SURVEILLANCE AND CONTROL OF TRANSMISSION SYSTEMS (SCOTS) LOCALLY DEFINED OPERATIONS

CONTENTS PAGE

1. GENERAL 2

2. PROGRAM SEQUENCER 2

A. General 2

B. Instruction Set 5

C. Generating a PSEQ Job 9

Defining the Problem 9

Flowcharting 10

Coding the PS and PF Files . . . 10

Testing and Debugging 15

Keyboard Entry 15

D. Manual Execution 17

E. Automatic Execution 17

F. Listing 20

PS or PF File 20

PSEQ Data Base 20

PSEQ File Directory 20

G. Editing 20

PS Files 22

PF Files 23

H. Advanced Techniques 23

I. Examples 24

Example 1 24

Example 2 25

Example 3 25

Example 4 26

Example 5 28

J. Error Messages 29

K. Delete a PS or PF File 29

3. LOCALLY FORMATTED DISPLAY SUMMARIES 29

A. General 29

B. Planning 31

4. AUTOMATIC REMOTE SWITCH PROVISION 31

A. General 31

B. Planning 32

5. ALARM PRINTOUT ROUTING 33

A. General 33

B. Planning 33

6. COMMAND FILTERING 35

A. General 35

Printed in U.S.A.

CONTENTS PAGE

B. Planning 36

C. Example 37

7. PROCESSING PROGRAM FILES 37

A. Operating Personnel-Identification (ID) File 37

B. Open-Door Reset Switch (ODRS) File 38

C. AC Restore Switch (ACRS) File . . 38

8. WELCOM FILES AND FILES FOR USE WITH THE MSG COMMAND 39

A. WELCOM 39

B. Files for Use With MSG Command . 39

9. REFERENCES 39

1. GENERAL

1.01 This section describes certain locally defined operations which can result in more efficient operation of the SCOTS (Surveillance and Control of Transmission Systems) central. Before attempting to master the material in this section, the operating personnel should read the following sections:

SECTION	TITLE
190-205-101	SCOTS System—General Description
190-205-102	SCOTS Central Terminal Description
190-205-302	SCOTS Data Base Management

1.02 This section has been reissued to update the information for the program sequencer and to remove information no longer applicable to this section.

1.03 This section contains information on how to create program sequencer jobs which can result in the performance of a series of SCOTS centrally related operations, how to create a visual display for displaying related information, how to provide for the automatic operation of a remote switch when an alarm or status point becomes set, how to provide for routing of alarm messages to certain CRT (cathode-ray tube) console terminals, how to designate CRT terminals, how to designate CRT terminals as full control or partial control and how to assign alarm responsibility to partial-control terminals, how to set up a standard system for logging operating personnel in and out of normally unattended remote stations, how to provide for automatic reset on the open-door alarm, and how to provide for automatic restoration of commercial AC power during power failure.

2. PROGRAM SEQUENCER

A. General

2.01 Program sequencer (PSEQ) is a software package which provides field personnel with the ability to create and execute meaningful jobs. These jobs describe an ordered sequence of SCOTS centrally related actions which perform a specific task.

2.02 An interesting possibility for PSEQ would be to provide restoration plans for communication routes. This job would describe the actions required to provide this restoration, and could be executed by a simple one-line keyboard command. The actions required to perform the restoration plan would be described in the PSEQ job as which remote switches are to be operated, which displays or scans for statuses and alarms are to be initiated, etc. The advantage of PSEQ is the ability to initiate the job much faster and more accurately than any SCOTS operator.

2.03 PSEQ jobs can also be executed automatically when an alarm or status point becomes set. For example, a PSEQ job could execute automatically on the occurrence of a particular alarm. This job could initiate a remote switch and then scan for certain status points. If the status points are set or clear in an appropriate manner, then the switch operated properly and no operator intervention would be required. The PSEQ job could also signal the operator only if the status points are not appropriately set or clear.

2.04 All PSEQ jobs produce outputs to one or more output LU numbers. At the very least, there will be a **SEQUENCE COMPLETE** message appearing. Other messages result from errors and from **PRINT** steps, **GO** steps, **SCAN** steps, and

VERIFICATION steps. The designations of the outputs can be directed in several ways as follows:

(1) The **PR** and **GO** steps each have a self-contained output LU specified by the creator of the sequence. These LUs are used unless there is an override LU as in one of two cases. The first case is when a sequence is manually executed with an LU specified in the command (**#SEQ, ABC, L29**). The second case is when the sequence is scheduled through an entry in the program sequence data base (PSDB) that specifies an output LU. In both of these cases the override LU is used for all functions.

(2) The **SCAN** and **VERIFY** results and also the general messages of errors or completion are directed according to the following priority scheme:

(a) For manual execution (**#SEQ**), see Part D:

- To the LU number specified with the **SEQ** command

- If no LU specified as above, to the LU number specified in the PF file

- If no LU specified as above, to the LU from which the sequence was requested.

(b) For automatic execution (via PSDB), see Part E:

- To the LU number specified in the PSDB

- If no LU specified as above, to the LU number specified in the PF file

- If no LU specified as above, to LU#1.

All of the above is summarized as shown below:

	SEQ,ABC	SEQ,ABC,LX	PSDB	PSDB,LX

PF	P=S
	F=CO
		P=LX
		E=LX
			P=S
			E=LU#1
				P=LX
				E=LX

PF,LZ	P=S
	E=LZ
		P=LX
		E=LX
			P=S
			E=LZ
				P=LX
				E=LX

Where:

P = **PRINT** and **GO** messages

E = Error messages, **END** messages, and **SCAN** results

S = as specified in the steps of the PF file

CO = on the requesting console

LX = on logical unit X

LZ = on logical unit Z

2.05 As previously mentioned, each PSEQ job describes actions required to accomplish a particular task. This description is in the form of a series of steps, and each step represents a PSEQ instruction. These instructions, described in detail later, provide the ability to initiate a remote switch, scan for alarms and statuses, test these alarms and statuses to determine if certain ones are set or clear, log messages into the History Log, operate the office minor or major alarm bell, terminal alarm processing of receiving alarms, and print messages on an output device such as a CRT console, teleprinter or line printer. There are other instructions which provide the capabilities that a PSEQ job can perform. These will be described later in this section.

2.06 PSEQ instructions are stored in files. A PSEQ file is a list of data stored on the disc which defines the steps required to perform a task. Two files are required for each PSEQ job. They are as follows:

PS—Program step file

PF—Parameter file

The PS file is an ordered list of data which describes each instruction in a PSEQ job on a step-by-step basis. The PF file contains the parameters associated with each instruction in the PS file. For example, a step in the PS file could be described as a remote switch instruction, while the PF file would contain the switch number and remote station number associated with the switch.

2.07 Each PS or PF file must be given a unique name. This name can be up to six characters in length, and the first character must not be a blank or a number. Embedded blanks are not allowed. Permissible characters are as follows:

A thru Z
0 thru 9
"
#
$
&
' (Apos)
(
)
*

.
/
;
<
=
>
?
@
\
]
↑

2.08 Meaningful names should be given to the PS and PF files. For example, a name for a file which provides a restoration plan can be given the name RSP001. This would convey more meaning than a name such as A9#/Z. In addition, special consideration should be given to the PF file name since the PSEQ job is executed by referencing that name.

2.09 PSEQ jobs are stored in the real-time executive (RTE) file manager (FMGR) area on the fixed disc (LU#43). Since there is a finite number of tracks available to FMGR, heavy users of sequencer may be required to carefully control the number and size of the sequences they create.

2.10 It is impossible to put an exact upper limit upon the number of sequences that may be introduced into the available disc space. The space required by a given sequence depends not only upon the number of steps it contains, but on the types of steps as well. There is room for about 1500 sequences if they are generally not much larger than the *styles* shown in Table A. (Table A shows several different styles of sequences that are approximately the same length.) A few sequences that are many times larger than those cited will also fit, because there will be many that will be much smaller than those cited. As for the maximum length of one sequence, it is far greater than one would ever need. When considering a very long sequence, consider breaking the job down into smaller sequences that are linked by *call* steps. In this way, changes may be introduced without incurring a vast amount of retyping. Both Table A and the above discussion will become more meaningful after several sequences have actually been created.

TABLE A

SEQUENCE STYLES

STEP TYPES	A	B	C	D	E	F	G
AL, CA, CO, EN, FL, GO, IN, JP, LP, RS, SC, TA, OR WT	0	15	35	55	80	100	125 (STEPS)
YR	0	5	4	3	2	1	0 (STEPS)
LG OR PR	11	5	4	3	2	1	0 (LINES)

2.11 The PS file describes the sequence and type of steps in the PSEQ job. When generating a PSEQ job, the PS file must be created first then the PF file can be created. The PSEQ software is arranged such that more than one PSEQ job can use the same PS file. Each PSEQ job ***must*** have a separate PF file.

B. Instruction Set

2.12 The PSEQ software provides a set of instructions which can be used to create a PSEQ job. The following is a description of each of these instructions.

PRINT — This instruction prints an operator-programmed message on a specified output device.

Entry in PS file — **PR**

Parameters in PF file — (1) ASCII message
(2) Output device

The message may be up to 250 characters in length. Any ASCII character except the exclamation point (!) may be used to construct the message. The output device symbolic code may be any of those specified in Table A.

ALARM — This instruction can turn on or turn off the office major or minor alarm bell.

Entry in PS file — **AL**

Parameters in PF file — (1) Alarm bell (major or minor)
(2) Turn on (set) or off (clear).

LOG – This instruction logs an operator-programmed message in the history log by identification number (IDNO), class number (CLA#), or station number (STA#)

Entry in PS file — **LG**

Parameters in PF file — (1) ASCII message
(2) Log by IDNO, CLA#, or STA#

The message may be up to 250 characters in length. Any ASCII character except the exclamation point (!) may be used to construct the message. The message may be logged into the history log by IDNO, CLA#, or STA#. If the PF file indicates station 0, logging will occur under the station referenced in the most recently executed RS or SC step. Identification numbers are sequentially tagged to alarms that are detected by the SCOTS central. An example of an IDNO is the 4-digit number outputted with each alarm and most command operations. Class numbers are assigned during generation of the SCOTS data base to certain categories of alarms. For example, radio channel alarms could be assigned a class number, and messages could be stored and retrieved via this number.

WAIT – This instruction provides a preprogrammed delay before execution of the next instruction.

Entry in PS file — **WT**

Parameters in PF file - A delay specified in either seconds, minutes, hours, or days

The minimum wait intervail is 1 second, and the maximum, 22 days. Longer waits can be created by multiple wait instructions.

GO – This instruction causes **CONTINUE (Y, N, OR STEP#)?** to be displayed at a selected I/O device. This in effect causes the PSEQ job execution to be stopped until the operator types either **Y**, **N**, or step number.

Where **Y** — continue with job at the next step
N — abort program
Step number — continue with job at the specified step

Entry is PS file — **GO**

Parameters in **PS** file — I/O device on which the **CONTINUE** message is to be displayed (see Table A)

The **GO** instruction is intended to be used after a **PRINT** instruction which would describe the operation that is about to occur.

Example: **PROTECTION CHANNEL UNAVAILABLE**
CONTINUE (Y, N, OR STEP #)?

JUMP — This instruction causes an unconditional jump to a specified instruction step number.

Entry in PS file — **JP**

Parameters in PF file — step number.

REMOTE SWITCH – This instruction causes a remote switch operation to be performed at either an E- or C-type remote station.

Entry in PS file — **RS**

Parameters in PF file — (1) Station number
(2) Type station (E— or C-type)
(3) Switch number
(4) Switch type (momentary, latch operate, or latch release)
(5) If desired, log by class number or ID number, or do not log
(6) Class number (if logging by class)

If it is desired to log this operation, the message that is logged is not operator-defined. The log message is created by the PSEQ software and contains information to identify the station number, switch number, and the PF file name.

SCAN – This instruction causes a scan operation to be initiated at a specified E- or C-type remote. The current state of alarms within the scan is printed.

Entry in PS file —**SC**

Parameters in PF file — (1) Station number
(2) Scan number or **CUR** operation
(3) If scan, **REGULAR** or **HISTORY** type of scan.

VERIFY – This instruction compares the status and alarm bits in the previous scan with the bits specified in the PF file. If the bits compare, the PSEQ job continues with the next instruction. If the bit do not compare (ie, verify errors), the PSEQ job can be terminated or the job can be made to jump to an instruction specified in the PF file.

Entry in PS file — **VR**

Parameters in PF file — (1) Bits, if any, that should be set (Logic 1)
(2) Bits, if any, that should be clear (Logic 0)
(3) If verify error occurs, terminate PSEQ job or which instruction to jump to.

The verify instruction uses the scan results of the most recent scan executed within the job. It is allowable to use more than one verify instruction without intervening scan instructions.

CALL — This instruction provides the ability for one PSEQ job to call for execution of another PSEQ job.

Entry in PS file — **CA**

Parameters in PF file — file name of task to be called.

This instruction is useful when it is desired to split a lengthy PSEQ job into many parts. After testing and debugging, the parts can be linked together via the **CALL** instruction. Another use is to link existing PSEQ jobs to a newly created job. This saves on disc storage since the new job does not have to provide storage for duplication of the existing job. Other details of the **CALL** instruction are given in 2.32 and 2.33.

TERMINAL ALARM — This instruction sends an alarm indication (CTRL-G) to any specified SCOTS operating terminal.

Entry in PS file — **TA**

Parameters in PF file — (1) Major or minor
(2) Set or clear
(3) LU number of terminal
(4) Station number — always use 0.

FLAG — This instruction administers a set of 1536 storage locations in which numerical values in the range of 0 through 9999 may be manipulated by the sequencer. The values are preserved even when no sequence jobs are running, so they may be accessed at a later time by the same or by a different sequence job.

Entry in PS file — **FL**

Entry in PF file — (1) Set the flag to a specified value.
(2) Increment the flag by specified value.
(3) Compare the flag value to a specified value and branch to one of three specified steps depending upon whether the comparison yields <, =, or >.

See paragraphs 2.33 through 2.44 for further details.

LOOP — This instruction provides the ability to repeat a series of steps a specified number of times.

Entry in PS File — **LP**

Parameters in PF File — (1) The step which is the beginning of the loop.
(2) Number which represents one less than the number of times the "loop" is to be repeated.

A "loop" can be programmed to perform a series of steps from 1 to 9,999 times. After the programmed number of loops have been completed, the instruction immediately after the loop instruction will be executed.

MCO – This instruction terminates the processing of alarms by the SCOTS central for a particular station

Entry in PS file —**CO**

Parameters in PF file — (1) Station number
(2) Terminate (set) or reinstate (clear) alarm processing.

If the PF file indicates station 0, the **MOC** will be applied to the station referenced in the most recently executed RS or SC step. Alarms associated with stations that have their alarm processing terminated are ignored by the SCOTS central and do not result in the ringing of the office alarm bell.

INITIALIZE — This instruction will return the state of the parameter file to the original operator defined state.

Entry in PS file — **IN**

Parameters in PF file — None.

See paragraphs 2.34 through 2.36 for details on the use of this instruction.

END – This instruction terminates execution of the PSEQ job. It is the last logical instruction to be executed in the PSEQ sequence. More than one **END** instruction may occur if the sequence has several branches.

Entry in PS file — **EN**

Parameters in PF file — None

/E — This is not one of the PSEQ instructions, but should be the last physical step in any PSEQ job. Only one/**E** is allowed in each PSEQ job. An **EN** instruction *must immediately precede* the/**E** instruction.

Entry in PS file — **/E**

Parameters in PF file — None

2.13 This paragraph presents a simple example of a few PSEQ instructions just described. The following is a PSEQ job which performs a remote switch (at E2 station 17, switch 3, log under CLA. NO 11), waits 4 hours, then scans display 12 at station 17 for alarms. Next, a message is outputted at the console assigned logical unit number 23 (L23) stating **OPERATOR SHOULD EXAMINE PREVIOUS SCANS—CLA. NO. 11.**

PS FILE	PF FILE
RS	Station 17, switch 3, CLASS NO. 11
WT	4 hours
SC	Station 17, display 12
PR	At L23, "OPERATOR SHOULD EXAMINE PREVIOUS SCANS — CLA. NO. 11"
END	
/E	

C. Generating a PSEQ Job

Defining the Problem

2.14 The first step in generating a PSEQ job is to define the problem. This definition might take the form of a paragraph or a listing of steps. The definition should provide details as to which remote stations, scans, etc, are involved. See Fig. 1 for the example.

```
TD-3 Restoration Plan Between Freehole and Podock
  1. Print message describing this PSEQ job.
  2. Allow for operator to abort this job — GO instruction.
  3. Scan display 12 at station 17.
  4. If bits 2, 16, and 60 are not set or if bits 3, 4 or 5 are not clear, move to Step 5. If bits 2, 16,
     and 60 are set and bits 3, 4, and 5 are clear, then:
     A. Operate switch 12 at E2 stations 17 and 18. Log this operation under class number 33.
     B. Scan display 12 at station 18.
     C. If bits 2, 3, and 6 are set, then:
        (1) Print message that restoration is complete.
        (2) Log similar message under class number.
        (3) End.
        Else: (a) Print message that restoration cannot be made.
              (b) Log similar message under class number 33.
              (c) End.
  5. — Else: A. Print message that restoration cannot be made.
             B. Log similar message under class number 33.
             C. End.
```

Fig. 1—Example of a PSEQ Job Description

Flowcharting

2.15 From the definition of the problem, a flowchart can be made. Each block in the flowchart should be equivalent to one of the instructions previously described. Figure 2 is an example of a flowchart developed from the definition given in Fig. 1. Note the **verify** decision block. The verify instruction operates as follows. If no errors occur, the job continues with the next instruction. If a verify error occurs, one can cause the PSEQ job to either terminate or jump to a specified instruction. A termination at this point will be announced on the appropriate output device. The step number and sequence name will be given. Suggested flowchart symbols are given in Fig. 3.

Coding the PS and PF Files

2.16 From the PSEQ job flowchart, a PS and PF file can be generated which can be used to enter the job into the SCOTS computer. To generate a PS or PF file, first translate the flowchart into code via the instructions previously described. Each block in the flowchart, if properly developed, represents one of the instructions. To generate the PS file, simply list these instructions as a series of steps. The parameters for each step are used to generate the PF file. Figure 4A is an example of the PS and PF files generated from the flowchart of Fig. 2. Figure 4B is a form which can be copied and used to generate the PS and PF files.

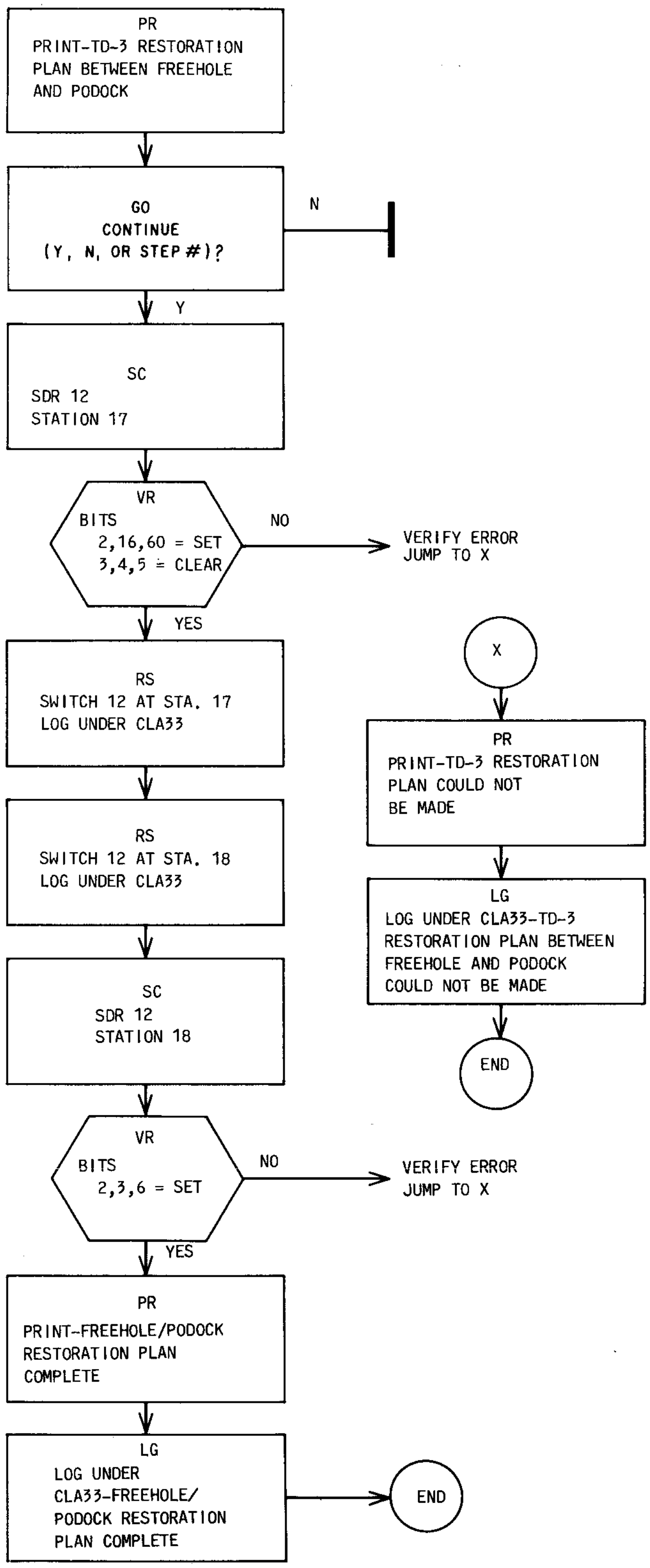

PR
PRINT-TD-3 RESTORATION PLAN BETWEEN FREEHOLE AND PODOCK
GO
CONTINUE (Y, N, OR STEP #)?
N
Y
SC
SDR 12
STATION 17
VR
BITS
2,16,60 = SET
3,4,5 = CLEAR
NO
VERIFY ERROR JUMP TO X
YES
RS
SWITCH 12 AT STA. 17
LOG UNDER CLA33
RS
SWITCH 12 AT STA. 18
LOG UNDER CLA33
SC
SDR 12
STATION 18
VR
BITS
2,3,6 = SET
NO
VERIFY ERROR JUMP TO X
YES
PR
PRINT-FREEHOLE/PODOCK RESTORATION PLAN COMPLETE
LG
LOG UNDER CLA33-FREEHOLE/ PODOCK RESTORATION PLAN COMPLETE
END
X
PR
PRINT-TD-3 RESTORATION PLAN COULD NOT BE MADE
LG
LOG UNDER CLA33-TD-3 RESTORATION PLAN BETWEEN FREEHOLE AND PODOCK COULD NOT BE MADE
END

Fig. 2—Example of a PSEQ Job Flowchart

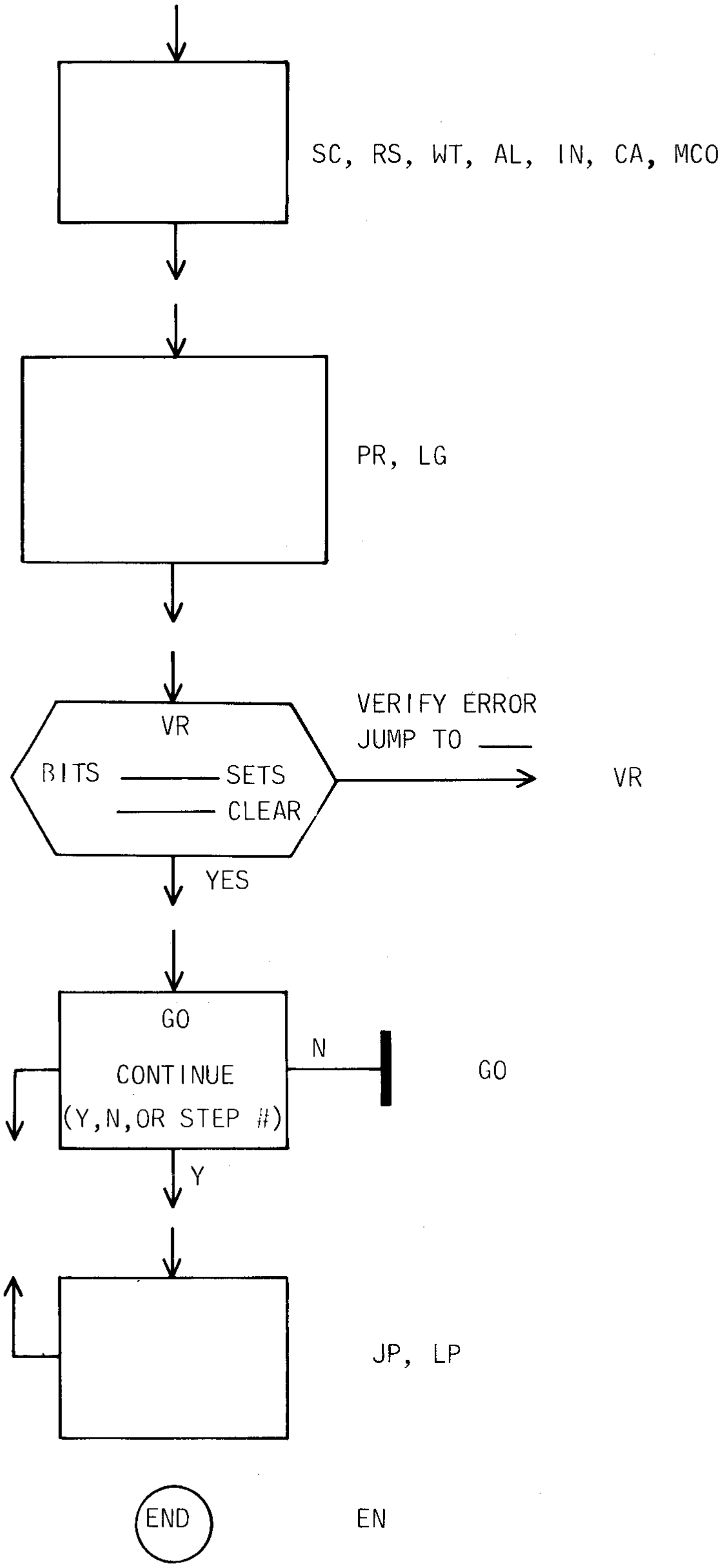


Fig. 3—Flowchart Symbols

PSEQ JOB

FILE DEFINITION PS: RSP TD-3 FREEHOLE - PODOCK
(MAX. 55 CHARS.) RSP TD-3 FREEHOLE - PODOCK

NAME: PSRPFP NAME: RP.FP

STEP	PS FILE	PF FILE
1	PR	TD-3 Restoration Plan between Freehole and Podock
2	GO	CO
3	SC	SDR 12, Sta. 17
4	VR	Bits 2,16,60 = set 3, 4, 5 = Clear / Verify error, Jump to Step 12
5	RS	STATION 18 - NOT A CI REMOTE SWITCH # 12 - MOMENTARY - LOG UNDER CLASS 33
6	RS	STATION 17 - NOT A CI REMOTE SWITCH # 12 - MOMENTARY - LOG UNDER CLASS 33
7	SC	SDR 12, Sta. 18
8	VR	Bits 2,3,6 = SET, none clear / Verify error, Jump to Step 12
9	PR	Freehole / Podock Restoration Plan Complete
10	LG	Under CLA #33 Freehole/Podock Restoration Plan Complete
11	EN	———
12	PR	TD-3 Restoration Plan Could Not Be Made
13	LG	Under CLA #33 Restoration Plan Between Freehole And Podock. Could Not Be Made
14	EN	———
15	/E	———

Fig. 4A—PSEQ Job—Example Worksheet

PSEQ JOB

FILE DEFINITION PS:
(MAX. 55 CHARS.) PF:

	NAME ______	NAME ______
STEP	PS FILE	PF FILE

Fig. 4B—PSEQ Job—Blank Worksheet

Testing and Debugging

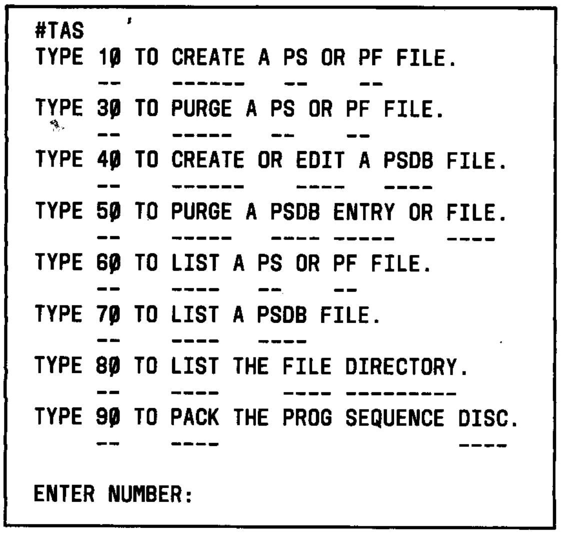
2.17 After completing the code for the PS and PF files, it may be wise to let another person who is knowledgeable with program sequencer examine the work. The reason for this is that testing and debugging a new program sequencer job is accomplished on a "live" SCOTS System. Defects in the PSEQ job could be catastrophic as far as the monitored equipment is concerned. Again, extreme care should be used in the testing and debugging phase of PSEQ job generation.

2.18 PSEQ jobs that are intended to be run automatically should be debugged before providing for automatic execution. Execute new PSEQ jobs manually via the **SEQ** command until they are completely debugged. The **SEQ** command will be described later in paragraph 2.23.

Keyboard Entry

2.19 The PSEQ job is entered into the SCOTS computer via an appropriate console. Part 6 discusses which consoles are available for this entry.

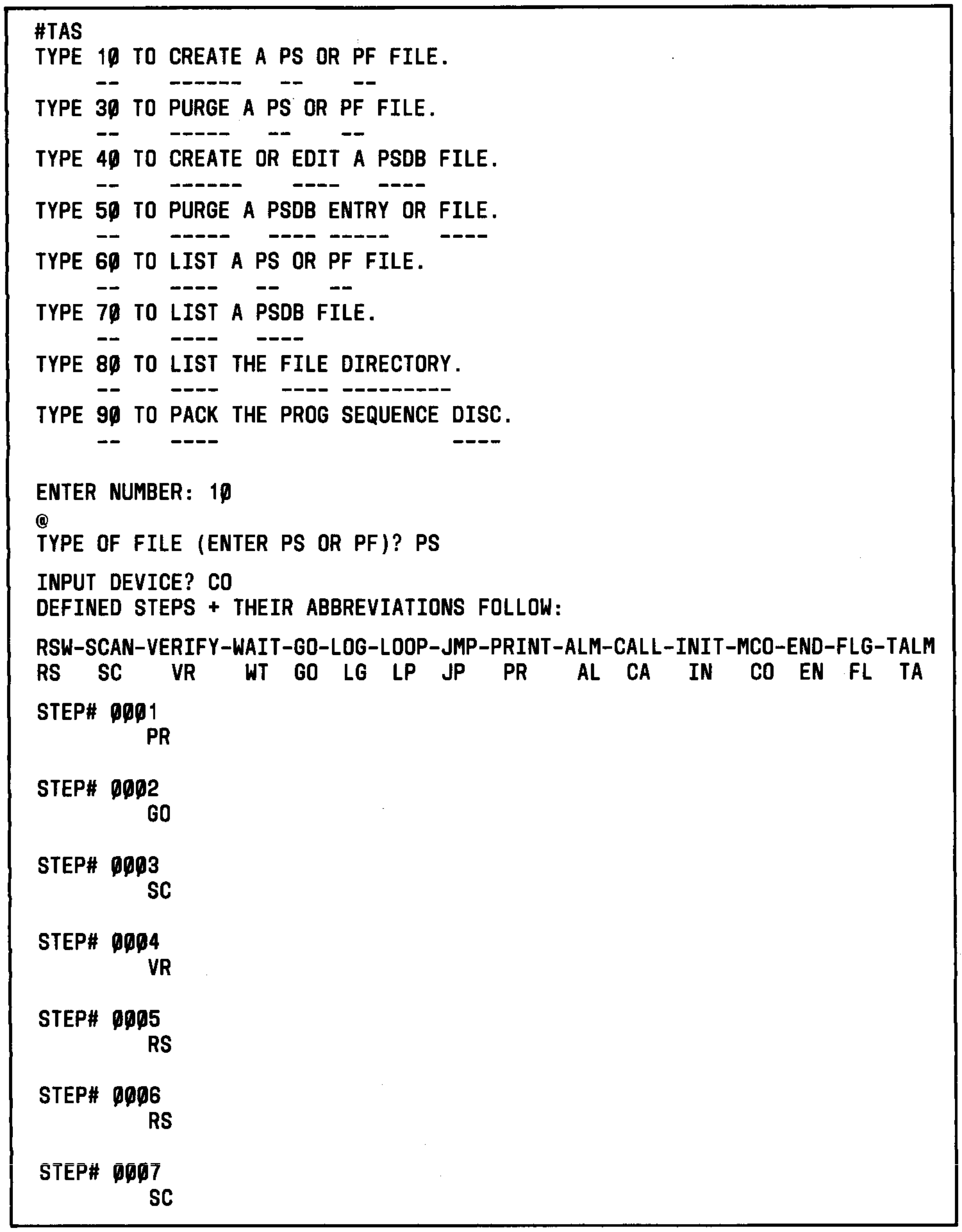
2.20 The PSEQ job is entered into the computer by initiating the **TAS** command. The **TAS** command is initiated by simply typing **TAS** followed by a carriage return. The SCOTS computer will then return several lines of messages similar to Fig. 5. The word **TAS** followed by an associated number can be typed, and the printout of Fig. 5 will not occur. For example, if a PS file is wanted, simply type **TAS 1Ø**.



```
#TAS
TYPE 1Ø TO CREATE A PS OR PF FILE.
     --    ------   --    --
TYPE 3Ø TO PURGE A PS OR PF FILE.
     --    -----   --    --
TYPE 4Ø TO CREATE OR EDIT A PSDB FILE.
     --    ------    ----   ----
TYPE 5Ø TO PURGE A PSDB ENTRY OR FILE.
     --    -----   ---- -----    ----
TYPE 6Ø TO LIST A PS OR PF FILE.
     --    ----   --    --
TYPE 7Ø TO LIST A PSDB FILE.
     --    ----   ----
TYPE 8Ø TO LIST THE FILE DIRECTORY.
     --    ----     ---- ---------
TYPE 9Ø TO PACK THE PROG SEQUENCE DISC.
     --    ----                   ----

ENTER NUMBER:
```

Fig. 5—Printout From Initiating the TAS Command

2.21 To enter a PS or PF file, enter **1Ø** when the computer responds with **ENTER NUMBER:**. The computer will begin asking for information; respond accordingly. Figure 6 is an example of the printout from entry of the code in Fig. 4A for the PS file. Figure 7 represents the entry for the PF file and also for Figure 4A. After the file has been entered, the computer will respond with **INPUT FILE DEFINITION (54 CHARS MAX)**. Here the operator may enter a 54-character description of the file. This is useful in that later the operator can request a listing of the file directory and a cross-reference of each of file names versus the file description. The keyboard entry for the file directory is described in paragraph 2.31.



```
#TAS
TYPE 1Ø TO CREATE A PS OR PF FILE.
     --    ------   --    --
TYPE 3Ø TO PURGE A PS OR PF FILE.
     --    -----    --    --
TYPE 4Ø TO CREATE OR EDIT A PSDB FILE.
     --    ------    ----   ----
TYPE 5Ø TO PURGE A PSDB ENTRY OR FILE.
     --    -----    ---- -----    ----
TYPE 6Ø TO LIST A PS OR PF FILE.
     --    ----   --    --
TYPE 7Ø TO LIST A PSDB FILE.
     --    ----   ----
TYPE 8Ø TO LIST THE FILE DIRECTORY.
     --    ----     ---- ---------
TYPE 9Ø TO PACK THE PROG SEQUENCE DISC.
     --    ----                   ----

ENTER NUMBER: 1Ø
@
TYPE OF FILE (ENTER PS OR PF)? PS
INPUT DEVICE? CO
DEFINED STEPS + THEIR ABBREVIATIONS FOLLOW:
RSW-SCAN-VERIFY-WAIT-GO-LOG-LOOP-JMP-PRINT-ALM-CALL-INIT-MCO-END-FLG-TALM
RS   SC    VR    WT  GO  LG  LP  JP   PR    AL  CA   IN   CO  EN  FL  TA
STEP# ØØØ1
         PR

STEP# ØØØ2
         GO

STEP# ØØØ3
         SC

STEP# ØØØ4
         VR

STEP# ØØØ5
         RS

STEP# ØØØ6
         RS

STEP# ØØØ7
         SC
```

Fig. 6—Printout Which Occurs When Entering a PS File (Page 1 of 2)

```
STEP# ØØØ8
          VR
STEP# ØØØ9
          PR
STEP# ØØ1Ø
          LG
STEP# ØØ11
          EN
STEP# ØØ12
          PR
STEP# ØØ13
          LG
STEP# ØØ14
          EN
STEP# ØØ15
          /E
NAME OF THE FILE (MAX 6 CHAR'S) PSRPFP
INPUT FILE DEFINITION (55 CHARS MAX)
RESTORATION TD-3 FREEHOLE PODOCK
REQUEST COMPLETE
```

Fig. 6—Printout Which Occurs When Entering a PS File (Page 2 of 2)

2.22 Although a paper tape punch is not provided as part of the SCOTS central, some SCOTS operating personnel may have access to some off-line equipment that is equipped with a punch or a reperforator. In that case the operator may choose to prepare all **TAS,1Ø** answers off-line on tape. To do this, perform the following operations:

(1) Begin the answer tape with the answers for Step 0001.

(2) Include all answers from that point up to but not including the file name and file definition.

(3) Follow every answer with a carriage return and a line feed in that order.

(4) Answer preliminary **TAS,1Ø** questions from the keyboard—when message **INPUT DEVICE?** is displayed, enter **TA** (for tape).

D. Manual Execution

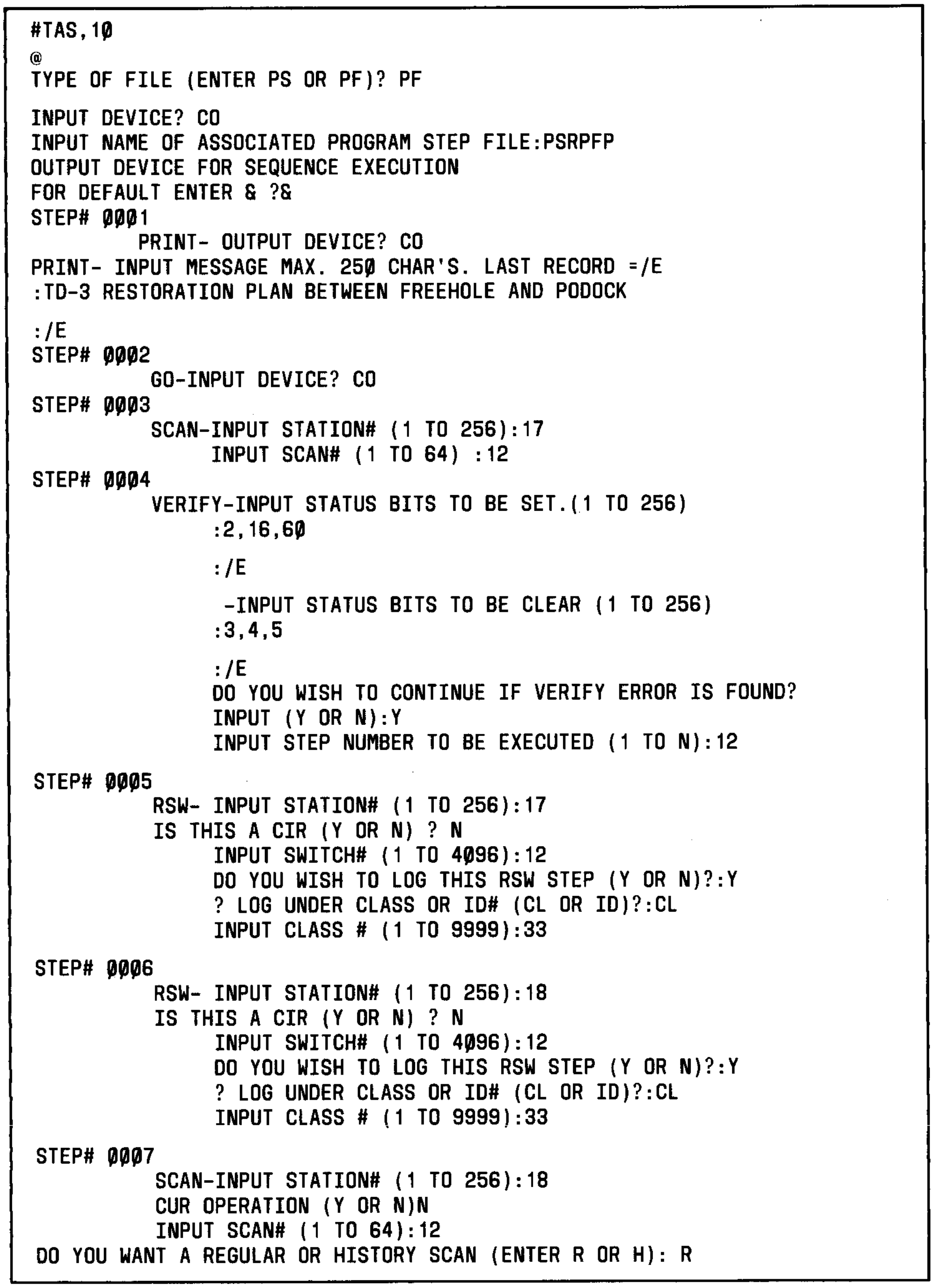
2.23 Any PSEQ job can be executed manually. This is accomplished by initiating the **SEQ** command in either of two methods: interactive and noninteractive entries. The interactive entry is accomplished by simply typing **SEQ** followed by a carriage return. The computer will respond with **INPUT PF FILE NAME MAX 6 CHARS:**. Type the PF file name associated with the PSEQ job followed by a carriage return. The computer will then respond with **READOUT DEVICE (& IF DEFAULT)?**. Generally, your response will be **&**. Other inputs are described in paragraph 2.04.

2.24 The **SEQ** command can also be executed noninteractively as follows. First, type **SEQ** followed by a comma; then type the PF file name, and follow this with a carriage return. If a readout device is desired, enter this as an additional parameter. Examples are: **SEQ,RSP1Ø1** or **SEQ,RSP1Ø,L29**, or **SEQ,RSP1Ø1,CO**. Note that the noninteractive way of manually executing a PSEQ job requires only one line of keyboard entry versus three lines for the interactive entry.

E. Automatic Execution

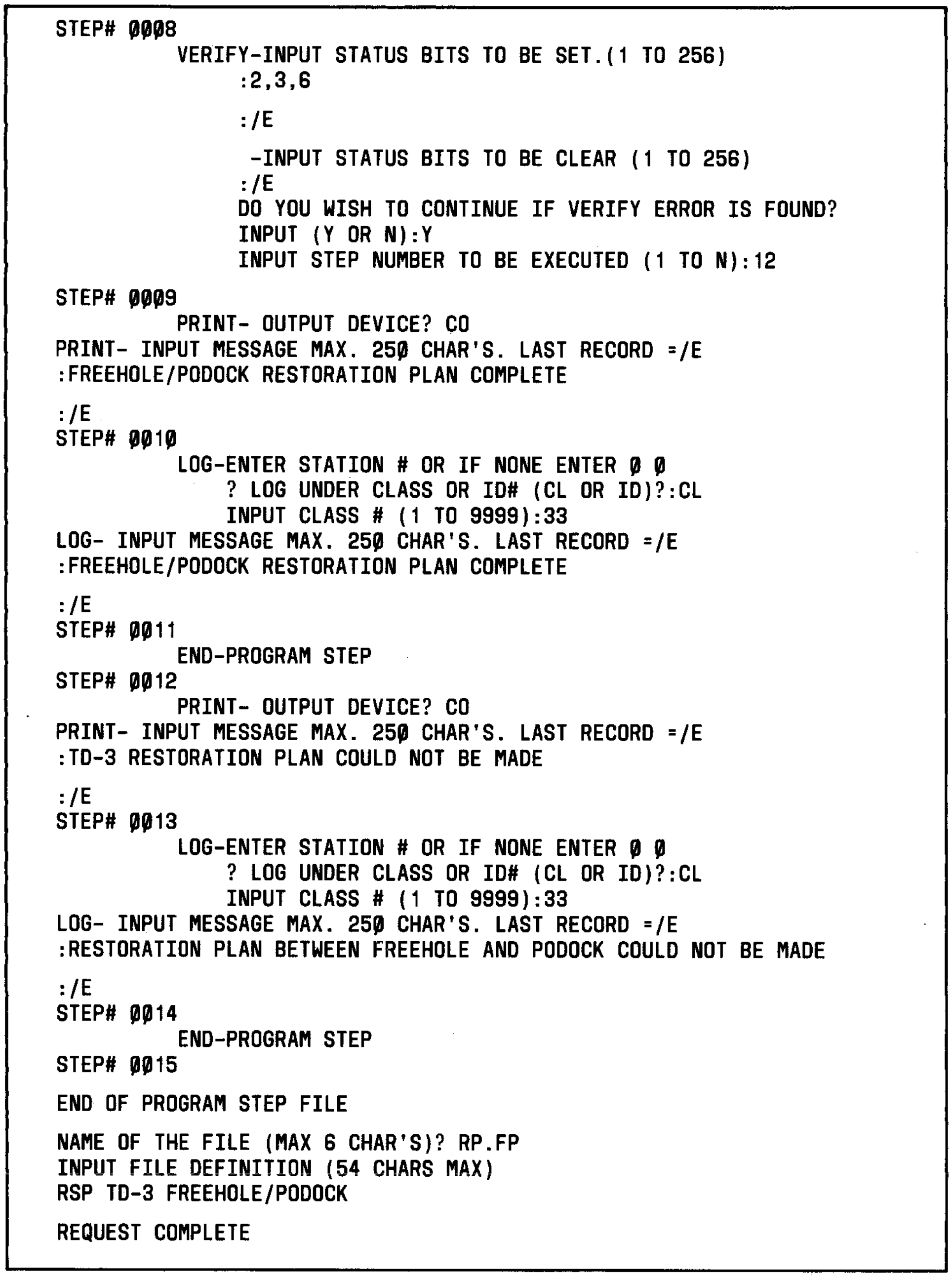
2.25 A PSEQ job can be made to execute automatically when an alarm or status point becomes set. This is accomplished by linking the SCOTS System data base to the PSEQ data base. The PSEQ data base will provide a cross-reference of station number, SDR number, and status point number versus the name of the PSEQ job (PF file name) to be executed. (Recall that the name of the PSEQ job is taken as the PF file name.) Thus, when an alarm or status point becomes set, the SCOTS System data base "links" to the PSEQ data base which in turn selects the appropriate PSEQ job to execute.

2.26 Figure 8 is an example of a typical printout of entering information to provide for automatically executing a PSEQ job. To provide for automatic execution of a PSEQ job, first make an entry in the PSEQ data base. This is accomplished by initiating the **TAS** command. To initiate the **TAS** command, simply type **TAS** followed by a carriage return. A printout similar to Fig. 5 will occur. Where the message **ENTER NUMBER** occurs, type **4Ø** followed by a carriage return. Enter the information requested by the computer as required. For further information reguarding the output device for sequence execution, refer to paragraph 2.04.



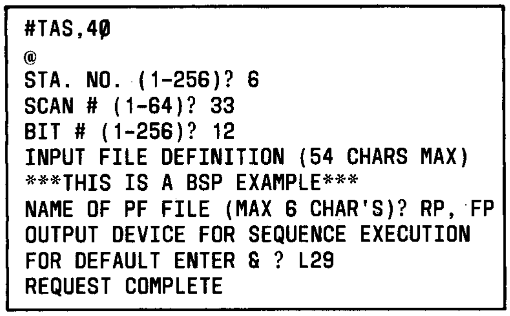
```
#TAS,1Ø
@
TYPE OF FILE (ENTER PS OR PF)? PF

INPUT DEVICE? CO
INPUT NAME OF ASSOCIATED PROGRAM STEP FILE:PSRPFP
OUTPUT DEVICE FOR SEQUENCE EXECUTION
FOR DEFAULT ENTER & ?&
STEP# ØØØ1
         PRINT- OUTPUT DEVICE? CO
PRINT- INPUT MESSAGE MAX. 25Ø CHAR'S. LAST RECORD =/E
:TD-3 RESTORATION PLAN BETWEEN FREEHOLE AND PODOCK

:/E
STEP# ØØØ2
          GO-INPUT DEVICE? CO
STEP# ØØØ3
          SCAN-INPUT STATION# (1 TO 256):17
               INPUT SCAN# (1 TO 64) :12
STEP# ØØØ4
          VERIFY-INPUT STATUS BITS TO BE SET.(1 TO 256)
               :2,16,6Ø

               :/E

                -INPUT STATUS BITS TO BE CLEAR (1 TO 256)
               :3,4,5

               :/E
               DO YOU WISH TO CONTINUE IF VERIFY ERROR IS FOUND?
               INPUT (Y OR N):Y
               INPUT STEP NUMBER TO BE EXECUTED (1 TO N):12

STEP# ØØØ5
          RSW- INPUT STATION# (1 TO 256):17
          IS THIS A CIR (Y OR N) ? N
               INPUT SWITCH# (1 TO 4Ø96):12
               DO YOU WISH TO LOG THIS RSW STEP (Y OR N)?:Y
               ? LOG UNDER CLASS OR ID# (CL OR ID)?:CL
               INPUT CLASS # (1 TO 9999):33

STEP# ØØØ6
          RSW- INPUT STATION# (1 TO 256):18
          IS THIS A CIR (Y OR N) ? N
               INPUT SWITCH# (1 TO 4Ø96):12
               DO YOU WISH TO LOG THIS RSW STEP (Y OR N)?:Y
               ? LOG UNDER CLASS OR ID# (CL OR ID)?:CL
               INPUT CLASS # (1 TO 9999):33

STEP# ØØØ7
          SCAN-INPUT STATION# (1 TO 256):18
          CUR OPERATION (Y OR N)N
          INPUT SCAN# (1 TO 64):12
DO YOU WANT A REGULAR OR HISTORY SCAN (ENTER R OR H): R
```

Fig. 7—Printout Which Occurs When Entering a PF File (Page 1 of 2)



```
STEP# ØØØ8
          VERIFY-INPUT STATUS BITS TO BE SET.(1 TO 256)
               :2,3,6

               :/E

                -INPUT STATUS BITS TO BE CLEAR (1 TO 256)
               :/E
               DO YOU WISH TO CONTINUE IF VERIFY ERROR IS FOUND?
               INPUT (Y OR N):Y
               INPUT STEP NUMBER TO BE EXECUTED (1 TO N):12

STEP# ØØØ9
          PRINT- OUTPUT DEVICE? CO
PRINT- INPUT MESSAGE MAX. 25Ø CHAR'S. LAST RECORD =/E
:FREEHOLE/PODOCK RESTORATION PLAN COMPLETE

:/E
STEP# ØØ1Ø
          LOG-ENTER STATION # OR IF NONE ENTER Ø Ø
              ? LOG UNDER CLASS OR ID# (CL OR ID)?:CL
              INPUT CLASS # (1 TO 9999):33
LOG- INPUT MESSAGE MAX. 25Ø CHAR'S. LAST RECORD =/E
:FREEHOLE/PODOCK RESTORATION PLAN COMPLETE

:/E
STEP# ØØ11
          END-PROGRAM STEP
STEP# ØØ12
          PRINT- OUTPUT DEVICE? CO
PRINT- INPUT MESSAGE MAX. 25Ø CHAR'S. LAST RECORD =/E
:TD-3 RESTORATION PLAN COULD NOT BE MADE

:/E
STEP# ØØ13
          LOG-ENTER STATION # OR IF NONE ENTER Ø Ø
              ? LOG UNDER CLASS OR ID# (CL OR ID)?:CL
              INPUT CLASS # (1 TO 9999):33
LOG- INPUT MESSAGE MAX. 25Ø CHAR'S. LAST RECORD =/E
:RESTORATION PLAN BETWEEN FREEHOLE AND PODOCK COULD NOT BE MADE

:/E
STEP# ØØ14
          END-PROGRAM STEP
STEP# ØØ15

END OF PROGRAM STEP FILE

NAME OF THE FILE (MAX 6 CHAR'S)? RP.FP
INPUT FILE DEFINITION (54 CHARS MAX)
RSP TD-3 FREEHOLE/PODOCK

REQUEST COMPLETE
```

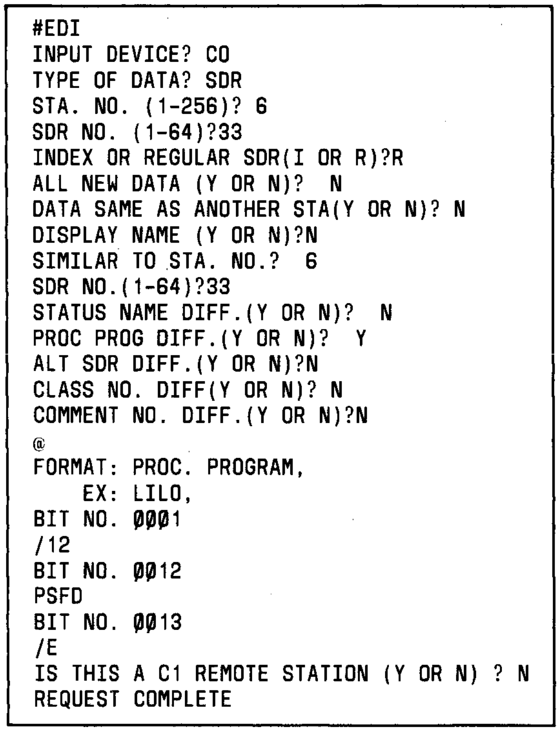
Fig. 7—Printout Which Occurs When Entering a PF File (Page 2 of 2)



```
#TAS,4Ø
@
STA. NO. (1-256)? 6
SCAN # (1-64)? 33
BIT # (1-256)? 12
INPUT FILE DEFINITION (54 CHARS MAX)
***THIS IS A BSP EXAMPLE***
NAME OF PF FILE (MAX 6 CHAR'S)? RP, FP
OUTPUT DEVICE FOR SEQUENCE EXECUTION
FOR DEFAULT ENTER & ? L29
REQUEST COMPLETE
```

Fig. 8—Printout Which Occurs When Making an Entry Into the PSEQ Data Base

2.27 When the PSEQ data base entry has been made, link the SCOTS data base to the PSEQ data base. This is accomplished by initiating the **EDI** command. To initiate the **EDI** command, type **EDI** followed by a carriage return. When the computer asks for **TYPE OF DATA?**, type **SDR** followed by a carriage return. Respond to the questions asked by the computer. When the alternate program name is to be entered, type **PSFD** (Fig. 9).



```
#EDI
INPUT DEVICE? CO
TYPE OF DATA? SDR
STA. NO. (1-256)? 6
SDR NO. (1-64)?33
INDEX OR REGULAR SDR(I OR R)?R
ALL NEW DATA (Y OR N)?  N
DATA SAME AS ANOTHER STA(Y OR N)? N
DISPLAY NAME (Y OR N)?N
SIMILAR TO STA. NO.?  6
SDR NO.(1-64)?33
STATUS NAME DIFF.(Y OR N)?  N
PROC PROG DIFF.(Y OR N)?  Y
ALT SDR DIFF.(Y OR N)?N
CLASS NO. DIFF(Y OR N)? N
COMMENT NO. DIFF.(Y OR N)?N
@
FORMAT: PROC. PROGRAM,
    EX: LILO,
BIT NO. ØØØ1
/12
BIT NO. ØØ12
PSFD
BIT NO. ØØ13
/E
IS THIS A C1 REMOTE STATION (Y OR N) ? N
REQUEST COMPLETE
```

Fig. 9—Printout Which Occurs When Making an Entry to Link the SCOTS Data Base to the PSEQ Data Base

F. Listing

2.28 Listing in accomplished by initiating the **TAS** command. The **TAS** command is initiated by simply typing **TAS** followed by a carriage return. The computer will return with several lines of messages similar to Fig. 5.

PS or PF File

2.29 To list a PS or PF file, type **6Ø** where the computer responds with **ENTER NUMBER:**. Follow this entry by a carriage return. Other responses will be required of the operator, such as the file name and the output device. Permissible entries for the output device are given in Table B. An example of a typical listing is given in Fig. 10.

PSEQ Data Base

2.30 To list the PSEQ data base, type **7Ø** where the computer responds with **ENTER NUMBER:** in the **TAS** command. Follow this entry by a carriage return. Figure 11 is an example of the listing of a typical PSEQ data base.

PSEQ File Directory

2.31 The PSEQ file directory lists each PS and PF file that has been entered into the SCOTS computer. To list this directory, type **8Ø** where the computer responds with **ENTER NUMBER:** in the **TAS** command. Follow this entry by a carriage return. Figure 12 is an example of a PSEQ file directory listing.

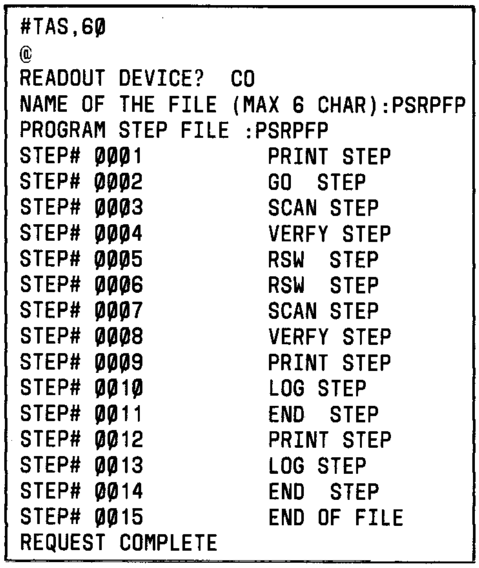
G. Editing

2.32 Editing any part of the sequencer package (ie, PSEQ data base editing) is accomplished by deleting the old entry and then entering the new entry. To delete a PSEQ data base entry, type **TAS,5Ø**. Figure 13 is an example of this operation. To delete a step file or a parameter file, type **TAS,3Ø**. Figure 14 is an example of this operation.

TABLE B

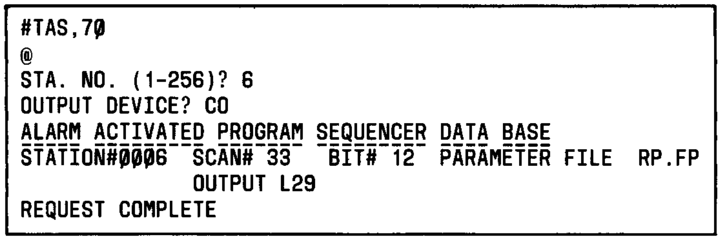
PERMISSIBLE ENTRIES FOR OUTPUT DEVICE

INPUT/OUTPUT DEVICE	LU# OR SYMBOL
Printer at SCOTS center	L6 or PR
CRT at SCOTS center	L1
Partial control terminals	L25, L26, L27, L28, L30, L31, L32, L33
Full control terminals	L23, L24, L29
Default to console where entry is made	CO



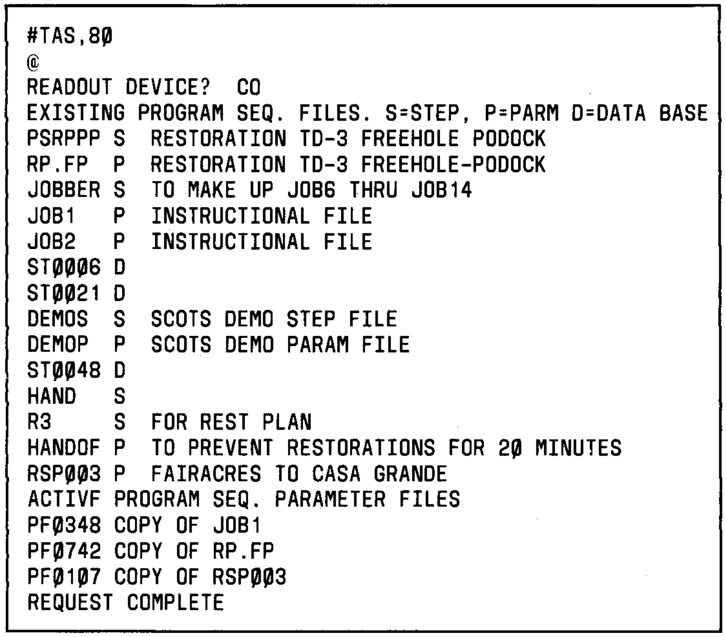
```
#TAS,6Ø
@
READOUT DEVICE?  CO
NAME OF THE FILE (MAX 6 CHAR):PSRPFP
PROGRAM STEP FILE :PSRPFP
STEP# ØØØ1          PRINT STEP
STEP# ØØØ2          GO  STEP
STEP# ØØØ3          SCAN STEP
STEP# ØØØ4          VERFY STEP
STEP# ØØØ5          RSW  STEP
STEP# ØØØ6          RSW  STEP
STEP# ØØØ7          SCAN STEP
STEP# ØØØ8          VERFY STEP
STEP# ØØØ9          PRINT STEP
STEP# ØØ1Ø          LOG STEP
STEP# ØØ11          END  STEP
STEP# ØØ12          PRINT STEP
STEP# ØØ13          LOG STEP
STEP# ØØ14          END  STEP
STEP# ØØ15          END OF FILE
REQUEST COMPLETE
```

Fig. 10—Printout Which Occurs When Requesting a Listing of a PS File



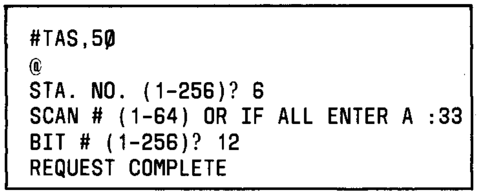
```
#TAS,7Ø
@
STA. NO. (1-256)? 6
OUTPUT DEVICE? CO
ALARM ACTIVATED PROGRAM SEQUENCER DATA BASE
STATION#ØØØ6  SCAN# 33   BIT# 12  PARAMETER FILE  RP.FP
              OUTPUT L29
REQUEST COMPLETE
```

Fig. 11—Printout Which Occurs When Requesting a Listing of the PSEQ Data Base Entries



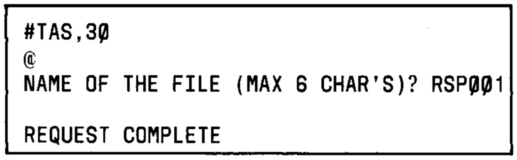
```
#TAS,8Ø
@
READOUT DEVICE?  CO
EXISTING PROGRAM SEQ. FILES. S=STEP, P=PARM D=DATA BASE
PSRPPP S  RESTORATION TD-3 FREEHOLE PODOCK
RP.FP  P  RESTORATION TD-3 FREEHOLE-PODOCK
JOBBER S  TO MAKE UP JOB6 THRU JOB14
JOB1   P  INSTRUCTIONAL FILE
JOB2   P  INSTRUCTIONAL FILE
STØØØ6 D
STØØ21 D
DEMOS  S  SCOTS DEMO STEP FILE
DEMOP  P  SCOTS DEMO PARAM FILE
STØØ48 D
HAND   S
R3     S  FOR REST PLAN
HANDOF P  TO PREVENT RESTORATIONS FOR 2Ø MINUTES
RSPØØ3 P  FAIRACRES TO CASA GRANDE
ACTIVF PROGRAM SEQ. PARAMETER FILES
PFØ348 COPY OF JOB1
PFØ742 COPY OF RP.FP
PFØ1Ø7 COPY OF RSPØØ3
REQUEST COMPLETE
```

Fig. 12—Printout Which Occurs When Requesting a Listing of the File Directory



```
#TAS,5Ø
@
STA. NO. (1-256)? 6
SCAN # (1-64) OR IF ALL ENTER A :33
BIT # (1-256)? 12
REQUEST COMPLETE
```

Fig. 13—Printout Which Occurs When Deleting a PSEQ Data Base Entry



```
#TAS,3Ø
@
NAME OF THE FILE (MAX 6 CHAR'S)? RSPØØ1

REQUEST COMPLETE
```

Fig. 14—Printout Which Occurs When Requesting to Delete a File

PS Files

2.33 Editing of PS files may always be accomplished by deleting the old file using **TAS,3Ø** and creating a new file using **TAS,1Ø.** Changes may also be made using the edit command **TAS,2Ø.** To use this feature, have available a paper reference copy of the PS file to be changed. At the terminal, type **TAS,2Ø** and answer the questions appropriately. When all the preliminary questions have been answered, the computer will type **STEP# ØØØ1** and wait for your input. At that point the 2-letter function may be changed by simply typing in the new abbreviation, or another step may be jumped to by using the slash feature (**/28** will move the prompt immediately to Step 28, etc). Upon arriving at a new step the 2-letter function may be changed as described above. If the edited file is to remain the same length as the original, terminate the edit by moving to the final step (END OF FILE) and entering **/E.** A given PS file may be shortened by changing one of the intermediate steps to an END OF FILE (using **/E**), or the file may be lengthened by changing the former END OF FILE to a different function and continuing with additional functions, ultimately finishing with an **EN** and a **/E.**

2.34 After the PS file steps have been modified and the **/E** has been entered, the computer will ask, **DO YOU WANT TO KEEP THE OLD FILE (Y OR N)?** When making the decision, remember the execution of sequence XYZ requires not only PF XYZ but also the PS file from which PF XYZ was made. This implies that the act of editing a PS file, destroying the original, and reassigning the same name should be followed by deletion and possibly recreation of the PFs (if any) that were created using the original PS file.

PF Files

2.35 Editing of PF files may always be accomplished by deleting the old file using **TAS,3Ø** and, creating a new file using **TAS,1Ø**. Changes may also be made using an edit command, **TAS,2Ø**. To use this feature, have available a paper reference copy of the PF file to be changed. Type **TAS,2Ø** and answer the questions appropriately. When all the preliminary questions have been answered, the computer will type **STEP# ØØØ1** followed by a question that is appropriate for the type of function called out in the specified PS file. At that point answers may be changed as desired (see limitations in next paragraph), or another step may be jumped to using the slash feature. Upon arriving at the new step, the computer will ask more questions appropriate to that particular function. Using the slash, it is possible to jump forward or backward, but the edit must ultimately be terminated by arriving at the final step in the file. See example in Fig. 15.

2.36 Limitations do exist in the editing of PRINT and LOG steps. It is recommended that neither of these steps be edited at all; but if they must be, the new copy must maintain the identical number of characters as the old copy. Character counts must include a value of two extra for each line (carriage return and line feed). In determining character count, be careful to include all spaces, all punctuation, and all nonprinting characters.

H. Advanced Techniques

2.37 The next few paragraphs will be concerned with the use of the **CALL, FLAG, INITIALIZE,** and **LOOP** instructions. In addition, the use of the incremental parameter will be discussed. There are several pecularities pertaining to these that deserve to be discussed in detail.

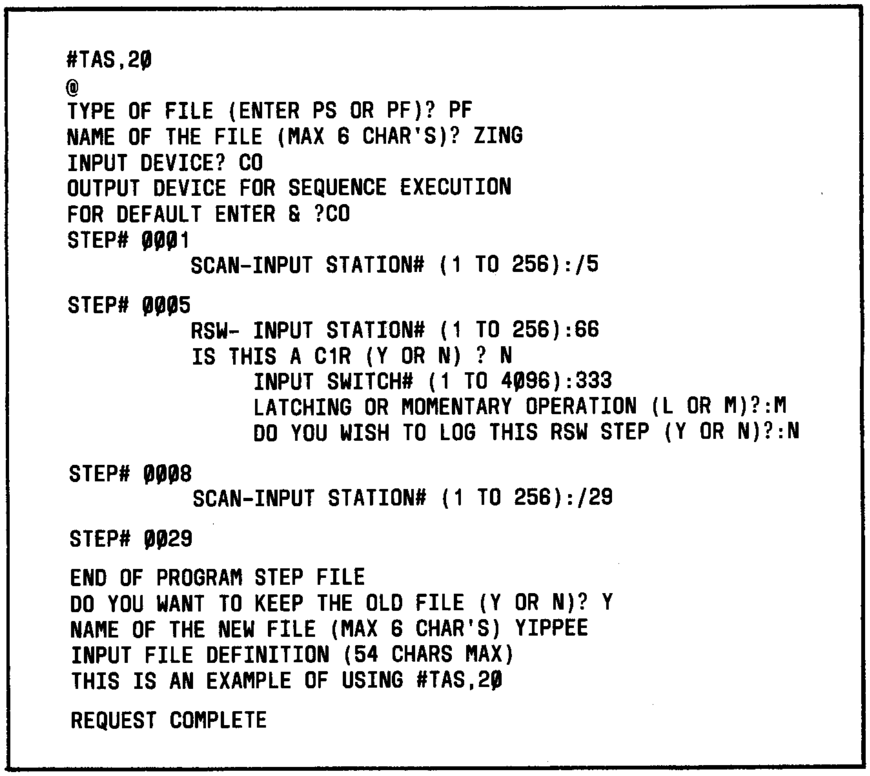
2.38 The **CALL** instruction is not exactly like the CALL instruction used in other systems' software. In program sequencer, the **CALL** instruction causes the called job to be scheduled immediately. Both the calling sequence and the called sequence will then execute more or less concurrently. See Example 1, Step 2, in Part I of this section.

2.39 The **FLAG** step manipulates a specified flag value as detailed in paragraph 2.12. This feature can be used in a myriad of ways. A detailed description of two methods of use is given in Examples 4 and 5 (paragraph 2.50). Other generalized examples are as follows:

(a) **FLAG** can be used as a counter—incrementing by 1 on every execution. After 50 executions, a branch to a **PR** step can indicate the status of the job. Another **FL** step can return the counter to 0.

(b) A **CALL** step is used in JOB1—calling for another job, JOB2, that must complete before JOB1 continues. JOB1 may set a specified flag, call JOB2, wait 5 seconds, test the flag, jump back to the **WAIT** step and remain in that loop until JOB2 changes the value of the flag to the new value expected by JOB1. JOB1 then tests again, finds the new value, and branches to another part of JOB1.

2.40 Consider the **INITIALIZE** and **LOOP** instructions. The **LOOP** instruction provides the ability to perform a series of steps a number of times. The number of times a step is to be performed is entered into the parameter file as one less than the number of times the sequence is to be performed. If the series of steps to be repeated contains an **RS** or an **SC** step, the station numbers, switch numbers, or the scan numbers may be made to increment by 1 immediately after that particular step is executed. In this way, looping back to that step several times can allow a series of related events to be requested. To introduce the incrementing feature, the station, switch, or scan number must be entered with the letter **I** appended (38I). See Example 2 in paragraphs 2.47 and 2.48.



```
#TAS,2Ø
@
TYPE OF FILE (ENTER PS OR PF)? PF
NAME OF THE FILE (MAX 6 CHAR'S)? ZING
INPUT DEVICE? CO
OUTPUT DEVICE FOR SEQUENCE EXECUTION
FOR DEFAULT ENTER & ?CO
STEP# ØØØ1
          SCAN-INPUT STATION# (1 TO 256):/5

STEP# ØØØ5
          RSW- INPUT STATION# (1 TO 256):66
          IS THIS A C1R (Y OR N) ? N
               INPUT SWITCH# (1 TO 4Ø96):333
               LATCHING OR MOMENTARY OPERATION (L OR M)?:M
               DO YOU WISH TO LOG THIS RSW STEP (Y OR N)?:N

STEP# ØØØ8
          SCAN-INPUT STATION# (1 TO 256):/29

STEP# ØØ29

END OF PROGRAM STEP FILE
DO YOU WANT TO KEEP THE OLD FILE (Y OR N)? Y
NAME OF THE NEW FILE (MAX 6 CHAR'S) YIPPEE
INPUT FILE DEFINITION (54 CHARS MAX)
THIS IS AN EXAMPLE OF USING #TAS,2Ø

REQUEST COMPLETE
```

Fig. 15—Example of TAS, 20 Edit Command

2.41 The **INITIALIZE** instruction is used with the incremental parameter. The incremental parameter can be used with the station number, scan number, and switch number as required in the **REMOTE SWITCH** and **SCAN** instructions. Its purpose is to increment the station number, scan number, or switch number each time the instruction is executed. The incremental parameter is initially set to the original operator-defined value.

2.42 The purpose of the **INITIALIZE** instruction is to reset the incremental parameter to the original operator-defined values. This in effect sets the station numbers, scan numbers, and switch numbers to their original values. Example 3 in paragraphs 2.49 and 2.50 illustrates the use of this instruction.

I. Examples

2.43 The following are a number of examples which illustrate the operation of certain program sequence instructions.

Example 1

2.44 This example illustrates the use of the **CALL** instruction. The following are two PSEQ jobs, JOE001 and JOE002. When JOE001 runs to completion, JOE002 executes.

JOE001

STEP	PS FILE	PF FILE
1	PR	"HI, I'M JOE001"
2	CA	JOE002
3	WT	59 Seconds
4	PR	"JOE001 OVER AND OUT"
5	END	
6	/E	

JOE002

STEP	PS FILE	PF FILE
1	PR	"JOE002 HERE"
2	END	
3	/E	

2.45 The printout from the PSEQ job is as follows:

```
"HI, I'M JOE001"
"JOE002 HERE"
"JOE001 OVER AND OUT"
```

2.46 Other messages may appear from program sequencer; however, only those pertaining to these PSEQ jobs are shown. Note the placement of the **CALL** instruction. It is executed before the **WAIT** statement in JOE001, so JOE002 executes long before JOE001 runs to completion.

Example 2

2.47 This example illustrates the use of the incremental parameter and **LOOP** instruction. The following is a PSEQ job which scans stations 13 through 16 for status display number 6.

SCAN01

STEP	PS FILE	PF FILE
1	SC	SDR6, station 13 with increment (13I)
2	LP	Loop to Step 1, 3 times
3	END	
4	/E	

2.48 Note that there are four stations to be scanned: 13, 14, 15, and 16. Yet, the parameter for the **LOOP** instruction is 3. The reason for this is that initially the station number is set to 13. Each succeeding time the **SCAN** instruction is executed, the station number is incremented. This requires three loops back to the **SCAN** instruction to get to station number 16. When the **LOOP** instruction has been satisfied, the next instruction, or **END**, gets executed. If this is not clear, read paragraphs 2.40 and 2.41.

Example 3

2.49 This example illustrates use of the **INITIALIZE** instruction. The **INITIALIZE** instruction resets the incremental parameter back to its original operator-defined value. The following is a PSEQ job which scans stations 13 through 16 for status display number 6, and then scans them again exactly the same.

SCAN02

STEP	PS FILE	PF FILE
1	SC	SDR6, station 13 with increment (13I)
2	LP	Loop to Step 1, 3 times
3	IN	
4	LP	Loop to Step 1, 1 time
5	END	
6	/E	

2.50 Note that as soon as Step 2 has been completed, the incremental parameter is set at 3. This in effect makes the station number in Step 1 equal to 16. The **INITIALIZE** instruction (Step 3) resets the incremental parameter to its original value, which in turn makes the station number equal to 13. If this is not clear, read paragraphs 2.40 and 2.41.

Example 4

This example illustrates **FLAG** communications between jobs. Job **HANDOF** simply sets a flag to a number greater than 0 to tell all of the restoration plan sequences to "forget it." If RSP003 runs during the hands-off interval, it will find flag 77 to be greater than 0 (>0), print out an appropriate message, and terminate. If the hands-off flag were 0, the restoration plan would have proceeded normally.

HANDOF

STEP	PS FILE	PF FILE
1	FL	Set flag 77 to value 999.
2	PR	"RESTORATION HANDS-OFF IN EFFECT."
3	WT	Wait 20 minutes.
4	FL	Set flag 77 to value 0.
5	PR	"HANDS-OFF CONDITION CLEARED."
6	EN	
7	/E	

RSP003

STEP	PS FILE	PF FILE
1	FL	Test flag 77: If 0, continue; if greater than 0, jump to Step 12.
2	RS	Switch 123, station 5.
3	SC	SDR 5, station 5
4	VR	(As required)
5	RS	(As required)
6	SC	(As required)
7	VR	(As required)
8	SC	(As required)
9	RS	(As required)
10	PR	"Restoration Plan 3 complete"
11	EN	
12	PR	"RSP003 impossible because of HANDS OFF CONDITION."
13	EN	
14	/E	

Example 5

This example illustrates a sequence that can proceed only if it is the first of its type to be initiated. For example, if several calls to sequence **SQUELCH** were made simultaneously, several copies of **SQUELCH** would be generated. However, all but the first to run would be quickly terminated.

SQUELH

STEP	PS FILE	PF FILE
1	FL	Test flag 92: If less than 500, go to Step 3. If equal to 500, go to Step 2. If greater than 500, go to Step 2.
2	EN	—
3	FL	Set flag 92 to value 2000.
4	RS	Switch 5,station 2051(with increment).
5	LP	To Step 4 30 times
6	WT	Twenty minutes.
7	FL	Set flag 92 to 0.
8	EN	—
9	/E	—

J. Error Messages

2.51 The following is a list of file manager error codes and their meanings which may occur when using program sequencer.

FMGR ERROR CODES

ERROR	MEANING
–17	ILLEGAL READ/WRITE ON TYPE 0 FILE
–16	ILLEGAL TYPE OR SIZE=0
–15	ILLEGAL NAME
–14	DIRECTORY FULL
–13	DISC LOCKED
–12	EOF OR SOF ERROR
–11	DCB NOT OPEN
–10	NOT ENOUGH PARAMETERS
–09	ATTEMPT TO USE A POSN OR FORCE TO 1 A TYPE 0 FILE
–08	FILE OPEN OR LOCK REJECTED
–07	BAD FILE SECURITY CODE
–06	CR OR FILE NOT FOUND OR NO ROOM
–05	RECORD LENGTH ILLEGAL
–02	DUPLICATE FILE NAME
–01	DISC ERROR

K. Delete a PS or PF File

2.52 A PS or PF File can be deleted from the SCOTS software when it is no longer needed. This will, in turn, free those file storage areas for other uses. To delete a PS or PF file, perform the following. Initiate the **TAS** command by typing **TAS,3Ø** followed by a carriage return. Respond to the questions asked by the computer. Figure 15 is a typical example.

2.53 After one or more PS or PF files have been deleted, the program sequencer disc storage areas should be packed. This packing entails closing the storage area gaps between the PS and PF files. To pack the PSEQ disc storage area, initiate the **TAS** command. When the computer responds with **ENTER NUMBER:**, type **9Ø**. The packing operation will cause considerable disc activity for 1 to 5 minutes. At the beginning of the PACK operation, a warning message is printed to indicate that rebooting must be avoided during packing. When packing is complete, an appropriate message is printed.

2.54 When new PS or PF files have been created and debugged, this information is stored on the FMGR tracks in LU#43. It is advisable to consider making backup copies of LU#43 using the copy (#COP) feature. If more detailed information is needed, use the train (#TRA) feature or refer to Section 190-205-302.

3. LOCALLY FORMATTED DISPLAY SUMMARIES

A. General

3.01 The display data base (DSDB) provides the field with the ability to create up to 12 display summaries. Each summary contains up to four columns of data, and each column may contain a title. Most of the data in each of these summaries is in the form of station numbers. Like the system status summary (Fig. 16), the summaries created by the field are automatically updated with information as to which stations are in the following positions:

(M) Manually cut off

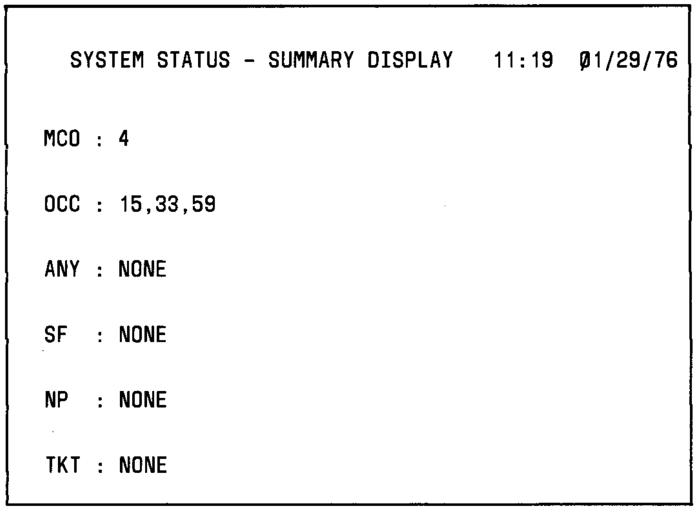
(O) Occupied

(A) With ANY alarms

(F) Station failed

(N) Nonpolling

(T) Trouble tickets assigned.



```
  SYSTEM STATUS - SUMMARY DISPLAY   11:19  Ø1/29/76

MCO : 4

OCC : 15,33,59

ANY : NONE

SF  : NONE

NP  : NONE

TKT : NONE
```

Fig. 16—System Status Summary

3.02 Figure 17 is an example of a typical summary which can be created by the field. Each column may contain a title which can be up to 14 characters in length. Each column may also contain a number of the following key words:

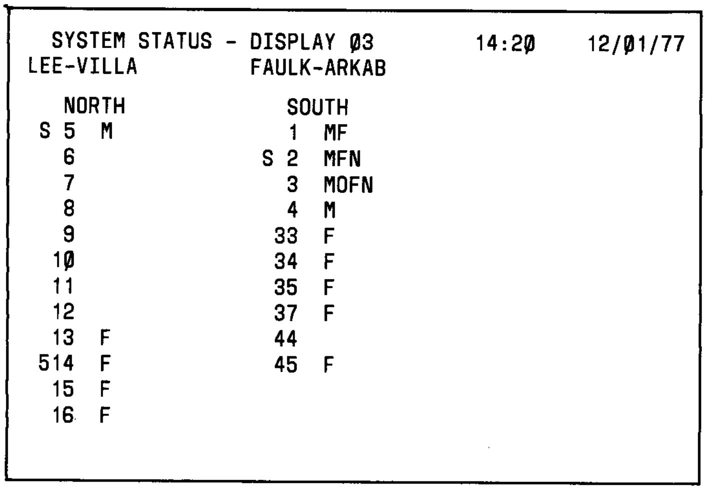
NORTH

SOUTH

EAST

WEST

These words were intended to designate direction in route-oriented summaries, but do not have to be used. Also provided is a designation symbolized by the letter "S" which specifies that the station is a switching main office. The switching office designation is entered with the station number when the display is created. Also, in order to make a display more attractive and easier to read, blank lines may be inserted into columns at appropriate points. Planning for the displays is covered in the following paragraphs. The actual entry of the data is described in Section 190-205-302.



```
 SYSTEM STATUS - DISPLAY Ø3            14:2Ø     12/Ø1/77
LEE-VILLA         FAULK-ARKAB

   NORTH              SOUTH
 S 5  M              1  MF
   6               S 2  MFN
   7                 3  MOFN
   8                 4  M
   9                33  F
  1Ø                34  F
  11                35  F
  12                37  F
  13  F             44
 514  F             45  F
  15  F
  16  F
```

Fig. 17—Example of a Display Summary

B. Planning

3.03 A certain amount of planning must be done before actually sitting down at a keyboard and entering information to create a summary. This planning should be a rough sketch of how the summary is to appear. Information such as station numbers, titles, key words (NORTH, SOUTH, etc), and which stations are switching main offices should be clearly defined in the sketch. Figure 16 may be used as a model for the sketch.

3.04 Display summaries are numbered from 1 through 12. The columns within each summary are numbered from 1 through 4 such that the left-most column is 1 and the right-most, column 4.

> ***Note:*** Note all columns of a summary have to be formatted.

3.05 Column titles in the field-generated summaries may be provided (optionally) with enhanced format; for example, blinking, highlighting, etc. For this technique, refer to the topic entitled HP 2640A CRT CONSOLE ENHANCEMENT FEATURES in Section 190-205-302.

3.06 When new displays are created, or when all displays are modified, the information is stored on the file manager tracks in LU#2. Consider making backup copies of the new look of LU#2 using the #COP feature. If more detailed information is needed, use the #TRA feature or refer to Section 190-205-302.

4. AUTOMATIC REMOTE SWITCH PROVISION

A. General

4.01 The remote switch data base (RSDB) provides the ability by which a remote switch command can be automatically sent to a remote station when an alarm or status point becomes set. (Recall that the remote switch command initiates a relay contact operation at a remote station.) RSDB maintains a file of data on the fixed disc via the RTE file manager program. This file provides a cross-reference of station number, display number, and alarm or status point number versus the switch number or C1 order code which is to be activated. This feature provides *only momentary* remote switch contact closures at E1, E2, E2A, and C1 remotes.

4.02 To use this file, an entry must be made in the SCOTS System data base. This entry (RSWT) is a processing program assigned to the alarm or status point which is called when the point becomes active. RSWT then reads the remote switch data base, selects the appropriate switch, and executes the remote switch command which activates the switch. Finally, RSWT initiates a status report for the display which indicates the status of the switch. A message is then printed indicating the name of the status point and whether the reset was successful.

B. Planning

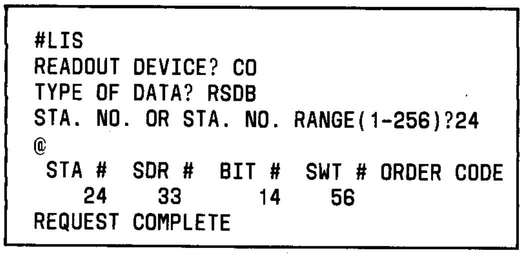
4.03 To facilitate keyboard entry, a worksheet should be prepared as shown in Fig. 18. The RSDB is organized the same as the SDR data base on a status display basis. After the worksheet is complete, or at least at a reasonable stopping point, data may be entered into the system with the **#EDI** command on the data type RSDB. Full details are covered in Section 190-205-302.

REMOTE SWITCH DATA BASE (RSDB)			
STA. NO. (1-256)	SWITCH NO. (1-4096) OR C1 ORDER CODE	SDR NO. (1-64)	BIT NO. (1-64)

Fig. 18—RSDB Worksheet

4.04 After RSDB has been formatted, it will be necessary to link the SDR data base with RSDB. For this operation, the same worksheet is used; however, the command **#EDI** is used now on the data type SDR. In editing the SDR data base, specify the processing program RSWT for every point that is to use this feature. For more details refer to Sȩction 190-205-302.

4.05 To list the contents of RSDB, type **LIS** followed by a carriage return. When the computer responds with **TYPE OF DATA?**, type **RSDB**. The computer will ask for other information pertaining to this operation (Fig. 19).



```
#LIS
READOUT DEVICE? CO
TYPE OF DATA? RSDB
STA. NO. OR STA. NO. RANGE(1-256)?24
@
 STA #  SDR #  BIT #  SWT # ORDER CODE
    24    33       14    56
REQUEST COMPLETE
```

Fig. 19—Printout Which Occurs When Listing RSDB Entries

4.06 After having entered RSDB information and/or having modified the SDR data base, use the #COP feature to make backup copies of RSDB (LU#2) and the SDR data base (LU#44). Copy details may be found using the #TRA feature or by referring to Section 190-205-302.

5. ALARM PRINTOUT ROUTING

A. General

5.01 A SCOTS central can have up to 12 attendant terminals. These terminals may be CRTs or TPTRs, and may be located at the SCOTS central location (SCOTS center) or remotely at a satellite alarm center.

5.02 It may be desirable to have several attendant terminals located at the SCOTS center. In this manner, the central can be operated from different work locations at different times of the day.

5.03 Another possibility is that one or more terminals can be located some distance from the SCOTS center. These remotely located terminals can be routed alarm messages associated with certain remote stations. This in effect is a satellite alarm center processing only those alarm messages associated with the assigned remote stations.

5.04 In order for an attendant terminal to function as an alarm center, it must receive all pertinent alarm messages. Provision to do this is via the alarm routines table (ART). ART is used to divide the status and alarm bits associated with a particular remote station into a maximum of 23 sections. Alarm and status point messages associated with each section can be sent to a specific terminal. For example, assume a particular E-telemetry remote station contains six SDRs. (Recall that an SDR is a status display report and represents a group of 64 status and alarm bits.) This example station can be divided such that all alarm and status bits between SDR1 through SDR3, bit 47, can be assigned to an attendant terminal; and SDR3, bit 48, through SDR6, bit 64 can be assigned to another. Note that only two divisions were made on this example station, and recall that up to 23 divisions can be made for each remote station.

B. Planning

5.05 A certain amount of planning should be done before actually providing for alarm printout routing. Line printers and CRT or TPTR terminals are each assigned a logical unit number (LU#) in SCOTS. Refer to Table C for a listing of the LU numbers.

TABLE C

LOGICAL UNIT NUMBER ASSIGNMENT OF ATTENDANT TERMINAL

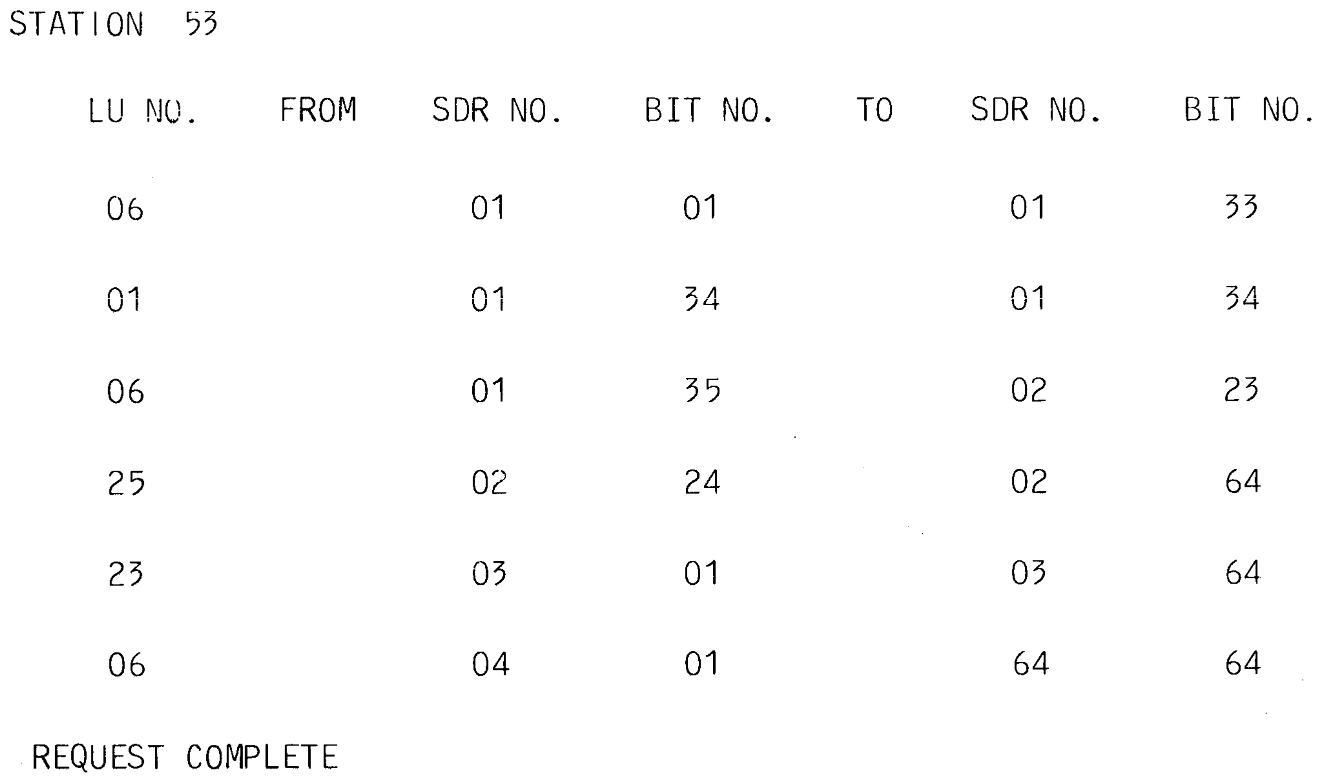
LOGICAL UNIT NUMBER (LU #)	TYPE
1	FULL
23	FULL
24	FULL
25	PARTIAL
26	PARTIAL
27	PARTIAL
28	PARTIAL
29	FULL
30	PARTIAL
31	PARTIAL
32	PARTIAL
33	PARTIAL

5.06 When planning for the alarm routing capability, the location and LU# of the attendant terminal must be known. Also, it should be planned such that each alarm message reaches the appropriate terminal where the person who can and will react is located. Similarly the operator at each attendant terminal must be aware of the responsibility being assigned to that .location to insure that alarms do not go unheeded.

Note: A status or alarm point cannot be assigned to more than one terminal.

5.07 If several points are reported as NEW in one status display, the complete identical message will be routed to all the LU numbers concerned with those points. The major alarm will be sounded on each of those LUs as well.

5.08 In preparation for entering the ART data base, plan ahead in order to be ready for questions asked by the computer when entering **#EDI** for type of data ART. The worksheet should be similar to the sample listing shown in Fig. 20.



```
STATION  53

    LU NO.    FROM    SDR NO.    BIT NO.    TO    SDR NO.    BIT NO.

     06                 01         01               01         33

     01                 01         34               01         34

     06                 01         35               02         23

     25                 02         24               02         64

     23                 03         01               03         64

     06                 04         01               64         64

  REQUEST COMPLETE
```

Fig. 20—Sample Listing of ART Data

5.09 The operator can have a continuous summary of the state of the remote stations associated with a specific terminal by creating a display summary as described in Part 3 of this section.

5.10 The alarm routing table is stored on disc on LU#44. After entering a large amount of data, consider using the #COP feature to make a disc pack backup for LU#44. For further information on the copy feature, use the #TRA file in SCOTS or refer to Section 190-205-302.

6. COMMAND FILTERING

A. General

6.01 A SCOTS central always contains one main terminal which is usually the CRT located at the SCOTS center (SCOTS central location). This terminal is referenced as LU#1.

6.02 There can be up to 12 attendant terminals connected to a SCOTS central. These can be either CRT or TPTR terminals, or simply printers with no keyboards (output only). The first terminal (LU#1) is the main console and is plugged directly into a computer I/O slot. The next six terminals interconnect to the computer via the first Hewlett-Packard I/O extender either directly or through data sets, if they are remotely located. The last five terminals interconnect with the second I/O extender. Table D provides a cross-reference of I/O port number versus LU#.

TABLE D

LOGICAL UNIT NUMBER ASSIGNMENT OF ATTENDANT TERMINALS

PORT ASSOCIATED WITH TERMINAL		LOGICAL UNIT NUMBER (LU #)	TYPE
Minicomputer Port	11	1	Full
First I/O Extender Port	26	23	Full
	27	24	Full
	30	25	Partial
	31	26	Partial
	32	27	Partial
	33	28	Partial
Second I/O Extender Port	46	29	Full
	47	30	Partial
	50	31	Partial
	51	32	Partial
	52	33	Partial
Line Printers			
Minicomputer Port	13	6	Output only

6.03 The alarm printout routing feature (discussed in Part 4) provides a way of directing certain messages to an auxilary CRT or TPTR terminal (ie, outputting to the terminal). The following paragraphs will be concerned with providing control to these terminals to perform certain keyboard operations for certain remote stations (ie, inputting from the terminal). For example, an auxiliary terminal can be provided control to initiate commands which affect only certain remote stations while commands could not be initiated on the remaining stations. This is accomplished via the command filtering table (CFT). CFT is a software package which provides a cross-reference of the attendant terminal LU# versus the remote station numbers to which the terminal can initiate commands.

6.04 The terminals fall into two classes—full-control and partial-control. There can be up to four full-control and eight partial-control terminals connected to a SCOTS central. The main console, the first two terminals in the first I/O extender, and the first terminal in the second I/O extender are the full-control terminals, whereas the rest are partial-control.

6.05 The four full-control terminals are able to perform all SCOTS functions, except that only the main console (LU#1) will output maintenance messages and run system diagnostics. Also, all data base initialization (**INI**) commands must be initiated from the main console.

6.06 The eight partial-control terminals are limited and are ***not*** able to perform the following functions:

- Load a new SCOTS System disc
- Perform data base manipulations
- Run system maintenance diagnostics
- Create program sequence files
- Start or stop polling remote stations
- Issue **RTE** commands (two-letter mnemonic commands).

6.07 The eight partial-control terminals are allowed to do the following functions on a specified set of remote stations:

- Issue switch commands
- Set station occupancy on MCO flags
- Perform log-in and logout functions.

6.08 The eight partial-control terminals have full SCOTS capabilities for performing the following functions:

- Alarm processing and analyzing
- Remote station status reports and scans
- Trouble ticket administration
- Log retrieval and entry of comments into log
- On-demand summary display or detailed displays
- Initiation of the running of a program sequencer job
- Use of the training package (ie, #TRA)
- Data base listing operations.

B. Planning

6.09 The use of auxiliary terminals must be carefully planned to match the terminal equipment with the expected responsibility at that site. If a terminal is used as a secondary SCOTS control location and has responsibility for the alarm administration of some remote stations, then the following possibilities are available.

6.10 If only a few stations with light activity are being monitored, a 110-bits-per-second (BPS) data line with a TPTR can be an auxiliary terminal. If there is expected to be a moderate amount of activity such that the TPTR is too slow, a 1200-BPS data line with a CRT and an associated printer can be used. The printer would be required if a hard copy is essential. If this auxiliary position is larger or very active, then possibly two data lines and two CRTs would be optimum.

6.11 An auxiliary terminal can be used with no remote stations designated. This terminal would not be able to effect switch closures at any remote, but could observe system functions such as the history file or the real-time status of any remote station. This could be used as an administrative terminal or for maintenance administration functions.

6.12 Another possibility is for more than one terminal to have command capability at a remote station. There can be overlapping domains of responsibility with two locations having responsibility for several remote stations.

C. Example

6.13 The following example illustrates some of the possibilities.

CONTROL	TERMINAL	LU#	CONTROL OVER THE FOLLOWING REMOTE STATIONS
FULL	HP 2640A CRT	1	ALL
FULL	HP 2640A CRT	23	ALL
PARTIAL	TTY MOD. 35 KSR	25	10-20, 39, 41
PARTIAL	HP 2640A CRT	26	21-27, 50-55, 72
PARTIAL	HP 2640A CRT	27	25-38, 40-49, 56
PARTIAL	DATASPEED 40 CRT KEYBOARD AND PRINTER	28	25-38, 40-49, 56

6.14 Without making any assignment, the main console (LU#1) has full capabilities at all remote stations. Similarly, all full-control positions such as LU#23 in the Example C have all remote stations as their range without any keyboard entry provision.

6.15 Preplanning this data base requires that a table, such as the one in Example C, be filled out. When all the data is available, the **#EDI** operation is used on the CFT type of data. Entry of the data is detailed in Section 190-205-302.

6.16 After having entered a large amount of data, consider making a backup copy of the appropriate disc (LU#44). Details of the #COP feature are located in the SCOTS #TRA files, or in Section 190-205-302.

7. PROCESSING PROGRAM FILES

A. Operating Personnel-Identification (ID) File

7.01 The ID file stores data base information for the log-in/logout (LILO) processing program. LILO provides a convenient way of tracking occupancy of normally unattended remote stations. For examples, when someone enters a remote station, the SCOTS central terminal will detect the door opening and print an open-door alarm message. After the person checks in via the order wire, the SCOTS central operator can log the individual(s) in via ID number. Similarly, personnel can be logged out. LILO tracks this logging in and out and prints message when appropriate.

7.02 The purpose of the ID file is to store the ID number(s) and names of the operating personnel. The ID numbers range from 1 through 9999. When entering a number into the ID file, each number must be a 4-digit entry. Therefore, leading zero's must be included when needed. Although there is a wide range of ID numbers, the maximum number of entries is 512. The name associated with each number can be a maximum of 16 characters. Figure 21 is a worksheet which can be copied and completed with the necessary information for the ID file. This worksheet should be completed whenever a large amount of data is being prepared. Data entries are as follows:

WORKSHEET BLOCK	CONTENT
ID NUMBER	Each entry is a number ranging from 0001 through 9999.
ID NAME	Each entry consists of a maximum of 16 characters.

7.03 Consult Section 190-205-302, SCOTS Data Base Management, for details on entering information for the ID file.

B. Open-Door Reset Switch (ODRS) File

7.04 Some open-door alarm application circuits require a switch to be operated at the remote station to reset the open-door alarm. The ODRS file stores the remote station and remote switch number to reset this alarm. This file is associated with the LILO processing program and is automatically called as required.

7.05 Figure 22 is worksheet which can be copied and completed with the necessary information for the ODRS file. (This worksheet can also be used with the ACRS file entries.) Data entries for this worksheet are as follows:

WORKSHEET BLOCK	CONTENT
TYPE OF DATA	Identify worksheet with: ODRS
STATION NUMBER	Enter remote station number
SWITCH NUMBER	Enter the switch number associated with the open-door alarm reset

7.06 Consult Section 190-205-302, SCOTS Data Base Management, for details on entering information for the ODRS file.

C. AC Restore Switch (ACRS) File

7.07 At some remote stations, the office load must be transferred back to the commercial AC source after an AC power failure has cleared. This can be accomplished automatically via the ACRS processing programs as follows. After receipt of the AC restore status indication which indicates that AC power has been restored, the ACRS processing program automatically sends a remote switch command to transfer back to command AC power. A data base entry in the ACRS file is required for each remote station with this equipment arrangement.

7.08 Figure 22 is a worksheet which can be copied and completed with the necessary information for the ACRS file. Data entries are defined as follows:

WORKSHEET BLOCK	CONTENT
TYPE OF DATA	Identify worksheet with: ACRS.
STATION NUMBER	Enter remote station number.
SWITCH NUMBER	Enter remote switch number for E1, E2, and E2A systems or C1 order codes for C1 remotes.

8. WELCOM FILES AND FILES FOR USE WITH THE MSG COMMAND

A. WELCOM

8.01 The WELCOM file is used to inform the SCOTS software about the characteristics of the various DATASPEED 40 terminals connected to the computer. The file is a transfer file stored with the file manager files on LU#2 and is actually executed like a program whenever the system is rebooted. Little detail is given on most aspects of this file because its creation and use are highly automated. Interpretation of the file, should it be necessary, is given in the following paragraphs.

8.02 For each DATASPEED 40, there are two control statements consisting of a **CN** command and three parameters separated by commas. The final statement is **EX** for exit. All statements begin with a colon. For example, refer to Fig. 23.

8.03 For planning the WELCOM file, it is necessary only to know the following information about the various DATASPEED 40 terminal:

- LU number
- HP interface card type
- Data speed
- Send/receive or receive only operation

Entry or modification of the file is described in Section 190-205-302, SCOTS Data Base Management.

B. Files for Use With MSG Command

8.04 Entry of modification of these files is covered in Section 190-205-302 along with the WELCOM file. Both the name and the contents of the file should be planned out in advance with all of the guidelines noted in paragraphs 2.07 through 2.09. File size should be kept small although the MSG feature will transmit messages up to 500 characters in length.

9. REFERENCES

9.01 The following is a list of Bell System Practices which should aid in the understanding of this section.

SECTION	TITLE
190-205-000	Transmission Maintenance System—Surveillance and Control of Transmission Systems (SCOTS) TOP
190-205-101	Transmission Maintenance System—Surveillance and Control of Transmission Systems (SCOTS) —System Description
190-205-102	Transmission Maintenance System—Surveillance and Control of Transmission System (SCOTS)—Central Terminal Description

SECTION	TITLE
190-205-300	Transmission Maintenance System—Surveillance and Control of Transmission System (SCOTS)—Central Terminal Operations
190-205-301	Transmission Maintenance System—Surveillance and Control of Transmission Systems (SCOTS)—Data Base Preparation
190-205-302	Transmission Maintenance System—Surveillance and Control of Transmission Systems (SCOTS)—Data Base Management

SCOTS ID FILE		
ID NUMBER		ID NAME
	- -	
	- -	
	- -	
	- -	
	- -	
	- -	
	- -	
	- -	
	- -	
	- -	
	- -	
	- -	
	- -	
	- -	
	- -	
	- -	
	- -	
	- -	
	- -	
	- -	
	- -	

Fig. 21—ID File Worksheet

SCOTS ODRS __________ ACRS FILE

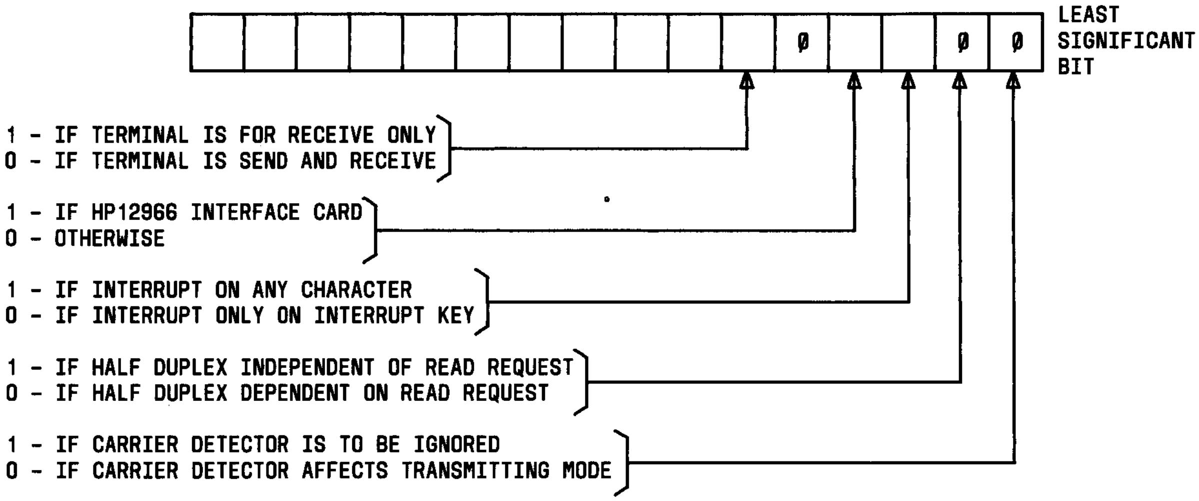
TYPE OF DATA ________________

STATION NUMBER	SWITCH NUMBER	STATION NUMBER	SWITCH NUMBER

Fig. 22—ODRS or ACRS File Worksheet

```
FIRST CONTROL STATEMENT (EXAMPLE    :CN,25,31B,14B)
                                     |  |   |
                        PARAMETER    1  2   3
```

PARAMETER 1: THE LU NUMBER
PARAMETER 2: ALWAYS 31B
PARAMETER 3: THE OCTAL EQUIVALENT OF THE
FOLLOWING 16-BIT BINARY WORD:


Ø
Ø
Ø
LEAST
SIGNIFICANT
BIT
1 - IF TERMINAL IS FOR RECEIVE ONLY
0 - IF TERMINAL IS SEND AND RECEIVE
1 - IF HP12966 INTERFACE CARD
0 - OTHERWISE
1 - IF INTERRUPT ON ANY CHARACTER
0 - IF INTERRUPT ONLY ON INTERRUPT KEY
1 - IF HALF DUPLEX INDEPENDENT OF READ REQUEST
0 - IF HALF DUPLEX DEPENDENT ON READ REQUEST
1 - IF CARRIER DETECTOR IS TO BE IGNORED
0 - IF CARRIER DETECTOR AFFECTS TRANSMITTING MODE

NOTE: PARAMETER 3 WILL ALWAYS BE ONE OF THE FOLLOWING:

14B (12966 CARD, INTERRUPT ON ANY CHARACTER)
40B (RECEIVE ONLY PRINTER, NON-12966 CARD)
50B (RECEIVE ONLY PRINTER, 12966 CARD)

```
SECOND CONTROL STATEMENT (EXAMPLE   :CN,25,30B,6)
                                     |  |   |
                        PARAMETER    1  2   3
```

PARAMETER 1: THE LU NUMBER
PARAMETER 2: ALWAYS 30B
PARAMETER 3: 3 FOR 110 BAUD
6 FOR 300 BAUD
9 FOR 1200 BAUD

Fig. 23—Example of Control Statements for DATASPEED 40