

# SHARE PROGRAM LIBRARY AGENCY



PROGRAM NUMBER

700001

---

## University of Miami

1365 MEMORIAL DRIVE - CORAL GABLES, FLORIDA  
(305) - 284-6257

# SHARE PROGRAM LIBRARY SUBMITTAL FORM



SPLA

CONTROL NUMBER: 220

SHARE PROGRAM LIBRARY AGENCY  
Triangle Universities Computation Center  
Post Office Box 12076  
Research Triangle Park, North Carolina USA 27709

This form should be completed and submitted with the program package to the SHARE Program Library Agency at the address shown above. Standards and instructions for submitting programs are in the SHARE Reference Manual, Section 6.

- (1) Program Number (to be filled by SPLA) . . . . . 360D-70.0.001
- (2) Title of Program . . . . . HASP OS/VS Workstation Program  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_
- (3) System Type(s) (Machine) . . . . . 3600S, 3700S, VS
- (4) Search Key(s) . . . . . Workstation Batch Terminal Program  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_
- (5) Programming Systems/Languages . . . . . Assembler
- (6) Primary Subject Code . . . . . 70.0
- (7) Minimum System Requirements 116K region or partition of OS or OS/VS
- (8) New (N) or Revision (R) (if revision, show prior Program Number in Item 1) N
- (9) Date of Submittal . . . . . January 3, 1978
- (10) Documentation (number of original pages submitted) . . . . . 21
- (11) Author's Name and Address . . . . . J. D. Snyder  
Westinghouse Tele-Computer Center  
Brinton Road at Parkway East  
Pittsburgh, PA 15221
- (12) Direct Technical Inquiries to Name & Address  
(if different than Author) \_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_
- (13) Submitter's Installation Membership Code . . . . . WEC
- (14) Abstract (should contain sufficient information for a reader to determine the value of the program). Listed on the reverse side of this form are subjects which may serve as a guide for a descriptive abstract.

## DISCLAIMER

Triangle Universities Computation Center (TUCC) serves solely as the distribution agent for contributed programs and does not test or maintain them. They are distributed essentially in the original form submitted by the author. Neither TUCC nor SHARE, INC., makes any warranty, expressed or implied, as to the documentation, function, or performance of the contributed programs.

## SHARE PROGRAM LIBRARY SUBMITTAL FORM

### Subject Guide:

- a. Purpose
- b. Programming Language used
- c. Version and modification level or release number
- d. Field of application
- e. Type of routine (main program, subroutine, etc.)
- f. Specific description of machine requirements

### DISCLAIMER

Triangle Universities Computation Center (TUCC) serves solely as the distribution agent for contributed programs and does not test or maintain them. They are distributed essentially in the original form submitted by the author. Neither TUCC nor SHARE, INC., makes any warranty, expressed or implied, as to the documentation, function, or performance of the contributed programs.

<p>The HASP OS/VS Remote Workstation package allows the user to communicate as a batch terminal to the ASP Version 3.2.2 system at the Westinghouse Tele-Computer Center. The package consists of three programs: the workstation terminal communication program, the reader utility and the writer utility. The reader utility spools to disk all jobs to be transmitted to the ASP system. The terminal communications program sends these jobs, retrieves the output and spools it to disk. The program records all files in a disk directory. The writer utility prints or punches the output files to a hardware device. This utility enables the user to print or punch files by forms or by job. The terminal program runs in one region or partition and is kept active as long as it is necessary to communicate with the ASP system. The writer utility runs in another region or partition and is kept active while the user wants to print or punch files. The reader utility may be invoked at any time from the batch card reader. The terminal program is 16K and the reader and writer utilities are 4K.</p>
(Please attach additional pages if necessary) . . . . . Total pages attached _____

An "Acknowledgement of Assistance" statement must be attached to this Submittal Form.

### Permission to Publish

"I hereby give the SHARE Program Library Agency permission to reprint, reproduce, and distribute this program"

(15) Signature of Submitter and Date John D. Snyder January 3, 1978

(15) Signature of Installation Addressee Charles E. Hurdman

```

* * * * * TAPE KEY * * * * *
FILE1 WORKSTATION COMMUNICATIONS TERMINAL PROGRAM ASSEMBLER SOURCE
      CHANGE DECK
      EBCDIC
      SEQUENCE F055200 THRU F97560000 IN CARD COLUMNS 73-80
      1595 CARDS BLOCKED 20 PER BLOCK
      80 BLOCKS OF 1600 CHARACTERS EACH
      T/M

FILE2 WORKSTATION READER UTILITY ASSEMBLER SOURCE DECK
      EBCDIC
      SEQUENCE 00000000 THRU 01830000 IN CARD COLUMNS 73-80
      258 CARDS BLOCKED 20 PER BLOCK
      13 BLOCKS OF 1600 CHARACTERS EACH
      T/M

FILE3 WORKSTATION WRITER UTILITY ASSEMBLER SOURCE DECK
      EBCDIC
      SEQUENCE 00000000 THRU 027600000 IN CARD COLUMNS 73-80
      418 CARDS BLOCKED 20 PER BLOCK
      21 BLOCKS OF 1600 CHARACTERS EACH
      T/M

FILE4 WORKSTATION COMMUNICATIONS TERMINAL PROGRAM OBJECT DECK
      EBCDIC
      SEQUENCE 000001 THRU 00337 IN CARD COLUMNS 73-80
      337 CARDS BLOCKED 20 PER BLOCK
      17 BLOCKS OF 1600 CHARACTERS EACH
      T/M

FILE5 WORKSTATION READER UTILITY OBJECT DECK
      EBCDIC
      SEQUENCE 0001 THRU 0054 IN CARD COLUMNS 77-80
      54 CARDS BLOCKED 20 PER BLOCK
      3 BLOCKS OF 1600 CHARACTERS EACH
      T/M

FILE6 WORKSTATION WRITER UTILITY OBJECT DECK
      EBCDIC

```

SEQUENCE 0001 THRU 0099 IN CARD COLUMNS 77-80  
 89 CARDS BLOCKED 20 PER BLOCK  
 5 BLOCKS OF 1600 CHARACTERS EACH  
 T/M

FILE7

THIS DOCUMENTATION FILE

EBCDIC

SEQUENCE 00000100 THRU 00722000 IN CARD COLUMNS 73-80  
 722 CARDS BLOCKED 20 PER BLOCK  
 37 BLOCKS OF 1600 CHARACTERS EACH  
 T/M

FILE8

SAMPLE JCL FOR ASSEMBLYING WORKSTATION COMMUNICATIONS TERMINAL  
 PROGRAM

EBCDIC

SEQUENCE 00000100 THRU 00002500 IN CARD COLUMNS 73-80  
 25 CARDS BLOCKED 20 PER BLOCK  
 2 BLOCKS OF 1600 CHARACTERS EACH  
 T/M

THE WESTINGHOUSE HASP OS/VIS TERMINAL WORKSTATION SUPPORT CONSISTS OF THREE PROGRAMS. THESE ARE (1) THE WORKSTATION COMMUNICATIONS PROGRAM, (2) THE INPUT READER UTILITY, AND (3) THE OUTPUT WRITER UTILITY.

THE WORKSTATION COMMUNICATIONS PROGRAM IS AN ASSEMBLER LANGUAGE SOURCE CHANGE DECK TO MEMBER HETPB360 OF DATA SET SYSL.HASPSRC FOR HASP VERSION 3.1. THE READER AND WRITER UTILITIES ARE WESTINGHOUSE-WRITTEN ASSEMBLER LANGUAGE PROGRAMS.

THE COMMUNICATIONS PROGRAM IS A REVISION OF THE HASP REMOTE WORKSTATION TERMINAL PACKAGE PROVIDED BY IBM TO WESTINGHOUSE IN 1972. THIS PROGRAM WAS REWRITTEN AND EXPANDED BY WESTINGHOUSE BECAUSE OF OPERATIONAL DIFFICULTIES WITH THE ORIGINAL VERSION. THE REWRITE HAS BEEN IN USE AT FOUR WESTINGHOUSE DATA CENTERS SINCE 1974.

MANY OF THE ORIGINAL IBM WORKSTATION PROGRAM SOURCE STATEMENTS ARE STILL USED IN THE COMMUNICATIONS PROGRAM SOURCE CHANGE DECK. THESE STATEMENTS ARE MARKED WITH THE IDENTIFYING CHARACTERS "UCC19" IN CARD COLUMNS 67-71. THE ADDITIONAL WESTINGHOUSE CHANGES CONTAIN "WTSC" IN COLUMNS 68-71 OR "W" IN COLUMN 71.

THE CHANGE DECK HAS CONDITIONAL ASSEMBLY STATEMENTS TO GENERATE REMOTE TAPE DEVICE SUPPORT, PLOTTER SUPPORT, DATE AND PASSWORD IN THE SIGNON CARD, AND OTHER OPTIONS. TO INCLUDE THE DESIRED FEATURE SET THE CORRESPONDING ASSEMBLER STATEMENT ON (SETA OR SETC).

WHEN ASSEMBLING THE WORKSTATION COMMUNICATIONS PROGRAM SPECIFY NUMBUFS=7 IN THE OPTIONS DD FILE. A NUMBER OF BUFFERS GREATER THAN 7 MAY CAUSE A PROBLEM.

\*\*\*\*\*  
 \* 1. HASP OS/VVS REMOTE WORKSTATION COMMUNICATIONS PROGRAM \*  
 \*\*\*\*\*

THIS PROGRAM ALLOWS A REMOTE OS/VVS SYSTEM TO COMMUNICATE AS A TERMINAL WITH THE 370/145 ASP VERSION 3.2.2 SYSTEM AT THE WESTINGHOUSE TELE-COMPUTER CENTER. THE WORKSTATION PROGRAM RUNS IN ONE REGION OR PARTITION OF MVT OR MFT.

A COMMUNICATIONS ADAPTER MUST BE ALLOCATED BY UNIT ADDRESS TO THE PROGRAM. THE OPERATOR MUST MAKE THIS ADAPTER UNAVAILABLE TO THE OS/VVS SYSTEM BEFORE HE STARTS THE WORKSTATION PROGRAM.

THE OPERATOR INITIATES THE PROGRAM BY ISSUING THE COMMAND "S WSTATION" AT HIS CONSOLE. AFTER LOADING, THE PROGRAM ISSUES THE FOLLOWING WTO:

HASP OS/VVS REMOTE WORKSTATION READY FOR COMMUNICATIONS  
 THIS INDICATES THAT THE PROGRAM IS READY TO SIGN ON TO THE TCC SYSTEM.  
 THE WORKSTATION TERMINAL PROGRAM ALWAYS HAS AN OUTSTANDING WTO, GIVING THE OPERATOR THE MEANS OF ENTERING CONSOLE COMMANDS.  
 THE WTO MESSAGE SAYS

HASP OS/VVS RMS - ENTER LOCAL COMMAND OR MSG TO ASP  
 A "MESSAGE TO ASP" IS CONSOLE INPUT TO BE TRANSMITTED TO TCC.  
 A "LOCAL COMMAND" IS A CONSOLE ENTRY THAT OPERATES STRICTLY WITHIN THE TERMINAL PROGRAM AND IS NOT TRANSMITTED.

WHEN THE PROGRAM IS INITIALLY LOADED, THE TERMINAL'S PRINTER DEVICE IS INHIBITED AND WILL NOT ACCEPT OUTPUT. THIS IS NECESSARY SO THAT THE OPERATOR CAN DEFINE PRINTER FORMS BEFORE THE TERMINAL PROGRAM RECEIVES ANY PRINT DATA. HE MUST VARY THE PRINTER TO STANDARD FORMS, RESTART THE PRINTER, AND THEN ENABLE THE PRINTER. THIS INVOLVES ENTERING THREE CONSOLE COMMANDS. THE FIRST IS THIS MESSAGE TO ASP:

X PRUT,OUT=PR1,F=FOO1006T

THE OPERATOR RECEIVES THESE RESPONSES FROM ASP:

JOB 9999 IS PRUT , CALLED BY RM027  
 JOB 9999, PRUT TO RM027PR1  
 CTAPE=00 , FORMS=FOO1006T, RLSE, TRAIN=PN  
 JOB 9999, PRUT PURGED

NEXT IS THE RESTART PRINTER COMMAND TO ASP:

R PR1

THE OPERATOR THEN ENTERS THIS LOCAL COMMAND TO ENABLE THE PRINTER:

ENPR

THE TERMINAL WILL NOW BE ABLE TO RECEIVE PRINT DATA.

WHEN THE TERMINAL INITIALLY ESTABLISHES COMMUNICATION WITH THE ASP SYSTEM, THE PROGRAM DISPLAYS THIS MESSAGE ON THE CONSOLE:

ISSUE "X PRUT,OUT=PR1,F=FOO1006T", "R PR1","ENPR"

THIS REMINDS THE OPERATOR THAT HE MUST VARY THE FORMS, RESTART, AND ENABLE THE PRINTER.

CONSOLE MESSAGES RECEIVED FROM ASP BY THE WORKSTATION ARE PADDED WITH ASTERISKS AT THE RIGHT END. THESE ASTERISKS FLAG THE MESSAGES, MAKING THEM OBVIOUS TO THE OPERATOR.

THE OPERATOR SHOULD COMPLETE HIS TERMINAL SESSION BY ENTERING THE LOCAL COMMAND "SIGNOFF". THIS DISCONNECTS HIS TERMINAL FROM THE ASP SYSTEM. THE PROGRAM WILL INDICATE TIMEOUTS ON THE COMMUNICATIONS ADAPTER. AT THIS POINT, ENTER THE LOCAL COMMAND "EOJ" TO TERMINATE THE WORKSTATION PROGRAM AND RETURN CONTROL TO US.

FOUR CONDITIONS CAN CAUSE AN UNTIMELY END OF A WORKSTATION



# TERMINAL SESSION:

1. THE TCC ASP SYSTEM GOES DOWN,
  2. THE TERMINAL OS/VS SYSTEM GOES DOWN,
  3. THE WORKSTATION PROGRAM ABENDS, OR
  4. THE TERMINAL SIMPLY LOSES COMMUNICATIONS WITH ASP.
- IN CASE 1 YOU MUST WAIT FOR TCC'S RESTART BEFORE YOU CAN BEGIN A NEW TERMINAL SESSION. IN CASES 2,3, AND 4 YOU MUST CONTACT THE TCC OPERATOR, HAVE YOUR LINE CANCELLED AND RESTARTED, AND THEN START THE WORKSTATION PROGRAM AGAIN.

IN ALL FOUR OF THESE CONDITIONS IT IS POSSIBLE THAT DATA BEING TRANSMITTED AT THE TIME OF THE INTERRUPTION COULD BE LOST. IN ORDER TO MINIMIZE THIS POSSIBILITY THE OPERATOR SHOULD RESTART DEVICES ACTIVE AT THE TIME OF THE INTERRUPTION. LOOK AT YOUR CONSOLE SHEETS TO DETERMINE WHAT OUTPUT DEVICES WERE ACTIVE. AFTER SIGNING ON AGAIN, RESTART THE JOB ON THAT DEVICE. IF THE PRINTER WAS ACTIVE ON A CERTAIN JOB AT THE TIME OF THE INTERRUPTION, AND THE SAME JOB BECOMES ACTIVE ON THE PRINTER AGAIN AFTER YOU HAVE SIGNED ON AND ENABLED THE PRINTER, ISSUE A "R PPI,J" MESSAGE TO ASP. ALL PRINT DATA SETS FOR THAT JOB WILL BE SENT IN THEIR ENTIRETY. THIS MAY RESULT IN SOME DUPLICATION OF DATA, BUT IT IS THE BEST WAY TO REDUCE THE POSSIBILITY OF LOST DATA.

THE OCCURRENCE OF AN ERROR CONDITION IN COMMUNICATIONS PROCESSING CAUSES THE WORKSTATION PROGRAM TO DISPLAY AN EIGHT-DIGIT LOG ERROR AND VERBAL DESCRIPTION MESSAGE ON THE CONSOLE. IN GENERAL, THE DISPLAYING OF ERROR MESSAGES IS INFORMATIONAL ONLY SINCE THE PROGRAM AUTOMATICALLY INITIATES THE APPROPRIATE RECOVERY. HOWEVER, FREQUENT LOGGING OF ERROR MESSAGES COULD INDICATE A SERIOUS PROBLEM. FOR EXAMPLE, REPEATED TIMEOUTS (05U10000 UNIT CHECK) MAY SHOW THAT THE TERMINAL HAS LOST COMMUNICATIONS WITH ASP. FOLLOWING IS A TABLE WITH ALL COMMUNICATIONS ERROR MESSAGES:

01PREE00	BLOCK SEQ CK	BLOCK SEQUENCE CHECK - A BUFFER WAS DUPLICATED OR LOST. KR = RECEIVED BLOCK NUMBER. EE = EXPECTED BLOCK NUMBER.
02000000	ERRPR REPLY	NEGATIVE REPLY RECEIVED - A BUFFER WAS NOT RECEIVED CORRECTLY BY ASP.

03RRRR00	INVALID RESP	UNKNOWN RESPONSE RECEIVED. RRRR = FIRST TWO CHARACTERS RECEIVED FROM ASP.
04000000	UNIT EXCPTN	UNIT EXCEPTION - TERMINAL HAS RECEIVED AN "EOT" CHARACTER.
05SS0000	UNIT CHECK	UNIT CHECK - A CHECK CONDITION IN THE COMMUNICATIONS ADAPTER. SS = SENSE BYTE. SS = 01 - TIMEOUT SS = 02 - LOST DATA SS = 04 - OVERRUN SS = 08 - DATA CHECK SS = 10 - EQUIPMENT CHECK SS = 40 - INTERVENTION REQUIRED SS = 80 - COMMAND REJECT
06CC0000	UNUSUAL END	UNUSUAL END - AN UNKNOWN CONDITION IN THE CHANNEL OR CONTROL UNIT INTERFACE. CC = CSW BYTE 5.
07CC0000	SIO FAILURE	SIO FAILURE - START I/O INSTRUCTION WAS REJECTED BY THE ADAPTER.

\*\*\*\*\*  
 \* 2. THE EXPANDED SPOOLING FEATURE \*  
 \*\*\*\*\*

THE TERMINAL PROGRAM USES A DISK DIRECTORY AND SPOOLS ALL PRINT AND PUNCH OUTPUT TO DISK. THE PROGRAM LOOKS FOR ALL INPUT CARD READER DATA ON A SPOOL DISK.

THE DIRECTORY CONTAINS AN ENTRY FOR EACH SPOOL DATA SET, BOTH INPUT AND OUTPUT. THE PROGRAM HANDLES OUTPUT ON A DATA SET BASIS. IF ONE JOB CREATES THREE PRINT DATA SETS, EACH IS SPOOLED AS A SEPARATE DATA SET AND EACH HAS ITS OWN DIRECTORY ENTRY.

A DIRECTORY ENTRY HAS THE FOLLOWING FIELDS:

- (1) WORKSTATION DATA SET NAME. THIS DSN IS IN THE FORM "WSXYYJ.JOBNAME" WHERE
  - (A) "WS" STANDS FOR "WORKSTATION",
  - (B) "XX" REPRESENTS A TWO-CHARACTER DEVICE IDENTIFIER,
    - RD FOR READER,
    - PU FOR PUNCH,
    - PR FOR PRINTER,
  - (C) "YYY" REPRESENTS A THREE-DIGIT NUMBER WHICH INSURES THAT EACH DATA SET HAS A DISTINCTIVE NAME,
  - (D) "JOBNAME" REPRESENTS THE NAME OF THE JOB THAT FAN
    - AT TCC TO CREATE AN OUTPUT DATA SET, OR THE NAME TAKEN FROM THE JOB CARD THAT IS THE FIRST RECORD OF AN INPUT DATA SET.
- (2) ASP JOB NUMBER. THIS IS THE NUMBER OF THE JOB THAT FAN AT TCC TO CREATE THIS OUTPUT DATA SET.
- (3) DNAME. THIS IS THE DD NAME ASSOCIATED WITH THIS SPECIFIC OUTPUT DATA SET.
- (4) FORMS. THIS IS THE EIGHT CHARACTER FORMS IDENTIFICATION REQUESTED FOR AN OUTPUT DATA SET. STANDARD FORMS ARE ASSUMED FOR BOTH THE PRINTER AND THE PUNCH. WHEN A FORMS CHANGE IS SCHEDULED, THE PROGRAM INSERTS THE NEW FORMS ID IN THIS FIELD. THE PROGRAM RESPONDS AUTOMATICALLY TO MOUNT FORMS MESSAGES ON PRINT AND PUNCH OUTPUT.
- (5) LINES. THIS IS THE NUMBER OF LINES OF OUTPUT FOR THIS

DATA SET, OR THE NUMBER OF CARDS IN AN INPUT DATA SET. AN INPUT DIRECTORY ENTRY HAS ONLY THE WORKSTATION DATA SET NAME AND LINES, WHILE AN OUTPUT ENTRY CONTAINS ALL FIVE FIELDS.

UPON RECEIVING AN OUTPUT DATA STREAM, THE TERMINAL PROGRAM ALLOCATES DIRECT ACCESS SPACE FOR A NEW DATA SET. AN ENTIRE DATA SET IS WRITTEN TO DISK, THEN IS CLOSED. THE PROGRAM ADDS TO THE WORKSTATION DIRECTORY A NEW ENTRY FOR THIS DATA SET. THEN THE PROGRAM REPEATS THIS PROCESS FOR THE NEXT DATA SET IN THE STREAM.

ON THE INPUT SIDE, THE READER UTILITY PLACES DATA SETS ON THE APPROPRIATE DISK PACK AND CREATES DIRECTORY ENTRIES. UNDER THE TERMINAL PROGRAM THE OPERATOR INITIATES THE READING AND TRANSMITTING OF INPUT BY ENTERING THE COMMAND "RDR" AT THE CONSOLE. THE PROGRAM SEARCHES THE DIRECTORY FOR INPUT DATA SET NAMES. UPON FINDING ONE, THE WORKSTATION OPENS AND READS THAT DATA SET. WHEN FINISHED, THE PROGRAM SCRATCHES THE DATA SET AND DELETES ITS DIRECTORY ENTRY. THEN THE PROGRAM CONTINUES TO SEARCH THE DIRECTORY FOR OTHER INPUT DATA SET ENTRIES. IT PROCESSES THEM UNTIL IT CAN FIND NO MORE.

THE PROGRAM ACCEPTS THESE LOCAL COMMANDS RELATED TO THE DIRECTORY:

1. LIST - LIST THE ENTIRE WORKSTATION DIRECTORY ON THE OPERATOR'S CONSOLE. THE PROGRAM WRITES THIS HEADER:

WS DATA SET NAME AJN DD NAME FORMS LINES

THEN THE PROGRAM DISPLAYS THE INFORMATION FOR EACH DIRECTORY ENTRY IN THE APPROPRIATE COLUMN. IF THE DIRECTORY CONTAINS NO ENTRIES, THE PROGRAM RESPONDS WITH THIS MESSAGE:

WORKSTATION DIRECTORY EMPTY

2. RDR - START WORKSTATION READER. INITIATE SEARCH OF THE DIRECTORY FOR INPUT DATA SETS.
3. MODIFY - MODIFY WORKSTATION DIRECTORY. IT MAY BECOME NECESSARY TO USE THE MODIFY COMMAND IF A WORKSTATION

DATA SET DOES NOT HAVE A DIRECTORY ENTRY OR IF AN ENTRY EXISTS FOR A DATA SET THAT HAS BEEN SCRATCHED. THE MODIFY COMMAND HAS THESE TWO FORMS:

MODIFY,ADD,WSXYYY.ZZZZZZZ (ADD AN ENTRY)  
MODIFY,DEL,WSXYYY.ZZZZZZZ (DELETE AN ENTRY)

THE PROGRAM DETECTS SEVERAL ERROR CONDITIONS AND WRITES ERROR MESSAGES TO THE OPERATOR. IF THE ALLOCATION OF SPACE FOR AN OUTPUT DATA SET FAILS, THE PROGRAM DISPLAYS THE MESSAGE

ALLOCATION ERROR ON DATA SET WSXYYY.ZZZZZZZ

THEN THE PROGRAM ABENDS WITH THE ALLOCATION RETURN CODE IN REGISTER 15.

IF IT ENCOUNTERS AN ERROR IN OPENING AN OUTPUT DATA SET, THE TERMINAL PROGRAM ABENDS WITH A 3332 CODE.

IF THE TERMINAL PROGRAM FINDS NO INPUT DATA SETS ON ITS INITIAL SEARCH, THE PROGRAM LOGS THIS MESSAGE

NO INPUT DATA SETS FOUND IN SEARCH OF DIRECTORY

IN CASE THE PROGRAM ENCOUNTERS AN ERROR IN OPENING OR READING AN INPUT DATA SET, THE PROGRAM WRITES

DATA SET WSXYYY.ZZZZZZZ NOT READ,I/O ERROR

THE PROGRAM CLOSES THIS DATA SET AND THEN OPENS AND READS THE NEXT ONE IT FINDS.

IN CASE THE PROGRAM IS UNABLE TO OPEN THE DIRECTORY DATA SET, THE WORKSTATION ABENDS WITH A 3329 COMPLETION CODE.

IF THE DIRECTORY IS FULL AND THE PROGRAM CANNOT ADD AN ENTRY, THE PROGRAM WRITES THE MESSAGE

DIRECTORY FULL,DATA SET WSXYYY.ZZZZZZZ HAS NO ENTRY

THE OPERATOR MUST TAKE NOTE OF THIS MESSAGE. SUBSEQUENTLY WHEN SPACE IS AVAILABLE HE CAN ADD A DIRECTORY ENTRY BY USING THE TERMINAL MODIFY COMMAND.

THE PROGRAM HAS FOUR WTO'S ASSOCIATED WITH THE MODIFY COMMAND. IF THE OPERATOR DOES NOT ENTER THE MODIFY COMMAND IN THE CORRECT FORMAT THE PROGRAM WRITES THE MESSAGE

ADD OR DELETE NOT SPECIFIED, REENTER COMMAND

IF THE DATA SET NAME IN THE MODIFY COMMAND IS NOT VALID, THE WORKSTATION RESPONDS

INCORRECT FORMAT FOR DSNNAME, REENTER COMMAND

ONCE IT HAS ACCEPTED THE MODIFY COMMAND, THE PROGRAM ATTEMPTS THE ADD OR DELETE. IF SUCCESSFUL THE PROGRAM DISPLAYS THE LINE

DATA SET WSXXYYY.ZZZZZZZZ HAS BEEN ADDED (OR DELETED)

IF THE REQUESTED OPERATION DOES NOT WORK THE PROGRAM WRITES

ENTRY NOT ADDED (OR DELETED)

```
*****
* 3. SPOOL UTILITY - READER *
*****
```

THIS PROGRAM IS NAMED WSREADER (WORKSTATION READER). THE WSREADER IS AN ASSEMBLER LANGUAGE PROGRAM. A PROCEDURE NAMED WSREADER RESIDES IN SYSL.PROCLIB. THE PROCEDURE INVOKES THE PROGRAM. THE OPERATOR CAN START THE PROCEDURE AT THE CONSOLE. THE PROGRAMMER ALSO IS ABLE TO EXECUTE THE PROGRAM THROUGH JCL.

THIS GIVES THE USER TWO DIFFERENT METHODS FOR SPOOLING JOBS TO BE READ BY THE WORKSTATION TERMINAL PROGRAM. THE OPERATOR CAN START THE PROCEDURE BY ENTERING "S WSREADER" AT THE CONSOLE. THIS ASSUMES THAT THE CARD READER SPECIFIED IN THE PROCEDURE HAS BEEN MADE UNAVAILABLE TO OS/VS AND CAN BE DEDICATED TO THE UTILITY. THIS METHOD HAS THE ADVANTAGE THAT THE JOBS TO BE SPOOLED DO NOT HAVE TO BE FRAMED BY JCL. THIS ALTERNATIVE HAS A DISADVANTAGE IN THAT THE CARD READER CANNOT BE USED FOR OTHER INPUT WHILE THE WSREADER IS ACTIVE.

THE OTHER METHOD OF USING WSREADER IS TO SUBMIT A BATCH JOB THAT HAS AS INPUT THE JOBS TO BE TRANSMITTED TO TCC. THE BATCH JCL IS

```
//WSUTILITY JOB
// EXEC WSREADER
//INPUT DD DATA,DLN='*?'
// (INSERT JOBS TO BE TRANSMITTED)
// *? (DELIMITED CONSISTS OF ANY TWO SPECIAL CHARACTERS)
```

THIS ALTERNATIVE HAS THE ADVANTAGE OF NOT REQUIRING ANY OPERATIONAL ADJUSTMENT. THE JOBS TO BE SUBMITTED CAN BE READ BY THE OS/VS READER ALONG WITH REGULAR BATCH JOBS. THIS METHOD HAS THE DISADVANTAGE OF REQUIRING JCL TO EXECUTE WSREADER.

WSREADER SPOOLS EVERYTHING READ IN ONE INVOCATION OF THE PROGRAM INTO ONE DATA SET. WHEN THE UTILITY DETECTS END OF FILE IT CLOSES THE ACTIVE DATA SET AND ADDS ITS ENTRY TO THE WORKSTATION DIRECTORY. THERE CAN BE MULTIPLE JOBS IN ONE INPUT DATA SET.

IF THE OPERATOR STARTS THE UTILITY, IT WILL WORK A MESSAGE

UPON REACHING END OF FILE. THIS MESSAGE SAYS

REPLY START OR END TO WORKSTATION READER UTILITY

TO BEGIN READING AGAIN, REPLY "START". TO TERMINATE THE  
UTILITY RESPOND "END".

THE READER UTILITY DETECTS SEVERAL OF THE SAME ERROR  
CONDITIONS AS THE TERMINAL PROGRAM. IN ADDITION, THE  
WSREADER ABENDS WITH A 3338 CODE IF THE INPUT DEVICE DOES  
NOT OPEN CORRECTLY.



\*\*\*\*\*  
 \* . SPOOL UTILITY WRITER \*  
 \*\*\*\*\*

THIS PROGRAM IS NAMED WSWRITER (WORKSTATION WRITER). THE WSWRITER UTILITY IS AN ASSEMBLER LANGUAGE PROGRAM. TWO PROCEDURES THAT EXECUTE THE PROGRAM RESIDE IN SYSL.PROCLIB. THESE ARE WSPRINT (WORKSTATION PRINTER) AND WSPUNCH (WORKSTATION PUNCH). THE OPERATOR INITIATES THESE UTILITIES BY STARTING THEM AT HIS CONSOLE.

THE OPERATOR COMMAND IS IN THE FORM "S WSPRINT" OR "S WSPUNCH". THE PROCEDURE SPECIFIES THE UNIT ADDRESS OF EITHER A PRINTER OR A PUNCH. THIS DEVICE MUST BE MADE UNAVAILABLE TO OS/VSE BEFORE THE PROCEDURE IS STARTED.

ALL CONSOLE MESSAGES WRITTEN BY WSWRITER ARE FLAGGED WITH POUND SIGNS (#) ON THE RIGHT END. THIS MAKES THE MESSAGES OBVIOUS TO THE OPERATOR.

ONCE IT HAS BEGUN, THE WSWRITER UTILITY PROMPTS THE OPERATOR FOR ADDITIONAL PARAMETERS DEFINING THE ORDER IN WHICH TO HANDLE THE APPROPRIATE OUTPUT DATA SETS. THE UTILITY ALWAYS HAS AN OUTSTANDING WTO# WHICH THE OPERATOR MUST USE. THE WTO# MESSAGE IS

REPLY TO WORKSTATION PRINT (OR PUNCH) UTILITY

THE FIRST PROMPT MESSAGE THE OPERATOR RECEIVES IS THIS WTO#:

ENTER PRIORITY FACTOR (FORMS=,JOB=,DSN=,FIFO), LIST, OR END

THE OPERATOR REPLIES TO THIS MESSAGE ACCORDING TO THE ORDER IN WHICH HE WANTS THE DATA SETS PRINTED OR PUNCHED.

IF HE WANTS ALL SPOOL DATA SETS FOR A PARTICULAR FORMS TO BE PRINTED, HE REPLIES "FORMS=FULLWTO#". WSWRITER SEARCHES THE DIRECTORY, AND FINDS AND PRINTS ALL DATA SETS ACQUIRING THESE FORMS. AFTER COMPLETING THESE DATA SETS, THE PROGRAM EXECUTES THE "PRIORITY" WTO AGAIN. THE OPERATOR CAN ENTER ANOTHER REPLY.

IF HE ENTERS "JOB=TX123DEF" ALL DATA SETS FOR THE JOB NAMED TX123ABC ARE PRINTED SEQUENTIALLY IN THE ORDER IN WHICH THEY HAVE BEEN ENTERED IN THE DIRECTORY. WHEN ALL DATA SETS FOR THAT PARTICULAR JOB ARE PRINTED, THE PROGRAM REISSUES THE "PRIORITY" WTD.

IF HE DESIRES TO OUTPUT ONE PARTICULAR WORKSTATION DATA SET THE OPERATOR REPLIES "DSN=WSXYY.Y.ZZZZZZZZ". WSWRITER COMPLETES THIS DATA SET AND THEN LOGS THE "PRIORITY" WTD AGAIN.

THE REPLY "FIFO" STANDS FOR "FIRST IN, FIRST OUT". THIS SIGNIFIES THAT ALL DATA SETS FOR THAT DEVICE (EITHER PRINT OR PUNCH) WILL BE HANDLED IN THE SEQUENCE IN WHICH THEY ARE LOCATED IN THE DIRECTORY, REGARDLESS OF FORMS OR JOB NAME.

FOR THE ABOVE REPLIES THERE ARE TWO OPTIONAL PARAMETERS. THE FIRST IS "HOLD". THIS SIGNIFIES THAT THE DATA SETS ARE NOT TO BE SCRATCHED AND DELETED AFTER WSWRITER HAS FINISHED OUTPUTTING THEM. NORMALLY WHEN IT HAS COMPLETED A DATA SET, WSWRITER SCRATCHES THE DATA SET FROM THE SPOOL DISK AND DELETES ITS DIRECTORY ENTRY. IN ORDER TO PRINT ALL THE DATA SETS FOR A PARTICULAR JOB AND RETAIN THEM, THE OPERATOR ENTERS "JOB=TX123ABC,HOLD".

THE SECOND PARAMETER IS "COPIES=NNN", WHERE NNN IS ANY NUMERIC VALUE FROM 2 THROUGH 999. IF YOU WANT 10 COPIES OF A PARTICULAR DATA SET, REPLY "DSN=WSXYY.Y.ZZZZZZZZ,COPIES=10". WSWRITER DOES THIS DATA SET 10 TIMES. IF YOU NEED 5 COPIES OF ALL DATA SETS FOR A JOB, ENTER "JOB=TX123ABC,COPIES=5". WSWRITER OUTPUTS THE JOB'S FIRST DATA SET 5 TIMES, THEN DOES THE JOB'S SECOND DATA SET 5 TIMES, AND CONTINUES UNTIL IT HAS COMPLETED ALL DATA SETS FOR THAT JOB.

YOU CAN ENTER BOTH "HOLD" AND "COPIES=" WITH THE SAME COMMAND. FOR EXAMPLE, REPLY "DSN=WSXYY.Y.ZZZZZZZZ,HOLD,COPIES=2". EITHER PARAMETER MAY BE SPECIFIED FIRST. IF YOU ENTER AN INVALID NUMBER FOR "COPIES=", WSWRITER IGNORES THE VALUE AND OUTPUTS ONE COPY.

THE OPERATOR CAN ENTER A "LIST" REPLY TO THE "PRIORITY" WTO.  
THIS "LIST" COMMAND WORKS EXACTLY AS DOES THE TERMINAL PROGRAM "LIST". IT WRITES THE CONTENTS OF THE ENTIRE DIRECTORY TO THE CONSOLE.

THE "END" REPLY TO THE "PRIORITY" WTO TERMINATES WSWRITER.

WHEN STARTED, WSWRITER ASSUMES THAT THE DEVICE HAS STANDARD FORMS. WHEN IT ENCOUNTERS A DATA SET REQUIRING DIFFERENT FORMS, THE UTILITY WRITES

MOUNT FORMS=FL17C0AT, ENTER START WHEN READY

THE OPERATOR CHANGES THE FORMS, THEN REPLIES "START". THE UTILITY IGNORES OTHER REPLIES. WHEN THE FORMS HAVE BEEN CHANGED, WSWRITER ASSUMES THE DEVICE CONTAINS THE NEW FORMS AND DOES NOT ISSUE ANOTHER "MOUNT" MESSAGE UNTIL IT ENCOUNTERS A DATA SET REQUIRING DIFFERENT FORMS.

ONCE IT HAS THE INFORMATION REQUIRED TO BEGIN WRITING RECORDS, THE UTILITY DISPLAYS THE MESSAGE

\*SXYY. ZZZZZZZZ ACTIVE, ISSUE -ESTART OR CANCEL IF NECESSARY

THE UTILITY THEN BEGINS TO OUTPUT WITHOUT WAITING FOR A REPLY. IF THE OPERATOR WANTS TO START A DATA SET FROM ITS BEGINNING HE CAN ENTER "RESTART" IN REPLY TO THE OUTSTANDING WTO. WSWRITER REISSUES THE "ACTIVE" WTO AND BEGINS THE DATA SET AGAIN.

IF THE OPERATOR DECIDES NOT TO FINISH THE ACTIVE DATA SET HE CAN REPLY "CANCEL". WSWRITER RETAINS THE DATA SET AND ITS DIRECTORY ENTRY, BUT QUITS WAITING AND REISSUES THE "PRIORITY" MESSAGE.

THE WSWRITER UTILITY DETECTS SEVERAL OF THE SAME ERROR CONDITIONS AS THE TERMINAL PROGRAM. ADDITIONALLY, WSWRITER WILL ABEND WITH A 3339 CODE IF THE OUTPUT DEVICE CANNOT BE OPENED.

\*\*\*\*\*  
 \* 5. TAPEIN, TAPEOUT SUPPORT \*  
 \*\*\*\*\*

THIS CODE ENABLES THE WORKSTATION PROGRAM TO USE THE REMOTE TERMINAL TAPE DEVICE CAPABILITY OF THE TCC ASP SYSTEM.

THE WORKSTATION PROCEDURE HAS A TAPE DRIVE ASSIGNED TO IT BY UNIT ADDRESS. THIS DRIVE MUST BE MADE UNAVAILABLE TO OS/VIS BEFORE THE WORKSTATION IS STARTED.

THE TERMINAL PACKAGE IS DESIGNED SO THAT TAPEIN AND TAPEOUT WILL NOT WORK SIMULTANEOUSLY. IF TAPEOUT BECOMES SCHEDULED WHEN TAPEIN IS ALREADY ACTIVE, YOU MUST LET THE TAPEIN STEP COMPLETE. AFTER THE TAPE IS READ AND TAPEIN TERMINATED, YOU CAN START TAPEOUT.

ON BOTH TAPEIN AND TAPEOUT YOU WILL RECEIVE A MOUNT MESSAGE FROM TCC SPECIFYING "VOL=XXXXXX" WHERE "XXXXXX" INDICATES THE EXTERNAL LABEL OF THE TAPE TO BE USED.

IF THE TAPE D-IVE IS NOT READY WHEN THE TERMINAL PROGRAM DOES I/O, YOU WILL RECEIVE AN INTERVENTION REQUIRED MESSAGE. SIMPLY READY THE DRIVE AND THE I/O WILL BE INITIATED.

WHEN A JOB BECOMES ACTIVE ON TAPEIN, YOU WILL RECEIVE THE MESSAGES:

JOB 1000, JOBNAM IS ON FMC27T11  
 MOUNT INPUT TAPE VOL = XXXXX, START WHEN READY

MOUNT THE APPROPRIATE TAPE ON THE D-IVE ALLOCATED TO THE PROGRAM. THEN ENTER THIS LOCAL COMMAND:

.STI (START TAPE IN)

THE TAPE WILL BE READ AND TRANSMITTED.

DURING THE READING OF A TAPE YOU MAY DECIDE TO TERMINATE THE TAPEIN STEP. YOU CAN DO THIS BY ENTERING AT THE CONSOLE

THIS LOCAL COMMAND:

.ETI

(END TAPE IN)

IF BECAUSE OF ERROR A TAPE WITH AN INCORRECT BLOCKSIZE IS READ, THE TERMINAL PROGRAM LINE MAY BE CANCELLED AND COMMUNICATIONS LOST. YOU WILL SEE TIMEOUTS AT YOUR CONSOLE. RESTART YOUR TERMINAL PROGRAM. THE JOB WILL STILL BE ACTIVE ON TAPEIN. MOUNT THE CORRECT TAPE AND START TAPE IN.

AT THE END OF THE TAPEIN STEP YOU WILL RECEIVE THE MESSAGE

JOB 1000, JOBNAME 4321 BLOCKS READ

WHEN A JOB BECOMES ACTIVE ON TAPEOUT, YOU WILL RECEIVE THE MESSAGES

JOB 1000, JOBNAME IS ON FM 27T01

MOUNT TAPE VOL = XXXXXX ON FM 27T01, START WHEN READY

MOUNT THE CORRECT TAPE ON THE ALLOCATED DRIVE. THEN ISSUE AT THE CONSOLE THE LOCAL COMMAND

.STO

(START TAPE OUT)

THE DATA WILL BE WRITTEN TO TAPE.

WHEN TAPEOUT TERMINATES YOU WILL RECEIVE THE MESSAGES

JOB 1000, JOBNAME ON FM 27T01 COMPLETED

JOB 1000, JOBNAME VOL=XXXXXX, 4321 BLOCKS WRITTEN

```

*****
* 6. IMPLEMENTATION OF THE EXPANDED SPOOLING VERSION *
*****

```

THREE OBJECT MODULES MUST BE LINKED INTO A LOAD LIST LIBRARY:

```

WSTATION
WSREADER
WSWRITER

```

THIS JOB CREATES THE WORKSTATION DIRECTORY:

```

//WSCREATE JOB
// EXEC PGM=IEBDCG
//SYSPRINT DD SYSOUT=A
//SYSUT2 DD DSN=WSDIRECT,UNIT=SYSDA,VOL=SER=VVVVVV,
// DISP=(NEW,KEEP),SPACE=(TRK,(1)),
// DCB=(RECFM=F,LRCL=10500,BLKSIZE=10500)
//SYSIN DD *
DSD OUTPUT=(SYSUT2)
FD NAME=A,LENGTH=42,STARTLOC=01,
PICTURE=42,'ZZ
CREATE FILL=' ',NAME=(A)
END
/*

```

VVVVVV IS VOLUME WHERE DIRECTORY IS TO BE ALLOCATED. THE DIRECTORY HAS SPACE FOR 250 ENTRIES.

THIS JOB WILL ADD THE FOUR REQUIRED PROCEDURES TO SYSL.PROCLIB:

```

//WSUPDATE JOB
// EXEC PGM=IEBUPDATE,PARM=NEW
//SYSPRINT DD SYSOUT=A
//SYSUT2 DD DSN=SYSL.PROCLIB,DISP=SHR
//SYSIN DD DATA
.. ADD NAME=WSPRINT,LIST=ALL
// EXEC PGM=WSWRITER,TIME=1440
//INPUT DD DISP=SHR,VOL=SER=XXXXXX,UNIT=SYSDA,
// DSN=WSOUTPUT,DCB=(LPTCD=C)
//OUTPUT DD DCB=(RECFM=FB,LRCL=133,BLKSIZE=133,LPTCD=C),
// UNIT=PPP
//DIRECTRY DD DISP=SHR,VOL=SER=VVVVVV,UNIT=SYSDA,DSN=WSDIRECT
//SYSDUMP DD DISP=SHR,VOL=SER=ZZZZZZ,UNIT=SYSDA,DSN=WSDUMP

```

```

// ADD NAME=WSPUNCH,LIST=ALL
// EXEC PGM=WSWRITER,TIME=1440
//INPUT DD DISP=SHR,VOL=SER=XXXXXX,UNIT=SYSODA,
// DSN=WSDUMP,DCB=(OPTCD=C)
//OUTPUT DD DCB=(RECFM=FB,LRECL=30,BLKSIZE=80,OPTCD=C),
// UNIT=UUU
//DIRECTRY DD DISP=SHR,VOL=SER=VVVVVV,UNIT=SYSODA,DSN=WSDIRECT
//SYSUDUMP DD DISP=SHR,VOL=SER=ZZZZZZ,UNIT=SYSODA,DSN=WSDUMP
// ADD NAME=WSREADER,LIST=ALL
// EXEC PGM=WSREADER,TIME=1440
//INPUT DD DCB=(RECFM=FB,LRECL=80,BLKSIZE=80,OPTCD=C),
// UNIT=RRK
//OUTPUT DD DISP=SHR,UNIT=SYSODA,VOL=SER=YYYYYY,
// DSN=WSINPUT,SPACE=(CYL,(A,B),RLSE)
//DIRECTRY DD DISP=SHR,UNIT=SYSODA,VOL=SER=VVVVVV,DSN=WSDIRECT
//SYSUDUMP DD DISP=SHR,VOL=SER=ZZZZZZ,UNIT=SYSODA,DSN=WSDUMP
// ADD NAME=KSTATION,LIST=ALL
// EXEC PGM=KSTATION,TIME=1440
//INPUT DD DISP=SHR,VOL=SER=YYYYYY,UNIT=SYSODA,
// DSN=WSINPUT,DCB=(BUFNO=1)
//OUTPUT DD DISP=SHR,VOL=SER=XXXXXX,UNIT=SYSODA,
// DSN=WSOUTPUT,DCB=(BUFNO=1),SPACE=(CYL,(A,B),RLSE)
//DIRECTRY DD DISP=SHR,VOL=SER=VVVVVV,UNIT=SYSODA,DSN=WSDIRECT
//TAPE DD UNIT=TTT
//ADAPTR DD UNIT=CCC
//SYSUDUMP DD DISP=SHR,VOL=SER=ZZZZZZ,UNIT=SYSODA,DSN=WSDUMP
// ENDUP
//

```

VVVVVV IS THE VOLUME WHERE THE DIRECTORY RESIDES.  
XXXXXX IS THE VOLUME WHERE WORKSTATION OUTPUT WILL BE CREATED.  
YYYYYY IS THE VOLUME WHERE WORKSTATION INPUT WILL BE FOUND.  
ZZZZZZ IS THE VOLUME WHERE THE DUMP DATA SET RESIDES.  
CCC IS THE UNIT ADDRESS OF THE COMMUNICATIONS ADAPTER.  
PPP IS THE UNIT ADDRESS OF THE PRINTER ASSIGNED TO WSP-INT.  
UUU IS THE UNIT ADDRESS OF THE CARD PUNCH ASSIGNED TO WSPUNCH.  
RRK IS THE UNIT ADDRESS OF THE CARD READER ASSIGNED TO WSREADER.  
TTT IS THE UNIT ADDRESS OF THE TAPE DRIVE ASSIGNED TO KSTATION.  
A AND B ARE PRIMARY AND SECONDARY QUANTITIES FOR SPACE ALLO-  
CATION FOR WORKSTATION DATA SETS. THEY SHOULD BE SUFFICIENTLY  
LARGE TO HOLD ALL THE DATA FOR A SINGLE JOB.

ONE CYLINDER OF DISK SPACE SHOULD BE ALLOCATED FOR THE WORK-STATION DUMP DATA SET. THE FOLLOWING JOB WILL PRINT THE CONTENTS OF THIS DATA SET:

```
//WSDUMP JOB
// EXEC PGM=IEBGENER
//SYSPRINT DD SYSOUT=A
//SYSIN DD DUMMY
//SYSUT1 DD UNIT=SYSOA,VOL=SER=ZZZZZ,DISP=SHR,DSN=WSDUMP,
// DCB=(RECFM=VBA,LRECL=125,BLKSIZE=764)
//SYSUT2 DD SYSOUT=A,DCB=(RECFM=VBA,LRECL=125,BLKSIZE=764)
/*
```

IN CASE OF UNTIMELY END OF A TERMINAL SESSION, THE WORK-STATION PROGRAM MAY CREATE INCOMPLETE OUTPUT DATA SETS THAT DO NOT HAVE DIRECTORY ENTRIES. PERIODICALLY THE SYSTEMS PROGRAMMER SHOULD DO A LISTVIOC OF THE DISK WHERE WORKSTATION OUTPUT DATA SETS RESIDE. IF THE VOLUME CONTAINS A WORKSTATION DATA SET NOT FOUND IN THE DIRECTORY, ADD A DIRECTORY ENTRY WITH THE MODIFY COMMAND. THIS WILL ALLOW YOU TO RETRIEVE THE DATA THROUGH WSWRITER. UNFORTUNATELY, THERE IS NO SURE WAY OF KNOWING THE FORMS REQUIRED FOR THESE DATA SETS.