

any screen I/O as the result of a DOS command or external program. Source code for CHK2COPY is provided in Attachment 1. CHK2COPY was written and compiled with Microsoft® C 5.1.

SAS/AF SCL Code

For this paper I prepared a stand alone application that copies or moves DOS files. The SAS/AF program (Attachment 2) contains SCL code that gets the file information and target disk specifications from the user, runs the CHK2COPY program, and then checks to see if the file 'OK2COPY.IT' has been created.

The figures below depict the AF screen for this application with messages displayed for the start of the system (Figure 3) and when there is insufficient space on the target disk (Figure 4).

Figure 3

```

AF
Command ==>
Fill in Fields below to move or copy a DOS file.

Copy or Move DOS File to a Floppy Disk

Drive, Path, and Name of File to Copy or Move: _____
Destination Floppy Disk Drive:  _ (A or B)
Move File? (Delete from source after Copy?)  _ (Y/N)

Fill in the fields above and press ENTER to execute COPY / MOVE.
Press F10 to Exit.

```

Figure 4

```

AF
Command ==>
Disk in Drive is full. Insert a new formatted disk.

Copy or Move DOS File to a Floppy Disk

Drive, Path, and Name of File to Copy or Move: C:\MYDIR\LARGE.FIL
Destination Floppy Disk Drive:  B (A or B)
Move File? (Delete from source after Copy?)  N (Y/N)

Fill in the fields above and press ENTER to execute COPY / MOVE.
Press F10 to Exit.

```

This paper won't get into the specific details of the entire SCL program code for this application, but will highlight the code specific to running the CHK2COPY program and checking if 'OK2COPY.IT' was created. Looking at the related sections of the SCL code,

```
cmdstring="X 'CHK2COPY "|| targetdrv ||" "|| filename ||"";
```

builds the command string passed to DOS, via the 'X' command, that executes the CHK2COPY program. The variable targetdrv is the target drive ID for the destination disk and filename is the name of the DOS file to be copied. (The variables filename, targetdrv, and moveit are assigned top down, respectively to the fields on the above displayed AF screens.)

```
call execcmd(cmdstring);
refresh;
```

The execcmd function above executes the specified command when the SCL program returns control to the procedure. At that point the command string is submitted to SAS for execution. The refresh statement updates screen values but also provides the necessary break so the command will be submitted to execute.

```
call filename('AOK','OK2COPY.IT');
RC=exist('AOK');
if RC=0 then do;
  _msg_ = msg{7};
return;
end;
```

The filename function associates the SAS fileref AOK with the external file OK2COPY.IT. RC is the code returned by the exist function to indicate if an external file exists. If RC is equal to 1 the signal file exists and the program proceeds with the file copy. If RC is equal to 0 the file does not exist and msg{7}, informing the user that there was not enough free space on the target disk, is assigned to _msg_ and the program returns to the top of the main program section.

Conclusions

This method and the associated program will work well to provide the needed function. There are also several other uses for this type of approach, i.e. using an external program to provide a function that SAS doesn't. Some of these include (1) Searching directories for a specific file, (2) finding the date and time a file was last modified, and (3) resetting specific file attributes to maintain or create hidden files for security purposes. In addition to this "signal file" concept, information or data can be written to a file that would then be read by a SAS data step to determine next steps, list files, signal success, etc.

Acknowledgements

I would like to thank Tim Mueller, Karen Bickel, and all of the other wonderful people that I work with for their technical assistance and encouragement.

References

- SAS Institute Inc., (1988), SAS Language Guide for Personal Computers, Release 6.03 Edition, Cary, NC: SAS Institute Inc.
- SAS Institute Inc., (1988); SAS Procedures Guide, Release 6.03 Edition, Cary, NC: SAS Institute Inc.
- SAS Institute Inc., (1987), SAS/AF Guide for Personal Computers, Version 6 Edition, Cary, NC: SAS Institute Inc.
- SAS and SAS/AF are registered trademarks of SAS Institute Inc., Cary, NC, USA.
- Microsoft is a registered trademark of Microsoft Corporation

For More Information:

If anyone has questions, comments, suggestions, or would like a copy of CHK2COPY the author can be reached at:

Craig Bickel

Lasley & Hopkins Inc.

4500 Cooper Road, Suite 201

Cincinnati, OH 45242

phone: (513) 791-3376

fax: (513) 791-8206

Attachment 1

```
#include <fcntl.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <io.h>
#include <stdio.h>
#include <stdlib.h>
#include <dos.h>
/* CHK2COPY.C */
main(argc,argv)
int argc;
char *argv[];
{
    FILE *infile;
    FILE *AOKfile;
    unsigned drive, maxdrives;
    unsigned long total_space, free_space, bytes_per_cluster;
    char *copydrv;
    int numclosed;
    struct diskfree_d dinfo;
    int fh, result;
    struct stat buf;
    if (remove("OK2COPY.IT") != 0)
        printf("OK2COPY.IT does not exist. No file to delete.\n");
    else
        printf("Previous verification file OK2COPY.IT has been deleted.\n");
    strcpy(copydrv, argv[1]);
    strcpy(copydrv);
    drive = 'copydrv' - 'A';
    drive++;
    if (_dos_getdiskfree (drive, &dinfo) != 0) {
        printf("Error in _dos_getdiskfree\n");
        exit(0);
    }
    bytes_per_cluster = dinfo.sectors_per_cluster * dinfo.bytes_per_sector;
    total_space = dinfo.total_clusters * bytes_per_cluster;
    free_space = dinfo.available_clusters * bytes_per_cluster;
    printf("%ld bytes free out of %ld bytes total for target disk %s.\n",
        free_space, total_space, copydrv);
    if ((fh = open(argv[2], O_RDONLY)) == -1) {
        printf("Error opening file: %s.\n", argv[2]);
        exit(0);
    }
    result = lstat(fh, &buf);
    if (result != 0) {
        printf("Bad file handle.\n");
        exit(0);
    }
    printf("Size of %s = %ld bytes.\n", argv[2], buf.st_size);
    if (buf.st_size > free_space) {
        if ((AOKfile = fopen("OK2COPY.IT", "a+")) == NULL) {
            printf("Error opening file.\n");
            exit(0);
        }
        else {
            printf("%ld bytes available on disk %s after copy. OK to proceed with copy.\n",
                free_space - buf.st_size, copydrv);
            printf("Creating OK2COPY.IT as verification of sufficient space available.\n");
            numclosed = fcloseall();
            printf("%lu file(s) closed.\n", numclosed);
            exit(0);
        }
    }
    if (buf.st_size > free_space) {
        printf("Insufficient disk space on drive %s for file.\n", copydrv);
        printf("Verification file not created.\n");
        exit(0);
    }
}
```

Attachment 2

/* SCL Code to Copy or Move a DOS File to a floppy disk */

```
array msg [ 7 ] $ 62
/* Fill in fields below to move or copy a DOS file.
   * A and B are the only valid floppy drive ID's. Re-enter.
   * Enter the drive ID for the target drive.
   * Y and N are the only valid responses for Move File. Re-enter.
   * Delete file to be copied from source drive? Enter Y or N.
   * Source File does not exist. Check your spelling.
   * Disk in Drive is full. Insert a new formatted disk.*/
length word $ 6;

init:
control enter;
cursor filename;
msg = msg[ 1 ];
return;

main:
if ( _status_ = 'E' or _status_ = 'C' ) then goto term;

if ( modified( targdrv ) or targdrv = _BLANK_ ) then do;
if ( targdrv = 'A' and targdrv = 'B' ) then do;
msg = msg[ 2 ];
return;
end;
end;
else do;
msg = msg[ 3 ];
return;
end;

if ( modified( movet ) or movet = _BLANK_ ) then do;
if ( movet = 'Y' and movet = 'N' ) then do;
msg = msg[ 4 ];
return;
end;
end;
else do;
msg = msg[ 5 ];
return;
end;

call filename( 'TEST', filename );
if ( !exist( 'TEST' ) ) then do;
msg = msg[ 6 ];
return;
end;

cmdstring = "X CHK2COPY " || targdrv || " " || filename || " ";
call execcmd( cmdstring );
refresh;
call filename( 'AOK', 'OK2COPY.IT' );
rc = exist( 'AOK' );
if rc = 0 then do;
msg = msg[ 7 ];
return;
end;

cmdstring = "X copy " || filename || " " || targdrv || " ";
call execcmd( cmdstring );
refresh;
word = copied;
if movet = 'Y' then do;
cmdstring = "X del " || filename || " ";
call execcmd( cmdstring );
refresh;
word = moved;
end;
msg = "DOS file " || filename || " has been successfully " || word || " ";
return;

term:
_status_ = 'H';
return;
```