

From MVS to Unix, a SAS/AF Migration Tool

Phil Busby, Live Data Systems Inc., RTP, NC
 Arturo Barrios, University of North Carolina, Chapel Hill, NC
 Sally Muller, University of North Carolina, Chapel Hill, NC

Abstract

This paper is intended for application developers who want to provide users with an easy and intuitive interface for deleting, browsing, and moving many different kinds of files from an IBM® mainframe to the UNIX environment. The system described, called the TERMINATOR is a SAS/AF® application. It presents IBM TSO users with a list of all of their files (or subset if specified) and as much helpful information as can be obtained from the operating system about each file. The user indicates an action for each file (delete, browse, or move) and when the user says "Go!" the TERMINATOR springs into action.

Introduction

The University of North Carolina Office of Information Technology's IBM mainframe has provided most of the campus computing cycles for nearly 25 years. Now we are shifting from a paradigm of a single host platform (the IBM mainframe) to a distributed system. As a first step, we will migrate users from the IBM mainframe to a UNIX-based Convex® supercomputer. The Convex, however, will serve only as a staging platform until we evolve to the new computing paradigm under which the network and the desktop are as important as any central machine.

The Challenge

The first task, is to move over 8,500 userids and 250,000 files from the IBM mainframe to the Convex. Some of the complications which we face are described below:

Since our IBM mainframe is dedicated to research and education (administrative computing is done in another department), the majority of software applications run on the IBM are for statistical and scientific purposes. For many years the vast majority of jobs run have been SAS System programs, with

Fortran jobs a distant second. Other applications run are SPSS®, SUDAAN®, PL/1, and many other less commonly known applications. This means that we have many different types of files on the mainframe diskpacks and tapes. Not only do we have CA1 tapes ("square" tapes) but also tape reels ("round" tapes).

Never having engaged in a move of this size before, we significantly underestimated the amount of disk space (staging area) and the amount of network bandwidth that would be required.

In addition to these complexities, the structure of MVS (the predominate operating system on the mainframe) files seems specifically designed to thwart UNIX systems. We found innumerable problems when trying to convert the IBM data so that it might be imported into our UNIX platform.

The Plan

Nevertheless, we set out to design the TERMINATOR to provide our TSO IBM users with an easy and intuitive interface for deleting, browsing, and moving their many different kinds of files from the IBM to the Convex. It was our plan that a primary function of the TERMINATOR would be to allow you to clean-up your files before moving them. We felt this would serve both our interests and it is imperative in conserving our scarcest of resources -- network bandwidth. Given the pervasive use of the SAS System in our organization, it was an easy choice to select SAS/AF for developing our front-end and SAS/CONNECT® for our back-end (i.e moving of files).

The Front-End

You invoke the TERMINATOR under MVS with a CLIST called TSOTERM. The CLIST allocates temporary files and calls the SAS system with the option INITSTMT= to go directly into the SAS/AF application. The TERMINATOR does as much as it can for you so keystrokes are minimized.

The first screen displayed (Figure 1) has your TSO USERID filled in as the default stem, or prefix for file names. It has a field for entry of some other prefix for files; for example, if you want just a subset of your files, you could enter USERID.SUBSET in the stem field, and TERMINATOR would display only those files beginning with the prefix USERID.SUBSET. Since it may take several seconds to build the list of files, TERMINATOR does not begin this work until you put the cursor on the OK push-button and press Return.

Figure 1

```

+-----+
| Command ==>                                     |
|                                                  |
| The University of North Carolina at Chapel Hill  |
|   File Transfer and Conversion System           |
|         TSO Terminator                          |
|                                                  |
| Enter mainframe userid where files are:         |
|                                                  |
|        UWALLY                                  |
|                                                  |
| Enter stem or prefix for file names (blank for  |
|        userid):                               |
|        _____                             |
|                                                  |
| ( OK )      ( EXIT )      ( HELP )            |
|                                                  |
| NOTE: Generating file list. Please wait.        |
+-----+
  
```

The file prefix is then passed to the host operating system using the SCL SYSTEM function. TERMINATOR builds a LISTCAT command to request the list of files:

```

cmd = "listcat level (" || trim(fname) || ")
outfile(tempfile);
rc = system(cmd);
  
```

The LISTCAT command stuffs the resulting list of files into the file referenced by the fileref TEMPFIL, which TERMINATOR opens and reads. TERMINATOR can find out about your files in at least three ways:

1. The SCL FINFO function can give a little information about the files, but not much. In particular, the Last-Accessed-Date, Create-Date, and

Size of the files are not available.

2. The JFCB and DSCB options on the INFILE statement give more information, but there are two drawbacks: TERMINATOR would have to submit DATA step code to use the INFILE statement, and the INFILE statement opens the files, changing the Last-Accessed-Date to today's date.

3. The TSO DSAT command is the preferred method to get file attribute information. The TERMINATOR calls TSO with the SYSTEM function for each file, and gets back a file containing the attributes of the specified file:

```

cmd = "dsat " || trim(fname) ||
      " hardcopy(sizefile) alloc lastref pds";
rc = system(cmd);
  
```

The DSAT command gives you back the file's size in tracks, Create-Date, Last-Accessed-Date, blocksize, logical record length, record format, and data set organization. Using these last four (blksize, recl, recfm, and dsorg), the system determines what the most likely kind of file it is, for you. It calls this field the "Probable File Type." Looking at the file name and its Probable File Type, you can easily remember the SPSS, BMDP, or SAS files you created many years ago that should now be deleted or moved to the new computer system.

To determine the "Probable File Type," we trolled far and near to collect all possible DCBs for every type of file that we know exists or was created on the IBM (see Appendix 1 for DCBs of many types of files). This was not an easy task. For example SAS Version 5.18 format libraries have exactly the same DCBs as MVS system load modules; SPSS and SAS system files both have different DCBs as a function of their release date; and both SPSS and BMDP portable files are simple ASCII files. For this reason, you can change the probable file type if so desired -- assuming that if anyone knows the file type, it ought to be you, the owner.

On the second screen (Figure 2) the TERMINATOR displays an extended table of all of the files in the catalog that you requested. Migrated files have less information, since you may not want to wait while they are being recalled. A proposed enhancement to the TERMINATOR is a restricted program to access detailed information for migrated files without recalling them.

On the left side of each column is a 1-character action field for you to specify Delete, Move, or Browse the file. You can process the files one at a time, or enter an action code for multiple files, and the TERMINATOR will process them all at once. The alternative of using a selection list with cursor positioning to indicate the selection of files was rejected because the system must run on many terminal types, and not all of them have the reverse video of selected fields.

Figure 2

```

Command ==>
Enter D next to file to delete:

( OK ) (RETURN)      Type Cr-date LA-date  Size VOLSER

- UWALLY.ACAD          WYLB 17MAY95 18JUL95  1  UDATAA
- UWALLY.ADLER2.PRG    WYLB 17MAY95 18JUL95  1  UDATHH
- UWALLY.BMDP.SYS      BMDP 10JUL95 10JUL95  2  UDATNN
- UWALLY.BMDP.SYS2     BMDP 10JUL95 10JUL95  2  UDATCC
- UWALLY.BMDP.XPT      BMDX 10JUL95 10JUL95  2  UDATEE
- UWALLY.CALLDSA.SRC   UNK  24MAY95 23JUN95 15  UDATDD
- UWALLY.CLIST         PDS  12APR95 18JUL95  4  UDATQQ
- UWALLY.CONVEK.TXT    PDS  09MAY95 13JUL95  3  UDATJJ
- UWALLY.DELFILE.SRC  UNK  30MAY95 23JUN95 15  UDATDD
- UWALLY.DPROMPT.SCR  UNK  30MAY95 18JUL95 15  UDATHH
- UWALLY.DPROMPT.SRC  UNK  30MAY95 23JUN95 15  UDATDD
- UWALLY.DSATOUT       UNK  15MAY95 18JUL95  1  UDATHH
- UWALLY.RACF.CLIST    UNK  18JUL95 18JUL95  1  UDATQQ
- UWALLY.FMTDCB.TEXT  UNK  28APR95 18JUL95  2  UDATHH
- UWALLY.FSIZE.SRC    UNK  30MAY95 23JUN95 15  UDATDD

```

After you have typed D action code(s) for the file(s) you want to delete, you press OK to start the deletion process. The TERMINATOR displays a requester window (Figure 3) for each file to be deleted, asking you for confirmation of the delete. If you reply Yes, the file is deleted at once; if No, it is skipped; if Quit, the entire delete operation is halted for all files; or if Run, the rest of the files are processed without confirmation. The system gives as much help as possible to the careful and cautious user, but also allows an expert to go with a minimum of intervention.

The operating system is actually called to do the file deletes: either DEL for on-line files or HDEL for migrated files is the command sent to the system. For Moves, a TSO batch job is submitted to run late at night. The Browse function invokes the appropriate system to display the file, depending on the Probable File Type.

Figure 3

```

Command ==>

( OK ) (RETURN)      Type Cr-date LA-date  Size VOLSER

- UWALLY.IRSS.MAINDAT8.SASDAT SAS09MAY95 18JUL95 152 UDATHH
- UWALLY +-----+ UDATSS
- UWALLY          UDATAA
- UWALLY          Delete this file? UDATGG
- UWALLY          UDATAA
- UWALLY          UWALLY.LOG.NEWDATA1 UDATAA
- UWALLY          UDATHH
- UWALLY          UDATAA
- UWALLY          ( YES ) ( NO ) ( QUIT ) ( RUN ) UDATHH
- UWALLY          UDATAA
- UWALLY          UDATBB
- D UWALLY          UDATRR
- UWALLY          UDATAA
- UWALLY +-----+ UDATAA
- UWALLY          UDATDD
- UWALLY.LOG.PORTDAT2 BMDX 09JUN95 18JUL95 1 UDATDD

```

The TERMINATOR keeps track of the number of files deleted or moved, and reports the count back to you on the message line when you press End and return to the initial screen. As shown in the following figure.

Figure 4

```

Command ==>
NOTE: 5 files were deleted.

( OK ) (RETURN)      Type Cr-date LA-date  Size VOLSER

- UWALLY.IRSS.MAINDAT8.SASDAT SAS 09MAY95 18JUL95 152 UDATHH
- UWALLY.ISPF.ISPPROF      PDS 07DEC93 18JUL95 1 UDATSS
- UWALLY.JCL                GONE _____
- UWALLY.JFCB.SAS          GONE _____
- UWALLY.JSE1              GONE _____
- UWALLY.LOG.BMDP          UNK 09JUN95 18JUL95 1 UDATAA
- UWALLY.LOG.BMDP1         UNK 09JUN95 18JUL95 1 UDATHH
- UWALLY.LOG.BMDP2         GONE _____
- UWALLY.LOG.DATA          UNK 09JUN95 18JUL95 1 UDATHH
- UWALLY.LOG.DATA2         UNK 09JUN95 18JUL95 1 UDATAA
- UWALLY.LOG.MISC          UNK 28APR95 13JUL95 1 UDATBB
- UWALLY.LOG.NEWDATA1      GONE _____
- UWALLY.LOG.NEWDATA2      BMDP 09JUN95 18JUL95 1 UDATAA
- UWALLY.LOG.PORTDAT1      BMDX 09JUN95 18JUL95 1 UDATAA
- UWALLY.LOG.PORTDAT2      BMDX 09JUN95 18JUL95 1 UDATDD

```

THE BACK-END

On our campus, SAS/CONNECT has always taken a backseat to using PROC CPORT or PROC COPY and creating a SAS "transport" file and FTPing it. In the beginning, we even had to convince people within our own organization that SAS/CONNECT was superior to FTP for moving large numbers of files. However, our success using SAS/CONNECT to move files from the IBM to the Convex, coupled with the fact that SAS/CONNECT supports TCP/IP, our primary method of communication on campus, has resulted in a new interest in SAS/CONNECT.

As seen in Table 1, SAS/CONNECT can move almost every file type that we know of on our system. This includes files containing both ASCII and EBCDIC data. This is a major advantage over FTP since we have a significant number of these "mixed" files.

Another major advantage of SAS/CONNECT is that with few exceptions SAS files do not have to be converted but simply can be moved exactly as they are. In addition, SAS/CONNECT converts the SAS files from EBCDIC to ASCII automatically and converts the SAS files to the host version (e.g 6.08 that runs on MVS to 6.09 that runs on the Convex) so that when they "arrive" on the UNIX platform, they are ready to use.

The only advantage of FTP over SAS/CONNECT is that when moving PDS (partitioned data set). With SAS/CONNECT, if the PDS has a LRECL greater than 80 then the records will be truncated.

Notice that SAS format libraries must be converted first before SAS/CONNECT can move them (and this is true for FTP also.) For examples of code to accomplish this task see "More Information" at the end of this paper.

Table 1. Comparison of SAS/CONNECT vs. FTP

Characteristic	SAS/CONNECT	TRANSPORT FILE
FILE TYPES MOVE	* raw (including those with both ASCII and EBCDIC data)	* raw (but not those with both ASCII and EBCDIC data)
	* SAS files	* SAS files MUST BE CONVERTED
	* SPSS portable	* SPSS portable
	* BMDP portable	* BMDP portable
	* PDS (if LRECL <=80)	* PDS
	* Wylbur uncompressed files	* Wylbur uncompressed files
DISK CAPABILITY	Can move anything on disk.	Can move anything on disk.
TAPE CAPABILITY	Can move DIRECTLY from tape	Cannot move directly with current OS from tape. File first must be copied to disk.
ERROR CHECKING	Dir. of Networking Systems says SAS/CONNECT is cleanest file transfer he has seen	Minimum error checking. FTP can have problems with large files, we have seen large files crash with FTP for no apparent reason.
PROGRAMMING STEPS	A single SAS program	3 programs: *Create transport file *FTP batch program *Convert back to SAS
RESOURCES	Requires only temporary work space	Requires disk space for transport file
CPU TIME	50.22 sec	*14.36 sec program to convert SAS to transport * 7.19 sec to FTP *59.42 program to convert back to SAS 80.97 sec TOTAL
AUDIT TRAIL	SAS Log includes number of bytes to be moved and number of bytes moved as well as number of obs and number of variables.	Apparently an audit trail can be obtained but this is not automatic and we currently have not found how to acquire.

Tape Capabilities

Perhaps the deciding advantage of SAS/CONNECT over FTP is the former's ability to move files directly from tape to the UNIX environment. In our environment FTP can only move a file that is on online disk. This is a triple-disadvantage for us because not only we are short of disk space, but also because it requires more programmer intervention (and so a greater opportunity for errors) and because it requires more time. The only shortcoming of

SAS/CONNECT and SAS tape files is that when the SAS tape file has more than one member (i.e. data set), you must move each member individually with SAS/CONNECT. This is not true if the SAS library is on disk, but currently there is no provision within the SAS System for moving the entire SAS library if it resides on tape. Fortunately, it seems that many of our users allow only one member per SAS library and this is a straightforward SAS/CONNECT move. For examples of code to accomplish this task see "More Information" at the end of this paper.

Of course, dealing with tapes in the UNIX environment is never easy. One of our most interesting experiences happened when trying to move a tape file with VB (variable blocked) records and a LRECL longer than 256 bytes. For a way to circumvent this problem see the SAS Usage Note (V6-Download-7608) available through the online SAS Notes facility of the SAS System.

Linking to TERMINATOR Front-end

Once we decided to use SAS/CONNECT, we wanted to link it with the TERMINATOR front-end to produce a full-blown application that would delete, browse, and move files. Unfortunately time constraints and a major shift in our migration strategy (from in-house to outsourcing) prevented us from ever making this link and we never "automated" our programs. It would be relatively straightforward to do this. A SAS code-generator program could read (i.e. parse) the record that the TERMINATOR front-end produces containing the Probable File and Type. As the program read the Probable File Type, it could write the appropriate SAS statements for moving that file type (e.g. the correct PROC UPLOAD statements, etc.)

Conclusion

Migrating from an IBM mainframe to UNIX can be a daunting and difficult experience. We found salvation in the fact that the SAS System is the same on all platforms, however, the SAS System cannot totally insulate you from the operating system.

SAS/AF proved to be a good tool for our front-end but SCL cannot obtain all the information needed and we had to search for MVS system techniques to acquire the information. Using SAS/CONNECT to move files, while far superior to FTP, still has a number of significant problems. The first of which is

that on the version of SAS available under Convex-OS, none of the enhancements to SAS/CONNECT are supported. In addition, due to the SAS/CONNECT overhead, and also to the state of our campus network, transfer speeds (as measured by bytes per second) are very slow.

Despite these limitations though, we find that our users are very satisfied with the procedure. This results from their files arriving intact; in other words, very few files moved have any problems once in the UNIX environment. Using our procedure means that you can specify the UNIX directory that you want the file to go in and then as soon as the file arrives there you can use it without making any changes and with confidence in the data integrity.

Trademarks

SAS, SAS/AF, and SAS/CONNECT are registered trademarks or trademarks of SAS Institute Inc. In the USA and other countries.

IBM is a registered trademark of International Business machines Corporation.

Convex is a registered trademark of Convex, INC.

References

SAS Institute Inc. (1989), SAS/AF Software: Usage and Reference, Version 6, First Edition, Cary, NC: SAS Institute Inc.

SAS Institute Inc. (1991), SAS Screen Control Language: Usage, Version 6, first Edition, Cary, NC: SAS Institute Inc.

SAS Institute Inc. (1994), SAS Screen Control Language: Reference, Version 6, Second Edition, Cary, NC: SAS Institute Inc.

SAS Institute Inc. (1994), SAS Software: Changes and enhancements, Release 6.10, Cary, NC: SAS Institute

SAS Institute Inc. (1994), SAS/CONNECT Software: Usage and Reference, Version 6, Second Edition, Cary, NC: SAS Institute Inc.

For More Information

Sample programs for a number of migration related task are available on Internet World Wide Web information server. If you have a Web browser you can access through the following URL:
<http://salsa.oit.unc.edu/TERMINATOR.html>

List of available programs:

DISK2UNIX -- Well documented SAS Program for moving disk file (SAS System File) from MVS to the Convex using SAS/CONNECT in batch mode.

FMTLIB5 -- SAS program for converting a SAS format library to a file that can be move either by SAS/CONNECT or FTP.

TAPE2UNIX -- SAS Program for moving tape files (SAS System files both release 5 and 6 and raw files) from MVS to the Convex using SAS/CONNECT in batch mode.

SPSS2UNIX -- SPSS program for converting a SPSS system file to a SPSS portable file and moving it from MVS to the Convex using batch FTP.

UNSCR -- The SAS/CONNECT script that we used (TCP/IP) with our batch SAS/CONNECT programs to move files from MVS to the Convex.

BATCH -- General instructions for submitting either a SAS batch job or an FTP batch job from MVS to UNIX.

Authors

Phil Busby may be reach via Email at Livedata@MERCURY.INTERPATH.NET, or at Live Data Systems, INC., P.O. Box 13144, Research Triangle Park, NC 27709. Phone: (919) 544-7279.

Arturo Barrios may be reached via Email at Arturo_Barrios@UNC.EDU on the Internet, or at Office of Information Technology, CB#3460, Wilson Library, UNC, Chapel Hill, NC 27599-3460. Phone: (919) 966-1533 and Fax: (919) 962-2654.

Sally Muller may be reach via Email at Sally_Muller@UNC.EDU on the Internet, or at Office of Information Technology, CB# 3460, Wilson Library, UNC, Chapel Hill, NC 27599-3460. Phone:(919) 962-5278 and Fax: (919) 962-2654.

APPENDIX I -- DATASET CONTROL BLOCK (DCB) ATTRIBUTES FOR MVS FILES

SAS--1979

	DISK FILE ATTRIBUTES	TAPE FILE ATTRIBUTES
DSORG	DA	PS
LRECL	32756	32756
RECFM	U	U
BLKSIZE	32760	32760

SAS--1982

	DISK FILE ATTRIBUTES	TAPE FILE ATTRIBUTES
DSORG	DA	PS
LRECL	{32756 (BLKSIZE-4)}	32756 (BLKSIZE-4)
RECFM	U	U
BLKSIZE	{32760 (max permitted for device)}	{32760 (max for device)}

SAS--1984

	DISK FILE
DSORG	DA
LRECL	{32756 (BLKSIZE-4)}
RECFM	U
BLKSIZE	32760 (max permitted for device)

SAS--Version 5.18

	DISK FILE	TAPE FILE	FORMAT LIBRARY
DSORG	DA	PS	PO
LRECL	{32756 (BLKSIZE-4)}	{32756 (BLKSIZE-4)}	0
RECFM	U	U	U
BLKSIZE	32760 (max for device)	32760 (max for device)	will vary

SAS--Version 6

	DATA FILE & FORMAT LIBRARY	TAPE FILE & TAPE FORMAT ON DISK
DSORG	PS	PS
LRECL	multiple of 512	32756 (BLKSIZE-4)
RECFM	FS	U
BLKSIZE	same as LRECL	32760 (max permitted for device)

BMDP

	Disk File Attributes			Tape File
	DATA FILE	PORTABLE FILE	OUTPUT (LISTING FILE)	TAPE FILE
DSORG	PS	PS	PS	PS
LRECL	32768	80	133	DOESN'T MATTER
RECFM	VS	F	UA	DOESN'T MATTER
BLKSIZE	80	80	133	DOESN'T MATTER

SPSS

	OLD SPSS (maroon manual, pre SPSS-X)	SYSTEM FILES (Version 1 through 4.1 and SPSS-X)	TRANSPORT FILES
DSORG	PS	PS	
LRECL	2008/20008	1024	
RECFM	VBS/VB	FB	FB
BLKSIZE	2012	3072 or 4096 (any even multiple of 1024)	3600 (any even multiple of 80)

WYLBUR

	WYLBUR SEQUENTIAL DISK FILE	WYLBUR PDS		
		PO	PO	PO
DSORG	PS			
LRECL	6233	3156	6360	6233
RECFM	U	U	U	U
BLKSIZE	6233	3156	6360	6233