UNITED STATES DEPARTMENT OF INTERIOR

GEOLOGICAL SURVEY

# Modifications of the IBM Personal Computer Asynchronous Communications Support Programs For Use With the Multics

By

John O. Kork[1]

Open-File Report 83-492



1. Lakewood, Colorado

1983

CONTENTS

# MODIFICATIONS OF THE IBM PERSONAL COMPUTER ASYNCHRONOUS COMMUNICATIONS SUPPORT PROGRAMS FOR USE WITH THE MULTICS

By JOHN O. KORK

ABSTRACT

Version 1.00 of the Asynchronous Communications Support supplied with the IBM Personal Computer must be modified to be used for communications with Multics. Version 2.00 can be used as supplied, but error checking and screen printing capabilities can be added by using modifications very similar to those required for Version 1.00. This paper describes and lists required programs on Multics and appropriate modifications to both Versions 1.00 and 2.00 of the programs supplied by IBM.

## INTRODUCTION

The IBM Personal Computer Asynchronous Communications Support, Version 1.00, implemented by the Cambridge Scientific Center contains a program, TERMINAL.BAS, written in the BASIC language to be used for communications with other computers. This program can be used as supplied for communications with other IBM computers, but for communication with Multics modifications must be made to the program. This paper describes appropriate modifications to the IBM program and lists Multics programs which provide the capability for uploading and downloading ASCII files.

There are two possible approaches to the problem of modifying a program such as TERMINAL.BAS. One is to write completely new modules which will cause the program to handle Multics output exactly in the form it is sent to a terminal, and the other is to make very slight modifications to the TERMINAL.BAS program and write drivers for Multics which will make Multics appear to be one of the computers or editors for which the program was originally written. I have chosen the second approach because it is easier to implement, even though it is more expensive to run.

There are two sets of modifications to Version 1.00 of the TERMINAL program described in this paper. The first makes minimal changes to the program and modifies or appends only 26 lines of code. The second set modifies or appends 119 lines of code, adds the option for printing the information being transferred on the terminal screen as uploading or downloading is taking place, and also provides for error checking of the data being transferred. Version 2.00 of the Asynchronous Communications Support communicates with Multics as it is supplied, but because the modifications required to add screen printing and error checking options to that version are similar to those required for Version 1.00, they are included also.

## PROGRAMS ON MULTICS

There are three programs that must be available on the Multics system in order for the modifications of the IBM program TERMINAL.BAS to work. Presently these three programs are available in the directory

>udd>Math_ora>JKork>ibm

with names toibm, fromibm and EDIT.ec on the Denver Multics system. The first two programs are written in the PL/1 programming language and the third program is a command level macro (exec_com). The PL/1 programs have multiple entry points, and the compiled segments must be given multiple names by use of the "add_name" command. The names "send" and "sendchk" must be added to the compiled program segment "toibm", and the names "recv" and "recvchk" must be added to the compiled segment "fromibm". The programs are listed in Appendices A, B, and C for future reference in case the copies in the above directory are removed for any reason.

In order to use these programs a user must set up an abbreviation by entering

```
.ab EDIT ec >udd>Math_ora>JKork>ibm>EDIT
```

on Multics. This must be done only once as the profile segment containing abbreviations is stored for future use even though the user logs off the computer. A file transfer is initiated when the IBM PC sends the keyword EDIT at the beginning of a line. The abbreviation then causes the exec_com, "EDIT", to execute, and the exec_com transfers control to the appropriate PL/1 program. The PL/1 program simulates elementary responses of the IBM TSO editor.

## IBM PC PROGRAM MODIFICATIONS

To adapt Version 1.00 of the Asynchronous Communications Support for communication with Multics, modifications must be made to the program TERMINAL.BAS. Version 2.00 is organized differently and requires that modifications be made to two programs, TERMINAL.BAS and TERMINIT.BAS. The modifications can most easily be made using the editing capabilities of the BASIC interpreter.

Lists of modifications were generated by transfering the original and the modified programs to Multics and then comparing them with the compare_ascii command. In the lists of changes program A is the original program, and program B is the modified program. The lists are in Appendices D, E, F, and G of this paper.

### Modifications to Version 1.00

The TERMINAL program as supplied is designed to run on a system with one or two floppy disks. The system used by the Branch of Resource Analysis in Denver includes one floppy disk and a hard disk. The modifications of the operating system required by the hard disk caused the location specified in line 115 to be inappropriate. One of the first operations the TERMINAL program does is to load a binary file called RS232INT.BAS into memory, and the location into which this file is loaded is defined by a DEF SEG command in line 115 of the program. If the standard operating system supplied with the IBM PC is being used, then the following modification should not be necessary. Changing program line 115 from

```
115 DEF SEG &HE80
```

to

```
115 DEF SEG &H1700
```

and line 210 from

```
210 IDSEG=&HE80
```

to

```
210 IDSEG &H1700
```

overcomes the addressing difficulty. Note that the original address is transformed to &HE800 and loads the subroutine into the top 4K bytes of a 64K system. The new address is translated to &H17000 and loads the subroutine into the top 4K bytes of a 96K system. If only 64K bytes are available, the system can be used without the hard disk operating system and the addresses in lines 115 and 210 can be left unchanged. In order to use the hard disk, disk D as the default disk, line 117 was changed from

```
117 D$(1)="A:":D$(2)="B:"
```

to

```
117 D$(1)="D:":D$(2)="B:".
```

The modifications to lines 635 and 775 change values of default terminal definition parameters. Code for displaying the additional menus for the screen printing and error check options is contained in lines 866, 867, and 8000 through 8140. Line 1520 provides control to allow a user to exercise these options. The change in line 13165 was required because the original code would not allow downloading of more than one file; the downloading operation would terminate before any lines of the second file were transferred. The inserted lines 13504 and 13606 provide screen printing capability for download and compare operations, and the new code in lines 13850 through 13854 prevents inadvertent destruction of an existing file on the local computer during download operations. The original program did not allow transfer of lines longer than 131 characters, a restriction not relevant for Multics; so the maximum line length was increased to 245 in line 13910. Error checking and screen printing during an upload operation are provided by the changes to lines 14049 through 14054, 14515 through 14518, and 31200 through 31235. Line 14527 was changed to prevent a troublesome transmission to Multics in case the target file for an upload operation was found to exist there already. It was easier to change the code in the TERMINAL program than to allow for the transmission in the programs on Multics. Error checking during a download operation is provided by the changes and additions to lines 14600 through 14608, 16005 through 16008, 16106, and 31000 through 31160. The change to line 16110 was made to prevent the addition of a single blank line at the end of the transferred file.

## Modifications to Version 2.00

Version 2.00 consists of three programs, RS232INT.EXE, TERMINIT.BAS, and TERMINAL.BAS; and changes must be made to the latter two programs to add the error checking option. If no error checking is desired, the programs work quite well without modification.

The changes to the program TERMINIT.BAS change the default terminal parameters and provide part of the code for adding the screen printing and error checking menus. The changes to TERMINAL.BAS are very similar to those required for Version 1.00, but the line numbers of the changes are not the same in some cases. Hence the explanation of the functions of the changes for Version 1.00 will suffice for Version 2.00 also.

## EXECUTING THE MODIFIED VERSION 1

### Establishing Communications with Multics

The program, TERMINAL.BAS, was modified to execute from the hard disk, disk d, and the modified versions of the program were saved on that disk with the names JKTRM.BAS AND JKTRMCHK.BAS. Line 117 was modified to cause the program to search for a file named RS232INT.BAS on disk d, and thus that file was also copied to disk d. If the configuration does not include a disk d, the modified TERMINAL program and the file RS232INT.BAS should be located together on disk a or b. To execute the program the user enters the command BASIC JKTRM or BASIC JKTRMCHK. The program then loads and a menu asks the user to specify the terminal type being used. Choice number 4, "User Specified Full Duplex Term", should be chosen. The option of changing any or all of the terminal specifications is then presented, and at this time the user can also choose to exercise the screen printing option and/or the error checking option. The option to start up as a terminal should be selected, and the message

You are Starting up as a Terminal.

Check Computer or Modem Connection.

will appear on the screen. The 300 baud Multics access line should then be dialed, and the modem should be connected. After the return key on the keyboard is pressed a few times, Multics will respond with a ready message. The login can then proceed as usual. Terminal type settings that should be made on Multics can then be accomplished by using the "stty" command as follows:

```
stty -ttp ascii_crt
stty -modes (^echoplex, ^pl).
```

If the ^pl mode is not specified, Multics will pause after printing 20 lines and print the message EOP at the left margin of the screen. Printing will continue only after the user enters a carriage return. This switch should be disabled for uploading or downloading files, although it may be left in operation for terminal use to prevent output from being moved off the screen by the terminal scrolling operation.

### Uploading, Downloading, and Comparing Files

To cause the function selection menu to appear on the screen, the special function key F2 on the left of the IBM PC keyboard must be pressed. An upload, download, or compare operation can be selected from this menu. During an upload, download or compare operation transmission can be terminated by pressing the special function key F1. If key F1 is pressed during a download or compare function, all files are closed and the TERMINAL program automatically returns to terminal mode. Depressing the F1 key during an upload, however, will leave the Multics file that is receiving the information open, and the TERMINAL program will ask the user to specify whether or not

the uploaded data are to be preserved. The user should answer "Y". The program will then return to terminal mode, at which time a carriage return, the word "SAVE", and the word "END" should be entered in that order.

To end communications with Multics the user should first log off the system and then press function key F2. The function choice should then be to "return to BASIC", after which an "OK" will appear on the screen indicating that the BASIC interpreter is loaded. Entering "SYSTEM" through the keyboard will cause a return to system operation. A BREAK signal can be sent to Multics at any time by pressing the special function key F1.

### What To Do In Case of Trouble

Trouble is most likely to occur when uploading or downloading a file. The user should always know the approximate number of lines to be transmitted so that if the program somehow gets into a loop that is transmitting the same line over and over again, transmission can be stopped. If too many lines seem to be uploading or the line indicator showing the number of the line being transmitted doesn't change for a while, the TERMINAL program can be stopped by pressing the CTRL and BREAK keys simultaneously. An "OK" should then appear on the screen indicating that the program has stopped and the BASIC interpreter is awaiting instructions. To start the program over again the command "RUN" should be entered through the keyboard, at which time the program will execute again from the beginning. Terminal operation should be re-entered and a BREAK signal should be sent to Multics (by pressing the function key F1). Multics files that were left open by the BREAK signal interrupting a file transfer operation can be closed by entering the sequence END, [carriage return], [carriage return] a few times. If the open files need not be preserved or Multics does not respond, another BREAK signal can be sent and then a new_proc can be executed to start over.

The TERMINAL program limits the length of transmitted lines to 245 characters. If an attempt is made to transmit a longer line, the program will stop and print the message "Line too long to upload". The user can then proceed by answering the questions printed on the screen about preserving uploaded data and starting again. The long line should be split into two shorter lines or deleted from the file to be transmitted.

If nothing seems to be working right, a procedure that is certain to terminate the session is to turn off the IBM PC, hang up the phone, and call the Multics operator to be logged off the system through the operator console. If the operator is not notified to log the user off Multics, it is possible that the user's process will be continued and unwanted charges will accrue.

## EXECUTING VERSION 2.00

### Executing the Unmodified Version 2.00

Version 2.00 can be used as it is supplied, but the user must set the terminal parameters the first time the program is used and then save these

parameters in a file for subsequent use. The "User Specified Terminal" should be selected from the terminal selection menu in this case. A Terminal Feature Menu, used for setting the terminal parameters, is then printed on the screen. Listed below is this menu with the proper selections for Multics communications indicated in brackets:

Terminal Feature Menu

1 Line Bit Rate [300]
2 Type of Parity Checking [Even]
3 Number of Stop Bits [Two Bits]
4 XON/XOFF Support [Absent]
5 Line Turnaround Char Sent to Host [CR]
6 Local or Host Character Echoing [Local]
7 First Character to be Deleted [LF]
8 Second Character to be Deleted [None]
9 Third Character to be Deleted [None]
10 Fourth Character to be Deleted [None]
11 Line End Character Sent by Host [CR]
12 Communications Adapter Address [1]
13 Start Up Selected Terminal
14 Save This Terminal Sepcification
15 Return to Terminal Selection Menu

The proper choice for selection number 6, Local or Host Character Echoing, depends on whether the Multics terminal mode "echoplex" is turned on. If Multics is set for "echoplex" then the selection should be host echoing; if Multics is set for "no echoplex", the selection should be local echoing. The terminal specifications should be saved in a file for later use. Then terminal operation should be selected, and Multics dialup and login can the proceed as usual. Terminal type settings on Multics should be made the same way they are for Version 1.00.

The Multics editor, qedx, can be used for uploading files to Multics. The editor should be put into insert mode, and then special function key F2 should be pressed. A prompt will ask for the name of the file to be transmitted, and another prompt will ask whether or not a return character will be sent after each line is transmitted. The response should be "N", and transmission of the file will proceed. When the file has been transmitted, the message "File sending complete" will appear on the screen. The transmitted file is now in the qedx buffer on Multics, and the user should press the enter key to return to terminal operation. Entering backslash f [carriage return], w filename [carriage return], and q [carriage return] will save the file and exit from qedx.

To download a file the command "print filename.ext 1", without a carriage return, should be sent to Multics. Special function key F8 should then be pressed on the IBM keyboard, and the file name of the local file to be saved on the IBM PC should be entered. When the message "You are back as a terminal" appears on the screen a carriage return should be entered, and Multics will transmit the file. The end of transmission will be indicated by Multics sending a "ready" message, at which time function key F8 should be depressed again to save the downloaded file.

## Executing the Modified Version 2.00

To execute the modified Version 2.00, two programs must be executed. The first is a compiled program RS232INT.EXE, which is executed from command level by entering RS232INT via the keyboard. Then the basic program, TERMINAL, must be executed by entering BASIC TERMINAL. The TSO option should be selected from the terminal selection menu, and from this point on the programs then execute the same way Version 1.00 does.

## APPENDIX A

### Program EDIT.ec

```
&command_line off
stty -ttp ascii_crt
stty -modes (11240 ^echoplex ^pl ^erkl ^can)
&if [equal "OLD" &2] &then &goto snd
&if [equal "NONUM" &2] &then &goto rec
&if [equal "DCK" &2] &then &goto sndck
&if [equal "UPCHK" &2] &then &goto upchk
&print "trouble starting load"
&goto out
&label snd
&if [not [exists entry &1]] &then &goto nosnd
>udd>Math_ora>JKork>ibm>send &1
&goto out
&label sndck
&if [not [exists entry &1]] &then &goto nosnd
>udd>Math_ora>JKork>ibm>sendchk &1
&goto out
&label rec
&if [exists entry &1] &then &goto norec
>udd>Math_ora>JKork>ibm>recv &1
&goto out
&label upchk
&if [exists entry &1] &then &goto norec
>udd>Math_ora>JKork>ibm>recvchk &1
&goto out
&label nosnd
&print NO HOST FILE: &1
&print READY
&goto out
&label norec
&print EDIT
&goto out
&label out
stty -modes (can erkl)
&quit
```

## APPENDIX B

### Program toibm.pll

```
toibm: proc (filnm);
dcl (i, last, chksum, decl, itmp) fixed bin;
dcl  strl char (1);
dcl  str2 char (10) varying;
dcl  filnm char (*);
dcl  fildsc char (200) varying;
dcl  check bit (1);
dcl (sysin, sysprint, ibmtrfl, ibmtrf) file;
dcl (varln, instr) char (250) varying;
dcl  char builtin;
dcl (error, endfile) condition;

send:     entry (filnm);
          check = "0"b;
          go to start;
sendchk:  entry (filnm);
          check = "1"b;
          go to start;

start:    fildsc = "vfile_ "||filnm;
          open file (ibmtrf) title (fildsc) stream input;
          open file (ibmtrfl) environment (stringvalue) record input
               title ("record_stream_ ibmtrf");
          on error go to closeout;
          on endfile (ibmtrfl) begin;
               last = 1;
               go to endat;
          end;

          last = 0;
          put skip list ("READY");
          put skip;
          itmp = 0;
          do while ("1"b);
getit:         get list (instr);
               if substr (instr, 1, 5) = "AGAIN" then do;
                    put edit (varln) (skip, a);
                    put skip;
               end;
               itmp = itmp+1;
               if substr (instr, 1, 3) = "END" then do;
                    put skip list ("READY");
                    put skip;
                    go to closeout;
               end;
```

```
               if last = 1 then go to endat;
               if substr (instr, 1, 4) = "DOWN" then do;
                    read file (ibmtrfl) into (varln);
                    if check then do;
                         if length (varln) = 0 then varln = "00000";
                         else do;
                              chksum = 0;
                              do i = 1 to length (varln);
                                   strl = substr (varln, i, 1);
                                   decl = index (collate (), strl)-1;
                                   chksum = chksum+decl;
                              end;
                              chksum = 100000 + mod (chksum, 32767);
                              str2 = char (chksum);
                              str2 = substr (str2, length (str2)-4, 5);
                              varln = varln||str2;
                         end;
                    end;
                    put edit (varln) (skip, a);
                    put skip;
               end;
endat:         if last = 1 then do;
                    varln = "END OF DATA,";
                    if check then varln = varln||"00754";
                    put edit (varln) (skip, a);
                    put skip (2);
               end;
          end;
closeout: close file (ibmtrfl);
          close file (ibmtrf);
     end;
```

## APPENDIX C

### Program fromibm.pll

```
fromibm: proc (filnm);
dcl (sysin, sysprint, inplin, ibmtrf) file;
dcl (i, ifirst, chksum, decl) fixed bin;
dcl  strl char (1);
dcl (str5, str5c) char (10) varying;
dcl  filnm char (*);
dcl  fildsc char (200) varying;
dcl  check bit (1);
dcl  varln char (250) varying;
dcl (error, endpage) condition;

recv:     entry (filnm);
          check = "0"b;
          goto start;
recvchk:  entry (filnm);
          check = "1"b;
          goto start;

start:    open file (inplin) environment (stringvalue) record input
               title ("record_stream_ user_input");
          fildsc = "vfile_ "||filnm;
          open file (ibmtrf) title (fildsc) stream print output;
          on error go to closeout;
          on endpage (ibmtrf) begin;
          end;

          put skip list ("DATA");
          put edit ("INPUT") (skip, a (5));
          put skip;
          ifirst = 1;
          do while ("1"b);
               read file (inplin) into (varln);
               if length (varln) = 0 then goto finish;
               if check then do;
                    if length (varln) < 5 then goto finish;
                    if length (varln) = 5 then do;
                         if varln ^= "00000" then do;
                              put list ("AGAIN");
                              put skip;
                         end;
                         else do;
                              put file (ibmtrf) skip;
                              put skip;
                         end;
                    end;
```

```
                    else do;
                         str5 = substr (varln, length (varln)-4, 5);
                         varln = substr (varln, 1, length (varln)-5);
                         chksum = 0;
                         do i = 1 to length (varln);
                              str1 = substr (varln, i, 1);
                              dec1 = index (collate (), str1)-1;
                              chksum = chksum+dec1;
                         end;
                         chksum = 100000+mod (chksum, 32767);
                         str5c = char (chksum);
                         str5c = substr (str5c, length (str5c)-4, 5);
                         if str5 = str5c then do;
                              if varln = " " then varln = "";
                              if ifirst = 0 then put file (ibmtrf)
                                   edit (varln) (skip, a);
                              else do;
                                   ifirst = 0;
                                   put file (ibmtrf) edit (varln) (a);
                              end;
                              put skip;
                         end;
                         else do;
                              put list ("AGAIN");
                              put skip;
                         end;
                    end;
               end;
               else do;
                    if varln = " " then varln = "";
                    if ifirst = 0 then put file (ibmtrf)
                         edit (varln) (skip, a);
                    else do;
                         ifirst = 0;
                         put file (ibmtrf) edit (varln) (a);
                    end;
                    put skip;
               end;
          end;
finish:
          put skip (2);
          read file (inplin) into (varln);
          if substr (varln, 1, 4) ^= "SAVE" then
               put skip list ("trouble closing");
          put edit ("EDIT") (skip, a (4));
          put skip;
          read file (inplin) into (varln);
          put edit ("READY") (skip, a (5));
          put skip;
          go to closeout;
```

```
errors:
          put skip list ("error termination");
closeout:
          put file (ibmtrf) skip;
          close file (ibmtrf);
          close file (inplin);
     end;
```

## APPENDIX D

### Minimal Modifications for Version 1.00

```
Inserted in B:
B1        1 'MODIFIED TO JKTRM.BAS BY JOHN O. KORK, USGS, 2-17-83
Preceding:
A1        5 GOSUB 50:GOTO 100 'RUN entry


A10       115 DEF SEG =&HE80
A11       117 D$(1)="A:":D$(2)="B:"
Changed by B to:
B11       115 DEF SEG =&H1700
B12       117 D$(1)="D:":D$(2)="A:"


A27       200 CLEAR ,&H9000
A28       205 DEFINT I-M
A29       210 IDSEG=&HE80
Changed by B to:
B28       200 CLEAR ,&HA000
B29       205 DEFINT I-M
B30       210 IDSEG=&H1700


A85       775 DATA 4,4,1,0,2,1,1,0,1,1,1,1,0,0,1,0,0,0,0,0
Changed by B to:
B86       775 DATA 4,3,1,0,2,2,1,0,1,3,1,1,0,2,1,0,0,0,0,0


A574      13165 PRINT "Host File Accessed":LCNT=0:LOCT=0
Changed by B to:
B575      13165 PRINT "Host File Accessed":LCNT=0:LOCT=0:MEN=FALSE


A679      13850 ON ERROR GOTO 13860:OPEN "o",2,Y$
Changed by B to:
B680      13850 ON ERROR GOTO 13853:OPEN "I",2,Y$
B681      13851 CLOSE 2:M$="LOCAL FILE ALREADY EXISTS.":ON ERROR GOTO 0:RETURN
B682      13853 RESUME 13854
B683      13854 ON ERROR GOTO 13860:OPEN "O",2,Y$


A688      13910 ON ERROR GOTO 0:IF LEN(RT$)<131 OR ICOMP OR IT=3 THEN RETURN
Changed by B to:
B692      13910 ON ERROR GOTO 0:IF LEN(RT$)<245 OR ICOMP OR IT=3 THEN RETURN


A787      14527 IF E$="EDIT" THEN M$="Host Data Set already Exists.":GO
SUB 15100:RETURN
```

```
Changed by B to:
B791      14527 IF E$="EDIT" THEN M$="Host Data Set already Exists.":RETURN
A873      16110 IF LEFT$(B$,12)="END OF DATA," THEN MEN=TRUE:GOSUB 14880
Changed by B to:
B877      16110 IF LEFT$(B$,12)="END OF DATA," THEN MEN=TRUE:GOSUB 14880:B$=""

Comparison finished: 9 differences, 26 lines.
```

## APPENDIX E

### Modifications for Version 1.00 to Add Screen Printing and Error Checking Options

```
A10       115 DEF SEG =&HE80
A11       117 D$(1)="A:":D$(2)="B:"
Changed by B to:
B11       115 DEF SEG =&H1700
B12       117 D$(1)="D:":D$(2)="A:"


A27       200 CLEAR ,&H9000
A28       205 DEFINT I-M
A29       210 IDSEG=&HE80
Changed by B to:
B28       200 CLEAR ,&HA000
B29       205 DEFINT I-M
B30       210 IDSEG=&H1700


A70       635 DATA 1,1,1,0,0,1,1,0,1,1,1,1,0,0,1,0,0,0,0,0
Changed by B to:
B71       635 DATA 1,1,1,0,0,1,1,0,1,1,1,1,0,0,1,0,0,0,1,1


A85       775 DATA 4,4,1,0,2,1,1,0,1,1,1,1,0,0,1,0,0,0,0,0
Changed by B to:
B86       775 DATA 4,3,2,0,2,2,1,0,1,3,1,1,0,2,1,0,0,0,1,1


Inserted in B:
B109      866 NA$(19)="Screen Printing of Upload/Download"
B110      867 NA$(20)="Checksum Error Checking"
Preceding:
A108      895 CCLEN = 5 'no ctl chars


A180      1520 ON IP GOSUB 6000,6200,6400,9000,6800,7000,7200,7400,7500,
7600,7625,7660,9000,9000,7800,9000,9000,9000,9000,9000
Changed by B to:
B183      1520 ON IP GOSUB 6000,6200,6400,9000,6800,7000,7200,7400,7500,
7600,7625,7660,9000,9000,7800,9000,9000,9000,8000,8100


Inserted in B:
B557      8000 'Menu for screen printing of upload/download
B558      8005 TITH$="Screen Printing of Upload/Download"
B559      8010 TITLE$="Screen printing option"
B560      8015 IMAX=2
B561      8020 C$(1)="Screen printing disabled"
```

```
B562      8025 C$(2)="Screen printing enabled"
B563      8030 GOSUB 20200
B564      8035 GOSUB 20100
B565      8040 RETURN
B566      8100 'Menu for checksum option
B567      8105 TITH$="Checksum Error Checking"
B568      8110 TITLE$="Checksum option"
B569      8115 IMAX=2
B570      8120 C$(1)="No error checking"
B571      8125 C$(2)="Checksum error checking/retransmission 20 times"
B572      8130 GOSUB 20200
B573      8135 GOSUB 20100
B574      8140 RETURN
Preceding:
A554      9000 PRINT "illegal option reached. ip= ";IP


A574      13165 PRINT "Host File Accessed":LCNT=0:LOCT=0
Changed by B to:
B595      13165 PRINT "Host File Accessed":LCNT=0:LOCT=0:MEN=FALSE


Inserted in B:
B674      13504 IF MPARM(19)=2 THEN PRINT "HOST : ";B$:PRINT "LOCAL: ";RT$
Preceding:
A653      13505 IF RT$=B$ THEN RETURN


Inserted in B:
B685      13606 IF MPARM(19)=2 THEN PRINT B$
Preceding:
A663      13610 ON ERROR GOTO 0


A679      13850 ON ERROR GOTO 13860:OPEN "o",2,Y$
Changed by B to:
B702      13850 ON ERROR GOTO 13853:OPEN "I",2,Y$
B703      13851 CLOSE 2:M$="LOCAL FILE ALREADY EXISTS.":ON ERROR GOTO 0:RETURN
B704      13853 RESUME 13854
B705      13854 ON ERROR GOTO 13860:OPEN "O",2,Y$


A688      13910 ON ERROR GOTO 0:IF LEN(RT$)<131 OR ICOMP OR IT=3 THEN RETURN
Changed by B to:
B714      13910 ON ERROR GOTO 0:IF LEN(RT$)<245 OR ICOMP OR IT=3 THEN RETURN


A714      14050 F$=RT$:IF F$="" THEN F$=" "
A715      14052 GOSUB 13300
Changed by B to:
B740      14049 IF MPARM(20)=2 THEN GOSUB 31200
B741      14050 F$=RT$:IF F$="" THEN F$=" "
```

```
B742      14051 IF MPARM(19) <> 2 GOTO 14054
B743      14052 IF MPARM(20)=1 THEN PRINT F$:GOTO 14054
B744      14053 IF LEN(F$)>5 THEN PRINT LEFT$(F$,LEN(F$)-5) ELSE PRINT
B745      14054 GOSUB 13300


A783      14515 GOSUB 15000:F$="EDIT "+F$+" NONUM":GOSUB 14900
Changed by B to:
B813      14515 GOSUB 15000
B814      14516 IF MPARM(20)=1 THEN F$="EDIT "+F$+" NONUM":GOSUB 14900:GOTO 14520
B815      14517 IF MPARM(20)=2 THEN F$="EDIT "+F$+" UPCHK":GOSUB 14900:GOTO 14520
B816      14518 PRINT "ERROR IN CHECKSUM CODING":RETURN


A787      14527 IF E$="EDIT" THEN M$="Host Data Set already Exists.":GO
SUB 15100:RETURN
Changed by B to:
B820      14527 IF E$="EDIT" THEN M$="Host Data Set already Exists.":RETURN


A793      14600 GOSUB 14900:GOSUB 14880
A794      14605 IF B$=LE$ OR B$= " " THEN M$="":RETURN
Changed by B to:
B826      14600 IF MPARM(20)=1 THEN GOSUB 14900:GOSUB 14880:GOTO 14608
B827      14601 KNC%=0
B828      14602 GOSUB 14900:GOSUB 14880:KNC%=KNC%+1
B829      14603 IF LEFT$(B$,5)<>"AGAIN" GOTO 14608
B830      14604 IF KNC%>20 THEN IE2=TRUE:PRINT "CHECKSUM ERROR":RETURN
B831      14605 IF B$=LE$ OR B$= " " THEN M$="":RETURN
B832      14606 LCNT=LCNT-1:TA$="CK":GOSUB 13300:TA$="UP"
B833      14607 PRINT F$+" again ",KNC%:GOTO 14602
B834      14608 IF B$=LE$ OR B$= " " THEN M$="":RETURN


A853      16005 GOSUB 15000:F$="EDIT "+F$+" OLD":GOSUB 14900
Changed by B to:
B893      16005 GOSUB 15000
B894      16006 IF MPARM(20)=1 THEN F$="EDIT "+F$+" OLD":GOSUB 14900:GOTO 16010
B895      16007 IF MPARM(20)=2 THEN F$="EDIT "+F$+" DCK":GOSUB 14900:GOTO 16010
B896      16008 PRINT "ERROR IN CHECKSUM CODING":IE2=TRUE:RETURN


A873      16110 IF LEFT$(B$,12)="END OF DATA," THEN MEN=TRUE:GOSUB 14880
Changed by B to:
B916      16106 IF MPARM(20)=2 THEN GOSUB 31000
B917      16110 IF LEFT$(B$,12)="END OF DATA," THEN MEN=TRUE:GOSUB 14880:B$=""


Inserted in B:
B1116     31000 'Download checksum routine
B1117     31010 KNC%=0
B1118     31020 IF LEN(B$)<5 GOTO 31110
```

```
B1119      31030 BV$=B$
B1120      31034 IF LEN(B$)>5 GOTO 31040
B1121      31035 IF LEFT$(B$,5)="00000" THEN B$=" ":RETURN
B1122      31036 GOTO 31110
B1123      31040 B$=LEFT$(B$,LEN(B$)-5)
B1124      31050 KCK%=-32767
B1125      31060 FOR KI%=1 TO LEN(B$)
B1126      31070 KCK%=KCK%+ASC(MID$(B$,KI%,1))
B1127      31071 IF KCK%>0 THEN KCK%=KCK%-32767
B1128      31080 NEXT KI%
B1129      31090 IF KCK%<=0 THEN KCK%=KCK%+32767
B1130      31100 IF KCK%=VAL(RIGHT$(BV$,5)) THEN RETURN
B1131      31110 IF KNC%>20 GOTO 31150
B1132      31115 KNC%=KNC%+1
B1133      31120 F$="AGAIN":GOSUB 14900:GOSUB 14880:GOSUB 14880
B1134      31121 PRINT B$;KCK%,"again",KNC%
B1135      31125 TB$=TA$
B1136      31130 LCNT=LCNT-1:TA$="CK":GOSUB 13300:TA$=TB$
B1137      31140 GOTO 31020
B1138      31150 LOCATE 23,1,1:PRINT "CHECKSUM ERROR: ";LCNT:LOCATE 24,1,1
B1139      31160 IE2=TRUE:RETURN
B1140      31200 'Upload checksum routine
B1141      31205 KCK%=-32767
B1142      31210 FOR KI%=1 TO LEN(RT$)
B1143      31215 KCK%=KCK%+ASC(MID$(RT$,KI%,1))
B1144      31216 IF KCK%>0 THEN KCK%=KCK%-32767
B1145      31220 NEXT KI%
B1146      31225 IF KCK%<=0 THEN KCK%=KCK%+32767
B1147      31226 KCK$=STR$(KCK%)
B1148      31227 KCK$="000000"+MID$(KCK$,2,LEN(KCK$))
B1149      31230 RT$=RT$+RIGHT$(KCK$,5)
B1150      31235 RETURN
at end

Comparison finished: 20 differences, 119 lines.
```

## APPENDIX F

### Modifications for TERMINAL.BAS, Version 2.00

```
A60       250 ON ERROR GOTO 13700:PRINT2,B$:ON ERROR GOTO 0:RETURN
A61       900 GOSUB 32200:CLS:PRINT "Please wait - program initializing..."
A62       930 ON ERROR GOTO 33000:CHAIN MERGE "TERMINIT",935,ALL
Changed by B to:
B60       250 ON ERROR GOTO 13700:PRINT2,B$
B61       251 IF MPARM(19)=2 THEN PRINT B$
B62       252 ON ERROR GOTO 0:RETURN
B63       900 GOSUB 32200:CLS:PRINT "Please wait - program initializing..."
B64       930 ON ERROR GOTO 33000:CHAIN MERGE "TRMINIT1",935,ALL


A86       1520 ON IP GOSUB 6000,6200,6400,9000,6800,7000,7200,7400,7500,
7600,7625,7660,7680,9000,7800,7850,9000,9000,9000,9000
Changed by B to:
B88       1520 ON IP GOSUB 6000,6200,6400,9000,6800,7000,7200,7400,7500,
7600,7625,7660,7680,9000,7800,7850,9000,9000,8000,8100


Inserted in B:
B353      8000 'Menu for screen printing option of upload/download
B354      8005 TITH$="Screen Printing of Upload/Download"
B355      8010 TITLE$="Screen printing option"
B356      8015 IMAX=2
B357      8020 C$(1)="Screen printing disabled"
B358      8025 C$(2)="Screen printing enabled"
B359      8030 GOSUB 20200:RETURN
B360      8100 'Menu for checksum option
B361      8105 TITH$="Checksum Error Checking"
B362      8110 TITLE$="Checksum option"
B363      8115 IMAX=2
B364      8120 C$(1)="No error checking"
B365      8125 C$(2)="Checksum error checking/retransmission 20 times"
B366      8130 GOSUB 20200:RETURN
Preceding:
A351      9000 PRINT "Invalid option";IP:STOP


A373      13165 PRINT"Host file accessed":LCNT=0:LOCT=0
Changed by B to:
B389      13165 PRINT"Host file accessed":LCNT=0:LOCT=0:MEN=FALSE


Inserted in B:
B453      13504 IF MPARM(19)=2 THEN PRINT "HOST : ";B$:PRINT "LOCAL: ";RT$
Preceding:
A437      13505 J$=STRING$(ABS(LEN(RT$)-LEN(B$)),32):IF LEN(RT$)<=LEN(B$)
THEN IF B$=RT$+J$ THEN RETURN ELSE ELSE IN
```

```
A470      13910 ON ERROR GOTO 0:IF LEN(RT$)<131 OR ICOMP THEN RETURN
Changed by B to:
B487      13910 ON ERROR GOTO 0:IF LEN(RT$)<245 OR ICOMP THEN RETURN
A495      14050 F$=RT$:IF F$="" THEN F$=" "
A496      14052 GOSUB 13300:ON IT GOSUB 14300,14600
Changed by B to:
B512      14049 IF MPARM(20)=2 THEN GOSUB 34200
B513      14050 F$=RT$:IF F$="" THEN F$=" "
B514      14051 IF MPARM(19) <> 2 GOTO 14054
B515      14052 IF MPARM (20)=1 THEN PRINT F$:GOTO 14054
B516      14053 IF LEN(F$)>5 THEN PRINT LEFT$(F$,LEN(F$)-5) ELSE PRINT
B517      14054 GOSUB 13300:ON IT GOSUB 14300,14600


A542      14515 GOSUB 15000:F$="EDIT "+F$+" NONUM":GOSUB 13420:GOSUB 14880
Changed by B to:
B563      14515 GOSUB 15000:F$="EDIT "+F$
B564      14516 IF MPARM(20)=1 THEN F$=F$+" NONUM":GOSUB 13420:GOTO 14520
B565      14517 IF MPARM(20)=2 THEN F$=F$+" UPCHK":GOSUB 13420:GOTO 14520
B566      14518 PRINT "ERROR IN CHECKSUM CODING":RETURN
B567      14520 GOSUB 14880


A545      14527 IF E$="EDIT" THEN M$="Host data set already exists":GO
SUB 15100:RETURN
Changed by B to:
B570      14527 IF E$="EDIT" THEN M$="Host data set already exists":RETURN


A550      14600 GOSUB 13420:GOSUB 14880
A551      14605 IF B$=LE$ OR B$= " " THEN M$="":RETURN
Changed by B to:
B575      14600 IF MPARM(20)=1 THEN GOSUB 13420:GOSUB 14880:GOTO 14608
B576      14601 KNC%=0
B577      14602 GOSUB 13420:GOSUB 14880:KNC%=KNC%+1
B578      14603 IF LEFT$(B$,5)<>"AGAIN" GOTO 14608
B579      14604 IF KNC%>20 THEN IE2=TRUE:PRINT "CHECKSUM ERROR":RETURN
B580      14605 IF B$=LE$ OR B$=" " THEN M$="":RETURN
B581      14606 LCNT=LCNT-1:TA$="CK":GOSUB 13300:TA$="UP"
B582      14607 PRINT F$+" again ",KNC%:GOTO 14602
B583      14608 IF B$=LE$ OR B$= " " THEN M$="":RETURN


A601      16005 GOSUB 15000:F$="EDIT "+F$+" NONUM OLD":GOSUB 13420
Changed by B to:
B633      16005 GOSUB 15000:F$="EDIT "+F$
B634      16006 IF MPARM(20)=1 THEN F$=F$+" OLD":GOSUB 13420:GOTO 16010
B635      16007 IF MPARM(20)=2 THEN F$=F$+" DCK":GOSUB 13420:GOTO 16010
B636      16008 PRINT "ERROR IN CHECKSUM CODING":IE2=TRUE:RETURN
A619      16110 IF LEFT$(B$,12)="END OF DATA," THEN MEN=TRUE:GOSUB 14880
Changed by B to:
B654      16106 IF MPARM(20)=2 THEN GOSUB 34000
B655      16110 IF LEFT$(B$,12)="END OF DATA," THEN MEN=TRUE:GOSUB 14880:B$=""
```

```
Inserted in B:
B854      34000 'Download checksum routine
B855      34010 KNC%=0
B856      34020 IF LEN(B$)<5 GOTO 34110
B857      34030 BV$=B$
B858      34034 IF LEN(B$)>5 GOTO 34040
B859      34035 IF LEFT$(B$,5)="00000" THEN B$=" ":RETURN
B860      34036 GOTO 34110
B861      34040 B$=LEFT$(B$,LEN(B$)-5)
B862      34050 KCK%=-32767
B863      34060 FOR KI%=1 TO LEN(B$)
B864      34070 KCK%=KCK%+ASC(MID$(B$,KI%,1))
B865      34071 IF KCK%>0 THEN KCK%=KCK%-32767
B866      34080 NEXT KI%
B867      34090 IF KCK%<0 THEN KCK%=KCK%+32767
B868      34100 IF KCK%=VAL(RIGHT$(BV$,5)) THEN RETURN
B869      34110 IF KNC%>20 GOTO 34150
B870      34115 KNC%=KNC%+1
B871      34120 F$="AGAIN":GOSUB 13420:GOSUB 14880:GOSUB 14880
B872      34121 PRINT B$;KCK%,"again",KNC%
B873      34125 TB$=TA$
B874      34130 LCNT=LCNT-1:TA$="CK":GOSUB 13300:TA$=TB$
B875      34140 GOTO 34020
B876      34150 LOCATE 23,1,1:PRINT "CHECKSUM ERROR:  ";LCNT:LOCATE 24,1,1
B877      34160 IE2=TRUE:RETURN
B878      34200 'Upload checksum routine
B879      34205 KCK%=-32767
B880      34210 FOR KI%=1 TO LEN(RT$)
B881      34215 KCK%=KCK%+ASC(MID$(RT$,KI%,1))
B882      34216 IF KCK%>0 THEN KCK%=KCK%-32767
B883      34220 NEXT KI%
B884      34225 IF KCK%<0 THEN KCK%=KCK%+32767
B885      34226 KCK$=STR$(KCK%)
B886      34227 KCK$="000000"+MID$(KCK$,2,LEN(KCK$))
B887      34230 RT$=RT$+RIGHT$(KCK$,5)
B888      34235 RETURN
At end

Comparison finished: 13 differences, 99 lines.
```

## APPENDIX G

### Modifications for TERMINIT.BAS, Version 2.00

```
A31       31160 DATA 1,1,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0:'TSO
Changed by B to:
B31       31160 DATA 1,1,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,1,1:'TSO


A39       31320 DATA 4,4,1,0,2,2,6,0,1,2,5,6,10,2,5,1,0,0,0,0:'TSO
Changed by B to:
B39       31320 DATA 4,3,1,0,2,2,1,0,1,3,1,1,0,2,1,1,0,0,1,1:'TSO


Inserted in B:
B62       31487 NA$(19)="Screen Printing of Upload/Download"
B63       31488 NA$(20)="Checksum Error Checking"
Preceding:
A62       31490 DIM JA$(3) 'set special titles


A96       31640 DIM KTD(20),DA$(8,10) 'Default table setup
A97       31644 DATA 1,2,3,0,0,4,5,5,6,7,7,7,7,0,5,8,0,0,0,0
Changed by B to:
B98       31640 DIM KTD(20),DA$(9,10) 'Default table setup
B99       31644 DATA 1,2,3,0,0,4,5,5,6,7,7,7,7,0,5,8,0,0,9,9


Inserted in B:
B112      31676 DA$(9,1)="DISABLED":DA$(9,2)="ENABLED"
Preceding:
A110      31681 BEL$=CHR$(7):LFD$=CHR$(10):ATTN$=CHR$(59) 'Term constants set here

Comparison finished: 5 differences, 11 lines.
```