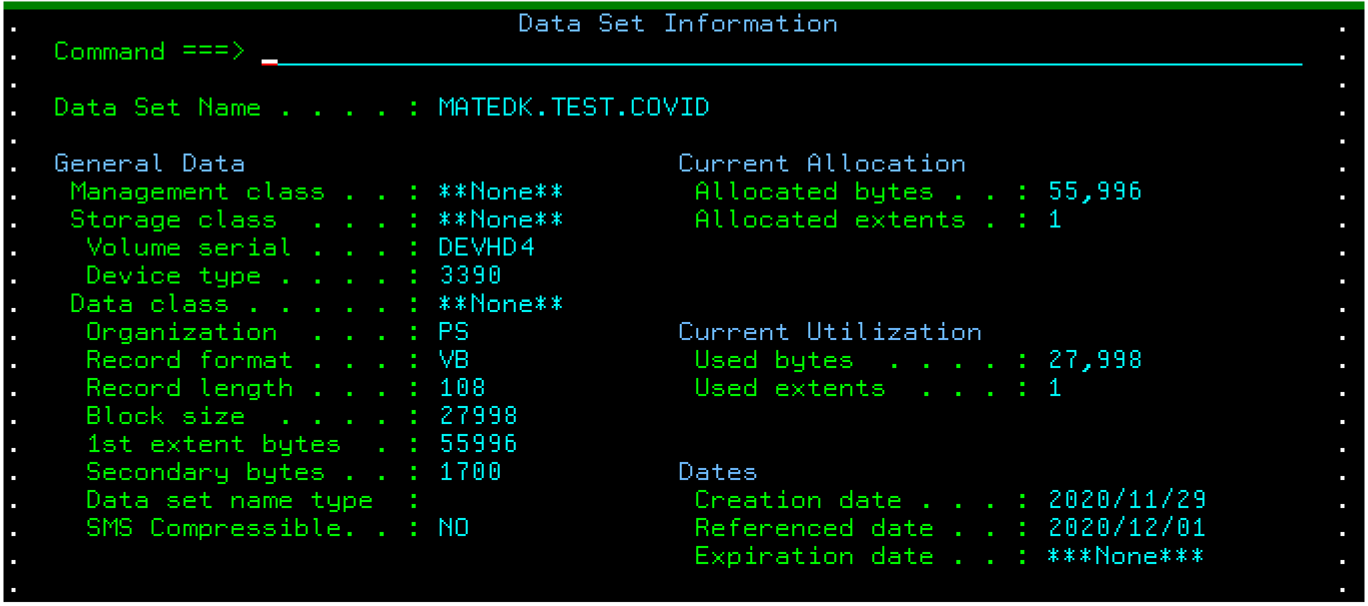
Set space units to byte, Record Format to VB, Record Length to max record length in chars + 4 (for my example 104 + 4) and Block Size 0.

Example:



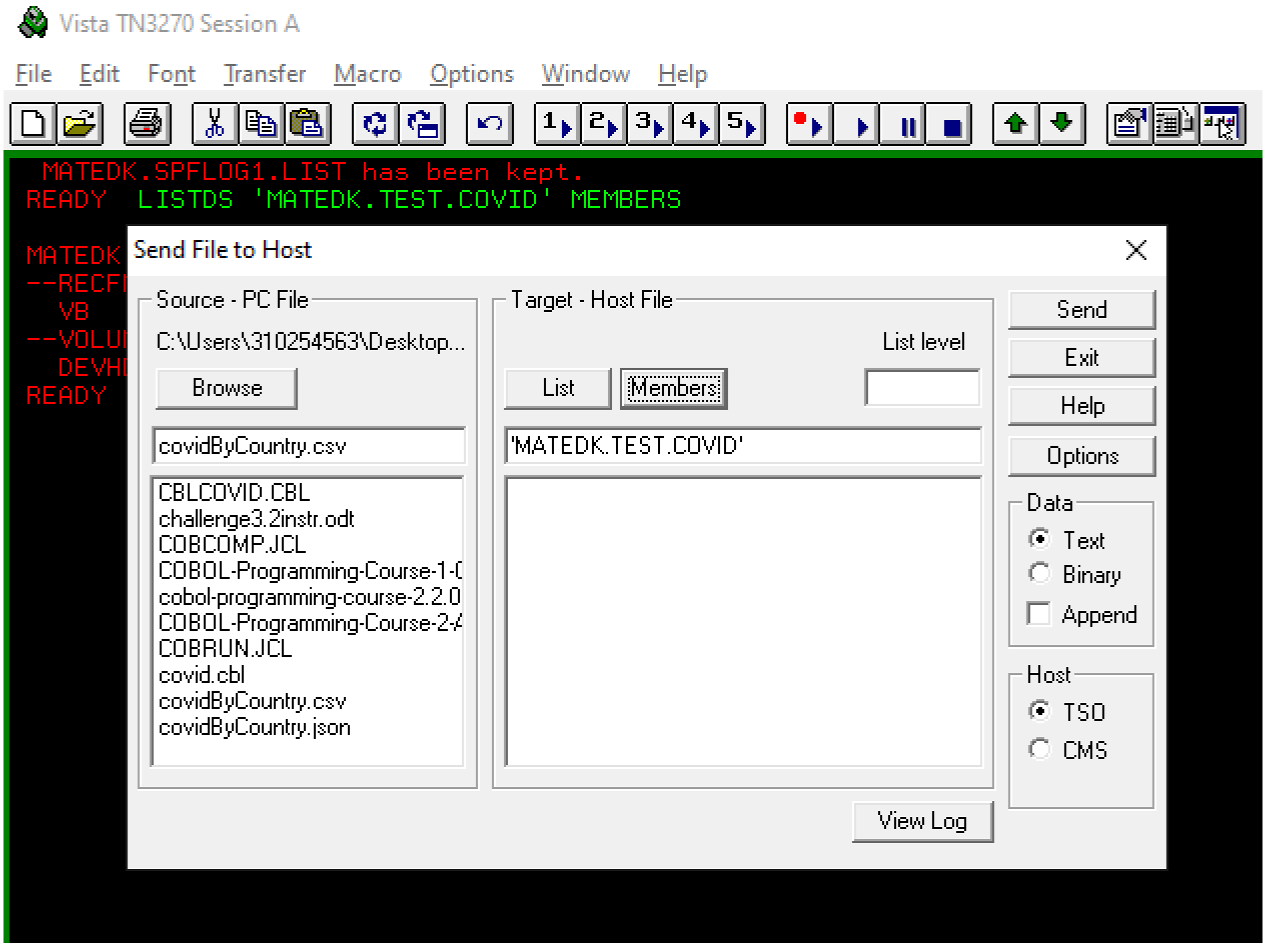
```
  Menu  RefList  Utilities  Help
 ------------------------------------------------------------------------------
                            Allocate New Data Set
 Command ===> _________________________________________________________________

 Data Set Name  . . . : MATEDK.TEST.COVID1

 Management class . . . ________      (Blank for default management class)
 Storage class  . . . . ________      (Blank for default storage class)
  Volume serial . . . . DEVHD4__      (Blank for system default volume) **
  Device type . . . . . ________      (Generic unit or device address) **
 Data class . . . . . . ________      (Blank for default data class)
  Space units . . . . . BYTE______    (BLKS, TRKS, CYLS, KB, MB, BYTES
                                       or RECORDS)
  Average record unit   _             (M, K, or U)
  Primary quantity  . . 55996_______  (In above units)
  Secondary quantity    1700________  (In above units)
  Directory blocks  . . 0_____        (Zero for sequential data set) *
  Record format . . . . VB____
  Record length . . . . 108____
  Block size  . . . . . 0_____
  Data set name type    ________      (LIBRARY, HFS, PDS, LARGE, BASIC, *
                                       EXTREQ, EXTPREF or blank)
  Extended Attributes   ___           (NO, OPT or blank)
  Expiration date . . . __________    (YY/MM/DD, YYYY/MM/DD
 Enter "/" to select option            YY.DDD, YYYY.DDD in Julian form
 _  Allocate Multiple Volumes          DDDD for retention period in days
                                       or blank)

 ( * Specifying LIBRARY may override zero directory block)

 ( ** Only one of these fields may be specified)
```

New Data Set Created.
Example:



```
                            Data Set Information
 Command ===> _________________________________________________________________

 Data Set Name . . . . : MATEDK.TEST.COVID

 General Data                          Current Allocation
  Management class . . : **None**       Allocated bytes . . : 55,996
  Storage class  . . . : **None**       Allocated extents . : 1
   Volume serial . . . : DEVHD4
   Device type . . . . : 3390
  Data class . . . . . : **None**
   Organization  . . . : PS            Current Utilization
   Record format . . . : VB             Used bytes  . . . . : 27,998
   Record length . . . : 108            Used extents  . . . : 1
   Block size  . . . . : 27998
   1st extent bytes  . : 55996
   Secondary bytes . . : 1700          Dates
   Data set name type  :                Creation date . . . : 2020/11/29
   SMS Compressible. . : NO             Referenced date . . : 2020/12/01
                                        Expiration date . . : ***None***
```

Step 6: Use the TSO transfer utility to copy the CSV to the new data set. Verify the contents.



```
 MATEDK.SPFLOG1.LIST has been kept.
READY  LISTDS 'MATEDK.TEST.COVID' MEMBERS

MATEDK
--RECF
  VB
--VOLU
  DEVH
READY
```

Send File to Host

Source - PC File: C:\Users\310254563\Desktop...

covidByCountry.csv

```
CBLCOVID.CBL
challenge3.2instr.odt
COBCOMP.JCL
COBOL-Programming-Course-1-(
cobol-programming-course-2.2.0
COBOL-Programming-Course-2-A
COBRUN.JCL
covid.cbl
covidByCountry.csv
covidByCountry.json
```

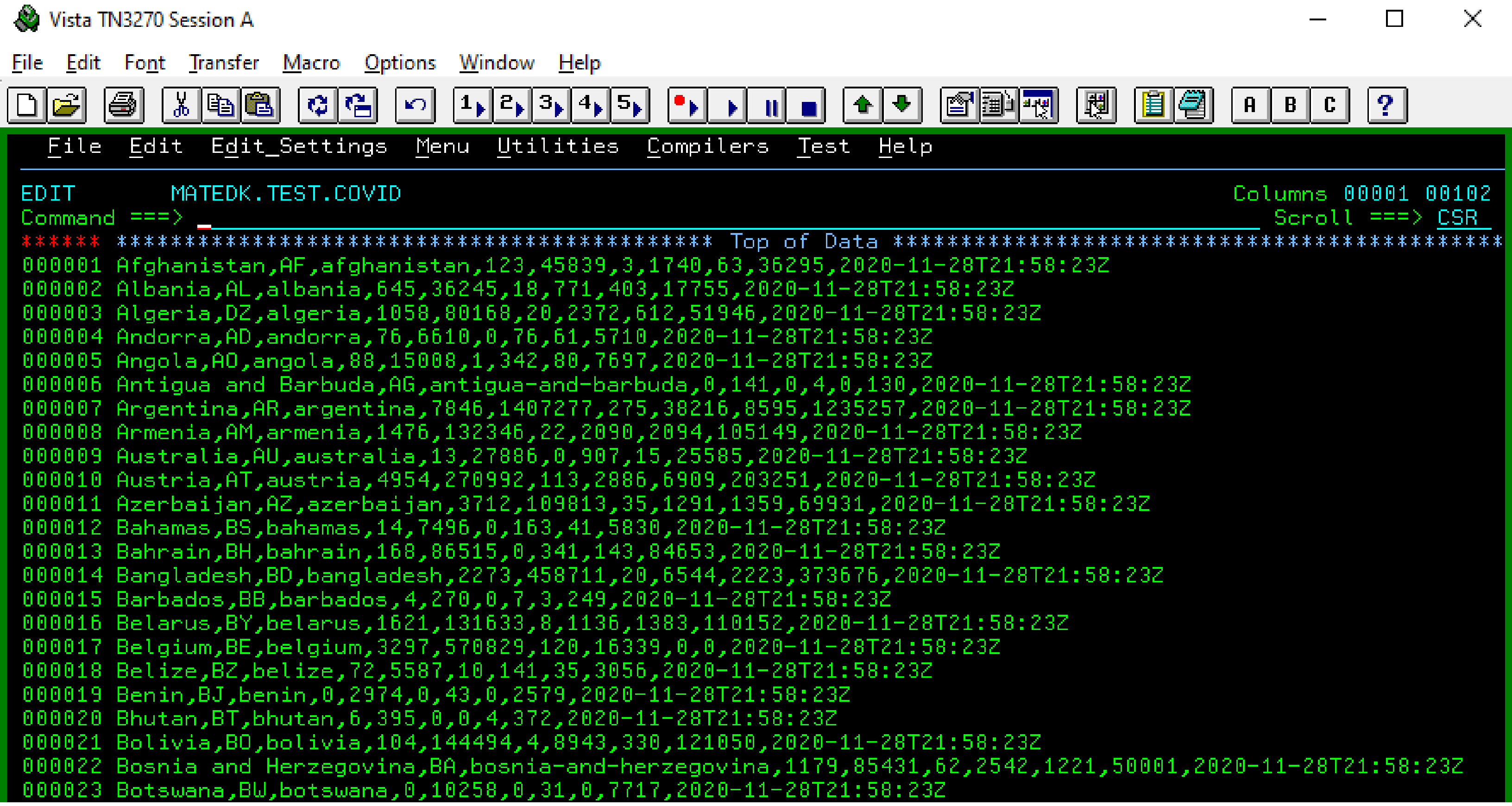
Target - Host File: 'MATEDK.TEST.COVID'

List level:

Data: ● Text ○ Binary □ Append

Host: ● TSO ○ CMS

Example: Content of CSV file transferred to *USERID.*TEST.COVID.



```
  File  Edit  Edit_Settings  Menu  Utilities  Compilers  Test  Help
EDIT       MATEDK.TEST.COVID                                   Columns 00001 00102
Command ===> _                                                    Scroll ===> CSR
****** ******************************************** Top of Data **************************************************
000001 Afghanistan,AF,afghanistan,123,45839,3,1740,63,36295,2020-11-28T21:58:23Z
000002 Albania,AL,albania,645,36245,18,771,403,17755,2020-11-28T21:58:23Z
000003 Algeria,DZ,algeria,1058,80168,20,2372,612,51946,2020-11-28T21:58:23Z
000004 Andorra,AD,andorra,76,6610,0,76,61,5710,2020-11-28T21:58:23Z
000005 Angola,AO,angola,88,15008,1,342,80,7697,2020-11-28T21:58:23Z
000006 Antigua and Barbuda,AG,antigua-and-barbuda,0,141,0,4,0,130,2020-11-28T21:58:23Z
000007 Argentina,AR,argentina,7846,1407277,275,38216,8595,1235257,2020-11-28T21:58:23Z
000008 Armenia,AM,armenia,1476,132346,22,2090,2094,105149,2020-11-28T21:58:23Z
000009 Australia,AU,australia,13,27886,0,907,15,25585,2020-11-28T21:58:23Z
000010 Austria,AT,austria,4954,270992,113,2886,6909,203251,2020-11-28T21:58:23Z
000011 Azerbaijan,AZ,azerbaijan,3712,109813,35,1291,1359,69931,2020-11-28T21:58:23Z
000012 Bahamas,BS,bahamas,14,7496,0,163,41,5830,2020-11-28T21:58:23Z
000013 Bahrain,BH,bahrain,168,86515,0,341,143,84653,2020-11-28T21:58:23Z
000014 Bangladesh,BD,bangladesh,2273,458711,20,6544,2223,373676,2020-11-28T21:58:23Z
000015 Barbados,BB,barbados,4,270,0,7,3,249,2020-11-28T21:58:23Z
000016 Belarus,BY,belarus,1621,131633,8,1136,1383,110152,2020-11-28T21:58:23Z
000017 Belgium,BE,belgium,3297,570829,120,16339,0,0,2020-11-28T21:58:23Z
000018 Belize,BZ,belize,72,5587,10,141,35,3056,2020-11-28T21:58:23Z
000019 Benin,BJ,benin,0,2974,0,43,0,2579,2020-11-28T21:58:23Z
000020 Bhutan,BT,bhutan,6,395,0,0,4,372,2020-11-28T21:58:23Z
000021 Bolivia,BO,bolivia,104,144494,4,8943,330,121050,2020-11-28T21:58:23Z
000022 Bosnia and Herzegovina,BA,bosnia-and-herzegovina,1179,85431,62,2542,1221,50001,2020-11-28T21:58:23Z
000023 Botswana,BW,botswana,0,10258,0,31,0,7717,2020-11-28T21:58:23Z
```

...

Step 7. Ready to go. Here is the COBOL program to read the DATA SET and DISPLAY the output in the requested format. I added a little extra formatting to the numbers for readability.

```
*-----------------------
 IDENTIFICATION DIVISION.
*------------------------
* My 'simple answer to Openmainframe challenge 3.2
* Using data set utility first create  a PDS with 'V'ariable
* record length. Then transfer the cvs file to the VB data set.
* It does not work if for a FB data set. (because of uneven record length)
*-----------------------
 PROGRAM-ID.    CBLCOVID
 AUTHOR.        J_MAC.
*--------------------
 ENVIRONMENT DIVISION.
*--------------------
 INPUT-OUTPUT SECTION.
 FILE-CONTROL.

      SELECT IN001 ASSIGN TO COVIDIN
      ORGANIZATION IS SEQUENTIAL
      ACCESS MODE IS SEQUENTIAL.

*-------------
 DATA DIVISION.
*-------------
 FILE SECTION.

 FD  IN001 RECORDING MODE V.
 01  COVID-REC-FIELDS   PIC X(104).

*
 WORKING-STORAGE SECTION.

 01  UNSTRING-COVID-RECORDS.
     05  UCR-COUNTRY         PIC X(50).
     05  UCR-COUNTRY-CODE    PIC X(4).
     05  UCR-SLUG            PIC X(50).
     05  UCR-NEW-CNFRM       PIC 9(5).
     05  UCR-TOT-CNFRM       PIC 9(8).
     05  UCR-NEW-DEATH       PIC 9(5).
     05  UCR-TOT-DEATH       PIC 9(5).
     05  UCR-NEW-RECVR       PIC 9(5).
     05  UCR-TOT-RECVR       PIC 9(8).
     05  UCR-TIMESTAMP       PIC X(25).

 01  WS-ASTER            PIC X(80) VALUE ALL '*'.

 01 NUMBER-DISPLAY-FORMAT.
     05 NEW-CNFRM            PIC ZZZ,999.
     05 TOT-CNFRM            PIC ZZ,ZZZ,999.
```

```
    05 NEW-DEATH            PIC ZZZ,999.
    05 TOT-DEATH           PIC ZZZ,999.
    05 NEW-RECVR            PIC ZZZ,999.
    05 TOT-RECVR           PIC ZZ,ZZZ,999.

 01 FLAGS.
  05 LASTREC          PIC X VALUE SPACE.
*------------------
 PROCEDURE DIVISION.
*------------------
 MAIN.
     OPEN INPUT IN001.

     PERFORM UNTIL LASTREC = 'Y'
       PERFORM READ-RECORD
       PERFORM DISPLAY-RECORD
     END-PERFORM.

     CLOSE IN001
     STOP RUN.

 READ-RECORD.
    READ IN001
    AT END MOVE 'Y' TO LASTREC
    END-READ.

 DISPLAY-RECORD.
    UNSTRING COVID-REC-FIELDS DELIMITED BY ','
       INTO UCR-COUNTRY
            UCR-COUNTRY-CODE
            UCR-SLUG
            UCR-NEW-CNFRM
            UCR-TOT-CNFRM
            UCR-NEW-DEATH
```

```
         UCR-TOT-DEATH
         UCR-NEW-RECVR
         UCR-TOT-RECVR
         UCR-TIMESTAMP.

****  // DISPLAY-FRIENDLY NUMBER FORMAT
    MOVE UCR-NEW-CNFRM TO NEW-CNFRM
    MOVE UCR-TOT-CNFRM TO TOT-CNFRM
    MOVE UCR-NEW-DEATH TO NEW-DEATH
    MOVE UCR-TOT-DEATH TO TOT-DEATH
    MOVE UCR-NEW-RECVR TO NEW-RECVR
    MOVE UCR-TOT-RECVR TO TOT-RECVR

    DISPLAY 'DATE: ' UCR-TIMESTAMP(1:10)
    DISPLAY 'TIME: ' UCR-TIMESTAMP(12:8)
    DISPLAY 'COUNTRY: ' UCR-COUNTRY
    DISPLAY 'COUNTRY CODE: ' UCR-COUNTRY-CODE
    DISPLAY 'SLUG: ' UCR-SLUG
    DISPLAY 'NEW CONFIRMED CASES: ' NEW-CNFRM
    DISPLAY 'TOTAL CONFIRMED CASES: ' TOT-CNFRM
    DISPLAY 'NEW DEATHS: ' NEW-DEATH
    DISPLAY 'TOTAL DEATHS: ' TOT-DEATH
    DISPLAY 'NEW RECOVERIES: ' NEW-RECVR
    DISPLAY 'TOTAL RECOVERIES: ' TOT-RECVR
    DISPLAY WS-ASTER.
```

Compile JCL:

```
//MATEDKD JOB MSGLEVEL=(1,1),NOTIFY=&SYSUID
//PLIB    JCLLIB ORDER=(MATE1.PROCLIB)
//*
//*  COMPILE A COBOL PROGRAM
//*
//CL      EXEC COBOLCL,
//             COPYLIB=MATEDK.COPYLIB,          <= COPYBOOK LIBRARY
//             LOADLIB=MATEDK.LOADLIB,          <= LOAD LIBRARY
//             SRCLIB=MATEDK.COBOL.SRCLIB,      <= SOURCE LIBRARY
//             MEMBER=CBLCOVID                  <= SOURCE MEMBER
```

Run JCL:

```
//MATEDKW JOB MSGLEVEL=(1,1),NOTIFY=&SYSUID
//*
//*  RUN A COBOL PROGRAM
//*
//STEP01  EXEC PGM=CBLCOVID
//STEPLIB  DD  DSN=MATEDK.LOADLIB,DISP=SHR
//COVIDIN  DD DSN=MATEDK.TEST.COVID,DISP=SHR
//PRTLINE   DD SYSOUT=*
//SYSOUT   DD  SYSOUT=*
```

Sample Output:

```
********************************************************************************
DATE: 2020-11-28
TIME: 21:58:23
COUNTRY: Bolivia
COUNTRY CODE: BO
SLUG: bolivia
NEW CONFIRMED CASES:     104
TOTAL CONFIRMED CASES:    144,494
NEW DEATHS:     004
TOTAL DEATHS:   8,943
NEW RECOVERIES:     330
TOTAL RECOVERIES:    121,050
********************************************************************************
```