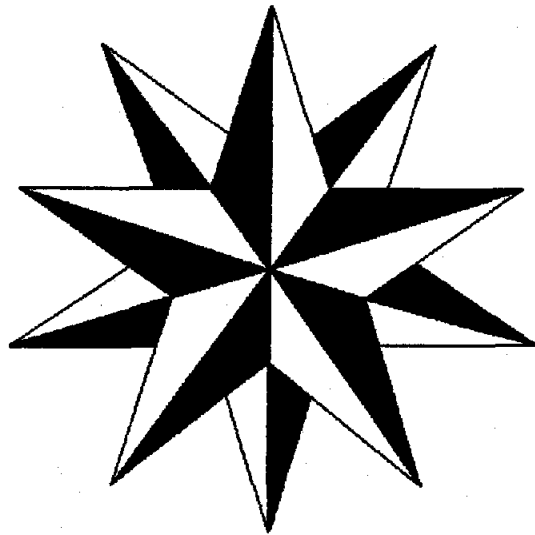


USING PROC UPLOAD/DOWNLOAD TO TRANSFER FILES BETWEEN THE PC AND MAINFRAME



**Elizabeth Hiller
3M IS&DP Statistical Consulting**

USING PROC UPLOAD/DOWNLOAD

TO TRANSFER FILES BETWEEN THE PC AND MAINFRAME

Elizabeth Hiller
3M IS&DP Statistical Consulting

ABSTRACT

There are times, when computing with the personal (micro) computer, that communication to the mainframe (host) computer is beneficial. Some examples are time saving when working with large datasets; maintaining, from the PC, files that reside on the host; and testing code on the PC before submitting it to the host session. For whatever reasons, the SAS System provides UPLOAD and DOWNLOAD procedures (*SAS Procedures Guide, Release 6.03 Edition, Chapters 18 and 35*). These procedures allow transfer of text files, binary files, and SAS datasets between the PC and host computers.

MICRO-TO-HOST LINK

First, establish a link between the micro and host computers. This requires:

- a supported communications card or asynchronous modem on the PC
- SAS Release 6.02 or later on the PC
- SAS Release 5.16 or later on the host
- a SCRIPT file, containing statements that define and control communication links between computers.

Within the SASLINK directory, there are three script files,

- TSO.SCR connects a PC to a host under TSO
- CMS.SCR connects a PC to a host under CMS
- VMS.SCR connects a PC to a host under VMS

(*SAS Guide to the Micro-To-Host Link, Chapters 8, 9, 10, 11, and 12*)

Once requirements are met, run your terminal emulation program, and log on to the host computer. Return to your PC environment, leaving an active session on the host, and invoke the SAS System. Submit (F10 or SUBMIT on the Program Editor's command line) the filename statement, to assign the fileref RLINK for the appropriate script file, and an OPTIONS statement to define your PC's communications (PC3270E, IRMA, or ASYNC), see figure 1. These two statements can be in the AUTOEXEC.SAS file to automatically submit when SAS is invoked. To do this, INCLUDE your AUTOEXEC.SAS file into the Program Editor. Type these two statements into the file, and issue the FILE command on the command line. You will need to end your SAS session, and restart SAS after the AUTOEXEC.SAS file has been changed, in order for the statements to automatically submit.

You are ready to issue the SIGNON command from any of the command lines. The log generated represents both PC and host log. The two logs can be distinguished by letter type. PC log is represented in lower case characters, while host log is represented in upper case characters.

RSUBMIT

Once your link has been established, you need to define your host file and PC directory with librefs. The host libref uses X ALLOC, with a remote submit (Alt+F10, or RSUB entered on the command line), see figure 2. (This file must be defined as direct access or your transfer will not be successful). The remote submit directs processing to the

mainframe's central processing unit. This is advantageous for executing lengthy programs from the PC. The PC libref is defined with the submit of a LIBNAME statement (e.g. LIBNAME PC 'C:\(directory)');, locally, see figure 3.

UPLOADING DATA SETS

Issue the RSUBMIT command to execute PROC UPLOAD. The DATA= option will define your source data set. The .SSD extension assigned to a SAS created data set need not be specified. If the DATA= option is not used, the last SAS data set created on the PC is used. The target data set is defined with the OUT= option, see figure 4. If this is not used, your data set will be transferred into WORK.DATAN, where n is 1 for the first such data set, 2 for the second, and so on.

UPLOADING FILES

Issue the RSUBMIT command to execute PROC UPLOAD. The INFILE= option identifies an external PC file that you want to upload to the host system. This can be done within the INFILE statement, see figure 5, or by using the FILENAME statement, see figure 6. The OUTFILE= option specifies where the external file is written. You cannot transfer a SAS data set to an external file, or transfer an external file to a SAS data set.

DOWNLOADING DATA SETS

Issue the RSUBMIT command to execute PROC DOWNLOAD. Again, the DATA= option will define your source data set;

and if this is not used, the last SAS data set created on the host system is used. The target data set is defined with the OUT= option, see figure 7. If this is not used, your data set will be transferred into WORK.DATAN, where n is 1 for the first such data set, 2 for the second, and so on.

DOWNLOADING FILES

Issue the RSUBMIT command to execute PROC DOWNLOAD. The INFILE= option identifies the fileref of an external file that you want to download from the host system to the PC. The OUTFILE= option specifies the PC file which you want a downloaded external file written, see figure 8.

PROBLEMS

In the first attempt to establish a link between computers, the log message stated my session on the host was not active. After a number of changes in the script file to define my equipment specifications, my link was successful. This led to another problem in the allocation of my dataset on the mainframe. The log message stated my dataset could not be found. This was due to the definition of the dataset not being direct access. After these discrepancies were adjusted, my UPLOADS and DOWNLOADS were successful.

SAS[®] is a registered trademark of SAS Institute Inc., Cary, NC, USA. >

```
COMMAND => SUBMIT
```

```
1 FILENAME RLINK 'C:\SAS\SASLINK\TSO.SCR';  
2 OPTIONS REMOTE = ASYNC;
```

The *TSO* script file, for the IBM mainframe connection, was chosen for this example.

Figure 1

```
COMMAND => RSUBMIT
```

```
1 X ALLOC FI(HOST) DA('A.B.C');
```

Where *HOST* defines the host libref, and *A.B.C* defines the data set on the mainframe.

Figure 2

```
COMMAND => SUBMIT
```

```
1 LIBNAME PC 'C:\SAS\DEF';
```

Where *PC* defines the micro libref, and *C:\SAS\DEF* defines the directory containing the file to be transferred.

Figure 3

```
COMMAND => RSUBMIT
```

```
1 PROC UPLOAD DATA = PC.UPTEST  
OUT = HOST.UPTEST;  
2 RUN;
```

Where *UPTEST* is the data set being transferred to the host.

Figure 4

```
COMMAND => RSUBMIT
```

```
1 PROC UPLOAD INFILE = 'C:\SAS\DEF\FILETEST.PGM'  
OUTFILE = HOST.FILETEST;  
2 RUN;
```

Where *FILETEST.PGM* is the name of the file being transferred, from the directory *C:\SAS\DEF*.

Figure 5

COMMAND => SUBMIT

```
1 FILENAME PC 'C:\SAS\DEF\FILETEST.PGM';
```

COMMAND => RSUBMIT

```
1 PROC UPLOAD INFILE = PC
      OUTFILE = HOST.FILETEST;
2 RUN;
```

Where *FILETEST.PGM* is the name of the file being transferred, from the directory C:\SAS\DEF.

Figure 6

COMMAND => RSUBMIT

```
1 PROC DOWNLOAD DATA = HOST.DOWNTEST
      OUT = PC.DOWNTEST;
2 RUN;
```

Where *DOWNTEST* is the name of the data set being transferred from A.B.C, to the PC.

Figure 7

COMMAND => SUBMIT

```
1 FILENAME HOSTF 'C.D.E(MAINFILE)'; RUN;
```

COMMAND => RSUBMIT

```
1 PROC DOWNLOAD INFILE = HOSTF
      OUTFILE = PC.MAINFILE;
2 RUN;
```

Where *MAINFILE* is the name of the file, within the C.D.E directory, to be transferred.

Figure 8