

## Paper 108-27

**Guide to extract/download multiple databases from Mainframe Tapes to PC using SAS® PC**

Fereydoun Foroudian, Blue Cross of California, Camarillo, CA

**ABSTRACT (HEADER 1)**

You received a request to extract/download multiple databases on to a PC from mainframe tapes to do an analysis. Now how do you proceed, especially if you wish to avoid submitting many jobs and multiple logins to the mainframe? SAS PC has built in functions: Proc Upload and Download, which can be used in addition to macros to transfer multiple databases between PC and Mainframe, but how about databases that exists on tape? Using SAS PC you can submit a collection of JCL jobs to query and extract data from tapes and use PROC download to download databases onto your PC and continue with your analysis.

**INTRODUCTION (HEADER 1)**

We frequently receive requests to do analysis on databases that exists on Mainframe Tapes. Before, we may had to wait for our mainframe colleague to download databases onto a network drive or do the extract/download ourselves even though we would rather avoid working with mainframe. With the SAS features like PROC Upload, Download and FSEDIT, you can have an entry screen for the user to enter criteria and use the built in procedures to upload the entered information, create and submit the JCL jobs and download the result. The first step is configuring tcptso.scr file to login to mainframe. Then you need to have a program, which creates the JCL job on the mainframe with the given criteria, saves and submits the JCL job to the queue. After the tape jobs run, you can use last part of the program and download the databases to your PC.

**CONNECTION TO MAINFRAME**

SAS has a PROC Upload and Download built into the system to be able to transfer data between two different systems using TCP/IP protocol. To be able to use those features, the first step is to create an autoexec.sas file with following codes:

```
filename rlink 'tcptso.scr';
%let rmthost= "IP address of Mainframe";
options nocenter comamid=tcp remote=rmthost
tbufsize=32768;
```

This file runs every time you start SAS. The above codes also can be included in your SAS program if you have multiple systems that you will remote connect to. In addition you also need to modify the SAS Connect script (tcptso.scr) to connect to Mainframe using TCP/IP protocol. You have to ask your mainframe administrators for information about your mainframe setup.

**INPUT CRITERIA**

Once you have setup your connection to mainframe, you need to query against multiple databases and get certain records based on your analysis criteria. First step is to create an entry database, screen at the beginning of your program for the user to enter all the needed information. You can do this by using a datastep and the following code:

```
Proc fsedit data=Inputlib.Input;
  Var starttape endtape begindate enddate;
Run;
```

These dates/values will be used to create macro variables on the mainframe. If you are the only user and do not like to have an entry screen, then you can setup your variables with their corresponding values in a datastep or setup your macro variables with the criteria values once you log on to the mainframe.

**SIGNON AND UPLOAD DATABASE**

After getting all needed information to setup your program, you have to signon and upload your database on to mainframe. You signon to mainframe by using the following 2 line of codes:

```
Signon;
Rsubmit;
```

Anything after Rsubmit will be executed and submitted on mainframe. Now you can upload your input database onto the mainframe and create macro variables to be used in your program. You upload the database by using the following codes:

```
Proc upload data=inputlib.input out=input;
Run;
```

**DEFINING YOUR INPUT DATASET**

Now that you have the input file uploaded and macro variables created, If you have several consecutive files that you would like to extract, then you can setup a macro, which uses your macro variable (starttape, endtape) to generate code. This code will define your input databases at the beginning of your JCL program. For example you can use the following code if you have multiple consecutive files:

```
%macro defineindatabases;
  %do i=%starttape. %to %endtape.;
    put "//IN&i. DD
      DSN=Mainframepath.filename&i.,DISP=SHR,";
    if &i. gt %starttape. then
      put "//UNIT=AFF=IN&starttape.,
        FREE=CLOSE";
    else
      put "//UNIT=CART,FREE=CLOSE";
  %end;
%mend;
```

This macro will be called later and it will create JCL codes, which will be used to define input databases in the JCL program.

**GENERATING CODE BASED ON YOUR CRITERIA**

The next step is to use starttape, endtape, startdate and enddate macro variables to generate the proper code for the SAS part of your JCL job. For example you can use the following code:

```
%macro SaspartofJCLprogram;
  %do i=%starttape. %to %endtape.;
    put "DATA OUT.ADHOC&I. ";
    put "SET IN&I..ADHOC";
    put "format fromdt mmdyy10.>";
    put "if fromdt ge '&begindate.'d and
      fromdt le '&enddate.'d";
    PUT "RUN.>";
  %end;
%mend;
```

Like the previous macro, this will be called later to create the SAS part of your JCL program.

**CREATING AND SUBMITTING JCL JOB**

The last step is to create your JCL program, save and submit it to the mainframe by using TSO commands. The following program uses the "file and put commands" to create and save statements that are needed for the JCL job to run in a file. This part will call the previous defined macro to generate codes for the input databases and SAS part of the JCL program:

```

data _null_;
  file "YourJCLProgramPath.Name"
  noprint notitles;
  Put "//JOBNAME JOB 65348000,'YOUR
      INIT',CLASS=4,NOTIFY=LOGIN,";
  Put "//MSGCLASS=Z,REGION=0K";
  Put "/*JOBPARM R=163,COPIES=1";
  Put "//STEP1 EXEC SAS";
  %Defineindatabases;
  Put "//OUT DD
DSN=Outdatabasepath,DISP=(,CATLG,DELETE),";
  Put "//
  UNIT=(TESTDATA,25),SPACE=(TRK,(9000,9000)
    ,RLSE)";
  Put "//SYSIN DD *";
  Put "OPTIONS OBS=MAX;";
  %SaspartofJCLprogram;
run;

```

Now that your JCL program is created and stored in a directory, you can submit the job by adding the following command to the bottom of your program and then logout from mainframe:

```

Tso submit (file path & Name);
Endrsubmit;
Signoff;

```

### DOWNLOADING TO PC

Now that the databases have been created on the mainframe, how do you download them to your PC? Previously, you had to create transport files and download each one individually using an FTP program, but now you can start downloading a series of files by entering needed information into your data entry screen. You can create your data entry screen at the beginning of your download program by using `datastep` and `PROC FSEDIT`.

```

data lib.download;
format outpath pcpath $50.;
format starttape endtape 2.;
label starttape='Starting tape #:'
  endtape='Ending tape #:'
  outpath='Database Path:'
  pcpath='PC Path:';
starttape=.;
endtape=.;
outpath='';
pcpath='';
run;
proc fsedit label data=lib.download ;
  var starttape endtape outpath pcpath ;
run;

```

After gathering all the needed information, you can define your PC Directory by using the above database and call `symput` command.

```

data _null_;
set lib.download;
  call symput ('PCPATH',trim(left(pcpath)));
run;
libname pc "&pcpath.";

```

Now you are ready to login and upload your database to the mainframe.

```

signon tbufsize=4096;
rsubmit;

proc upload data=lib.download out=temp;
run;

```

Next step is to create your Macro variables for starttape, endtape and outpath so you can use it in a Macro with `PROC download` to download the consecutive databases to your PC.

```

data _null_;
set temp;
call symput ('starttape',trim(left(starttape)));
call symput ('endtape',trim(left(endtape)));
call symput ('outdatabase',trim(left(outpath)));
run;

%macro part3;
  %do i=&starttape. %to &endtape.;
    libname out "&outdatabase.";
    proc download data=out.adhoc&I.
      out=pc.adhoc&I.;
      run;
  %end;
%mend;

%PART3;

```

Congratulations, you are done with extracting, downloading your database and can go on with your project.

### CONCLUSION

In this day and age it may be hard to find SAS analysts who have experience in the desired field and additionally know mainframe job control language (JCL). Also, upon hiring a new staff member, the pressing business needs may supercede the desire for extensive operating system training. The SAS features highlighted here can leverage existing PC skills to quickly become productive with data that reside on the mainframe. Initially, some assistance may be needed from a mainframe expert, but after the system is set up the PC user is empowered to proceed quickly with minimal assistance. Key areas where help may be initially needed is the customization of the login script for the specific mainframe site. Once this is set up, the reading of the mainframe tapes, the writing to disk, and the ultimate download to PC should proceed smoothly.

### ACKNOWLEDGMENTS

Thanks to Darius Jatulis in helping me to write some segments of this paper and Alexis Neal, my mainframe colleague.

### CONTACT INFORMATION

Contact the author at:

|                |  |
|----------------|--|
| Author Name    | Fereydoun J. Foroudian   |
| Company        | Blue Cross of California   |
| Address        | 18557 Plummer Street #24   |
| City state ZIP | Northridge, CA, 91324  |
| Work Phone:    | (805) 384-7697   |
| Fax:           |  |
| Email:         | <a href="mailto:fereydoun.foroudian@wellpoint.com">fereydoun.foroudian@wellpoint.com</a> |
| Web:           | <a href="http://www.ecs.csun.edu/~forudia">http://www.ecs.csun.edu/~forudia</a>          |

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration. Other brand and product names are trademarks of their respective companies.