2C3

Honeywell

HONEYWELL
INFORMATION
SYSTEMS INC.

TITLE: No. 43A177886

ENGINEERING PRODUCT SPECIFICATION (EPS)
DOCUMENT HANDLER SOFTWARE FOR THE DES6000

Total Pages 319

DIST: L16-92

REVISION RECORD

REVISION LETTER	DATE	PAGES AFFECTED	APPROVALS	AUTHORITY
A	JAN 25 1972	ISSUED SH. 1-319F	1/19/72	

TABLE OF CONTENTS

1.0 INTRODUCTION

1.1 PURPOSE

The purpose of this document is to describe the organization of the 355/6000 Document Handler Software (for the DRD200 and MRS200 Reader/Sorter hardware) and its integration into the DATANET 355/6000 Communication Subsystem. The intent has been to make the software an independent package that operates within the communication system environment and, wherever possible, uses the services provided by the communications system, such as buffer allocation, inter-computer message handling, etc.

1.2 SCOPE

The Document Handler Subsystem is comprised of one or more DRD200 and/or MRS200 Document Readers, with their channel adapters attached to a DATANET 355. In turn, the DATANET 355 is attached to a 6000, through the Inter-Computer Channel. A document handler can be used in the "off-line Mode", that is, not operating under the control of the DATANET 355, for basic sorting operations. A plug-board can be "programmed" to handle certain functions for this purpose. This document will not be concerned with "off-line" operation of the reader/sorter.

For on-line sorters, the software must manage the hardware operations of start, stop, read and pocket select, analyze the data from the record to determine the proper pocket destination, and transfer detail records and control information to the 6000 for recording on mass media and later processing. The software must be capable of providing an audit trail that will satisfy the most stringent examiner. In addition, the software must attempt to compensate for problems in the hardware whenever possible.

This document will discuss in detail the hardware and software configurations necessary to accomplish "on-line" operation of the reader/sorters in the DATANET 355/6000 environment. The first section discusses the operational requirements of the document handler subsystem. This followed by descriptions of the necessary software requirements for the 355 and 6000. As much as possible, specific details about the implementation will be included. For those items where the hardware or the communication subsystem software is still under development, this document will be extended or revised as necessary.

1.3 APPLICABLE DOCUMENTS

Hardware Documentation
43A133617, PPS, MRS200, 12-Pocket Document Handler - also CPB336A
CPB1475, DRD200, COC-5 Document Reader
43A219612, EPS-1, 355 ICC - Inter Computer Channel

1.3 APPLICABLE DOCUMENTS (continued)

43A219608, EPS-1, 355 Computer
Preliminary, EPS-1, 355 Document Handler Channel
43A219613, EPS-1, DATANET 355 Document Handler Subsystem
43A291604, EPS-1, 655 Input/Output Multiplexor
43A219601, EPS-1, DATANET 355
M50EB00323, EPS-1, Two-Pocket COC-5 Document Reader

Software Documentation
CPB-1426, Dual Document Handler; Lister-Entry Shell
43A219619, EPS-1, General Remote Terminal Supervisor-355 (GERTS)
CPB-1518B, GE-625/635 GECOS III, Comprehensive Operating Supervisor
CPB-1494A GE 615/635 I/O Supervisor (SMD)

2.0 HARDWARE DESCRIPTION

2.1 MRS-200 12-POCKET DOCUMENT HANDLER

The MRS-200 hardware is fully described in CPB 336 and document 43A133617, its Product Performance Specification.

This device is used to read and pocket paper documents such as checks or cards, which have data encoded in either magnetic ink or the COC-5 print font. Pocket decision can be based upon the content of a specific character position or on field content, depending on the application. Three characteristics differentiate this device from other peripherals:

- Data characters are read in reverse sequence.
- The device is meant to run in a continuous feed mode.
- Commands can be issued to the device even though it is processing a previous command.

The first of these items should not present any particular problems to implementors. However, the last two place a "real time" requirement on the software in that the software must service the handler within a certain fixed time constraint. Generall, the software should always have a Read command out to get the next record while it is processing the data from the previous record. Failure to issue the read or pocket command within the given time limit will result in a bad status return and will require manual and software correction procedures.

The physical characteristics of the MRS-200 are as follows:

Pocket Definitions - the twelve pockets are labeled 0-9, Reject and Special. The Special Sort pocket is used for selecting records that meet pre-defined exception conditions; however, it is no different from the others in command usage. The Reject pocket also has one pocket select command available. It is used to hold the records in error, either on purpose from bad data or just the failure to issue the desired pocket command in time.

Speeds - The MRS-200 runs at a nominal speed of 1155 documents per minute, one every 52 millisecond. The transfer rate is approximately 2340 characters per second (when using COC-5 font at 10 characters per inch).

Character Set - The character set is a subset of the GE six bit graphic character set with certain redefinitions.

2.1 MRS-200 12-POCKET DOCUMENT HANDLER (continued)

Document Character	Octal Representatives	Equivalent GE Graphic
(transit)	33	.
(amount)	53	$
(dash)	73	,
(on-us)	54	*
0	00	0
.	.	.
.	.	.
.	.	.
9	11	9
unreadable	13	#

MRS-200 devices that have been modified for the COC-5 font use only the characters 0-9 and the "special", equivalent to the "dash" (octal 73).

Commands - The commands for the MRS-200 are as follows:

Octal	Command Definitions
41	Feed Continuous
61	Stop Feed
01	Read Document
43	Pocket Document
00	Request Status
40	Reset Status
62	Remote Manual Halt

The Device Code for all commands except Pocket Document must be zero. The Pocket Command uses Device Code to select a particular pocket.

Octal	Pocket Select Destination
12	Special Sort Pocket
00	Pocket Zero
01	Pocket One
02	Pocket Two
03	Pocket Three
04	Pocket Four
05	Pocket Five
06	Pocket Six
07	Pocket Seven
10	Pocket Eight
11	Pocket Nine
13	Reject Pocket

2.1 MRS-200 12-POCKET DOCUMENT HANDLER (continued)

Status Returns on Terminate - The status returns follow the formats for the Common Peripheral Interface. Codes are in binary.

Major Status	Minor Status	Definition
0000		Channel Ready
0010		Attention
	00XXX1	Feeder/Pocket Alert
	010001	Last Batch
	00XX1X	Manual Halt
	X01XXX	Feed Alert
	X0X1XX	Doc. Jam or Feed Failure
0011		Data Alert
	000001	Transfer Timing
	XXX10X	Late/No-Read Command
	XX1X0X	No Pocket Command
	X1XX0X	TCD Alert
	000010	Multiple Feed
0101		Command Reject
	X000X1	Invalid Op Code
	X0001X	Invalid Device Code
	000010	Multiple Feed
1000		Channel Busy

A unique major status bit is also defined:

1XXXXX	Special Echo

The Special Interrupt is also used to flag certain conditions in the handler.

2.2 DRD-200 COC-5 DOCUMENT READER

The DRD-200 is a two pocket document reader capable of reading COC-5 font records and, optionally, mark sense type records. Its command structure and status returns are similar but not identical to those of the MRS-200. It is intended to be an on-line device, but can be used for simple verification runs and T&D in the off-line mode. Its hardware and programming characteristics are described in CPB 1475.

Pocket Definitions - The two pockets are labeled the Accept Stacker and the Reject Stacker.

Speed - The DRD-200 runs at a nominal feed rate of 1200 records per minute, one every 50 milliseconds. The transfer rate is 2400 characters per second when reading COC-5 font, ten characters to the inch.

2.2 DRD-200 COC-5 DOCUMENT READER (continued)

Character Set -

Document	Character	Octal Representation	Equivalent GE Graphic
COC-5	MARK SENSE		
0-9	0-9	00-11	0-9
REJECT	REJECT	13	#
(HYPHEN)		73	,
	+	60	+
	-	40	
	SPACE	20	SPACE

Commands -

Commands for the DRD-200 are as follows:

Octal	Command Definition
41	Feed Continuous
61	Stop Feed
01	Read Document
43	Pocket Document
00	Request Status
40	Reset Status
62	Remote Manual Halt

The Device Code is used in conjunction with Pocket Document to determine the destination pocket number

Octal	Pocket Select Destination
12	Accept Stacker
13	Reject Stacker

Status Returns on Terminate -

The status returns for the DRD-200 are as follows:

Major Status	Minor Status	Definition
0000		Channel/Peripheral Subsystem Reading
0010		Attention
	0XXXX1	Bin Alert
	0XXX1X	Manual Halt
	0XX1XX	Document Jam
	0X1XXX	Feed Alert
	01XXXX	Last Batch

2.2 DRD-200 COC-5 DOCUMENT READER (continued)

Major Status	Minor Status	Definition
0011		Data Alert
	000001	Transfer Timing Error
	00XX10	Multiple Feed
	00X1X0	Late/No Pocket Command
	001XX0	No Pocket Command
0101		Command Reject
	0000X1	Invalid Op Code
	00001X	Invalid Device Code
1000		Channel/Peripheral Subsystem Busy

2.3 DOCUMENT HANDLER CHANNEL

The Document Handler Channel is described in the EPS-1, "GE-355 Document Handler Channel", document 43A219618 (Preliminary, dated 2/21/69).

1.4 GE-355 COMPUTER

The GE-355 Computer, comprising processor, store and IOM is described in the EPS-1, "GE-355 Computer", document 43A219608. Since this is the foundation of the system, the reader should examine the entire document and, consequently, no information has been excerpted.

2.5 INTERCOMPUTER CHANNEL

The Intercomputer Channel is described in the EPS-1, "355-Intercomputer Channel", document 43A219612. All data communication between the GE-355 and the 6000 is done through this device. The device is effectively "owned and operated" by the communications system software. The document handler subsystem will use the Intercomputer Channel indirectly through the interface defined by the communication system.

3.0 SYSTEM OBJECTIVES

3.1 PRODUCTION GOALS

This system will be designed to drive, pocket select, transfer to the 6000 and write data from six document handlers at maximum efficiency in a multiprogramming environment under control of GECOS III. The system will be designed to deliver multi-programming thru-put utilizing one or more 6000 processors, a 355 mass media storage, and magnetic tape.

3.2 OPERATIONAL CONCEPT

As envisioned for this system, the user shall be able to trigger the initiation of an entry run via a request from the 355 control console assigned to a particular handler. The user request will include directions as to which set of parameters (from the pocket select tables) is to guide the run. In response to this request, the 6000 will retrieve the designated run parameters (user pocket select tables) from the file system and transmit them to the controller. Run control will then reside in the controller until the run is completed or terminated by the user via the control console.

3.3 SYSTEM FLEXIBILITY

The system will be constructed, software-wise, in a manner which lends itself to a wide range of variations. This will be accomplished by permitting the user to develop his own pocket select tables and sort patters via pocket Select Generator. Hardware-wise the system will be designed to operate at maximum efficiency, within the hardware time restraints, using a variety of mass media devices, using one or more processors, and within any 6000 core size.

4.0 SYSTEM ORGANIZATION

4.1 DESIGN OVERVIEW

The Document Entry System will consist of three basic programs; the 355 Document Handler Control Program (355 DHCP), the 6000 Document Entry Program (6000 DEP), and the 6000 Pocket Select Table Generator (6000 PSTG). The 355 Document Handler Control Program (along with 355 GERTS) will reside in the 355, operate concurrently with the 6000 Document Entry Program, and handle all I/O and communciations (coming from and to the 6000) needed for the document handlers and 760's/TTY's. The 6000 DEP running concurrently with the 355 DHCP, residing in the 6000, will issue requests to the 355 for data from the document handlers, write the file, initiate and process messages to and from the TTY's/760's, for initiating runs, transferring pocket select table to the 355 DHCP, restart procedures, etc. These two programs will be operating in a real-time environment. The 6000 Pocket Select Table Generator will be a 6000 slave program. It will be capable of creating and maintaining the pocket select tables necessary to make pocket select decisions in the 355 DHCP. Figure 1 is an illustration of the total system.

4.2 355 DOCUMENT HANDLER CONTROL PROGRAM

The 355 DHCP will be constructed in a modular manner and written in 355 assembler language (355 MAP). It will utilize pocket select tables to modify itself to the various user sorting requirements. Users will not be permitted access to the 355 coded logic. The user will accomplish sorting requirements in the 355 through the use of the pocket select tables.

The 355 DHCP will interface with the 355 GERTS package to the fullest extent possible. It will utilize the present GERTS message logic, TTY's/760 logic, data transfer logic to the 6000, etc.

4.3 6000 DOCUMENT ENTRY PROGRAM.

The 6000 DEP will be written in COBOL with the exception of the procedural division, which due to interface requirements with GECOS III will be written in 6000 GMAP. It will be modular in form (segmented) which will permit users the option to select all or parts of the program. It will be, however, a stand alone program capable of accepting, processing, and writing data from the 355 DHCP. It will permit the addition of user logic written in COBOL segments.

4.4 6000 POCKET SELECT TABLE GENERATOR

This will be a slave program written in either GMAP or COBOL or both. It will accept user supplied table data, verify as much as possible that it is correct and create and/or update the pocket select table file. Figure 1A is an illustration of how the program will load and execute.

4.5 POCKET SELECT TABLES

The pocket select tables will be user supplied control and pocket select information which will be used by the 600 DEP and the 355 DHCP to handle the requirements of pocket selection, data verification, editing, record and block size, etc., for each individual run. This information will be used in much the same way as sort parameters are used in a generalized sort. Both the 355 DHCP and the 6000 DEP will be capable of modifying their logic, from run to run, to the specifications given on the pocket select tables.

4.6 LOADING AND INITIALIZING 355 DHCP

The DHCP along with its necessary GERTS modules will be loaded to the 355 by the 6000 to 355 bootload program. The 6000 to 355 bootload program will be spawned by the 6000 console operator. Figure 2 illustrates the 355 bootload process.

4.7 LOADING THE 6000 DEP

the 6000 DEP will be spawned by the 6000 operator from the 6000 console following the 355 bootload. The 6000 DEP will determine through a message to 355 GERTS-DHCP if it has a request to process. If there is no request it will be swapped out so that the memory core will be available for other programs. If there is a request to process, it will set its urgency to 62 to assure it will not be swapped and process the request. Figure 3 illustrates the 6000 DEP load and request process. Figure 4 shows the request and process request procedure. Note that each action required to process the request is numbered and represents the following:

1. The operator requests a job to be run by run no., run and the the messages are sent to the 6000 DEP.

2. 6000 DEP reads associated pocket select table for the requested run.

3. Pocket table and control information is sent to 355 GERTS-DHCP.

4. "Ready" message is sent to the operator.

5. Document handler is started.

4.8 TEST AND DIAGNOSTICS

The Document Entry System will be capable of running T&D for one or more handlers within the production environment. Whenever a handler is taken out of production, due to the handler malfunctioning, it will be possible to sapwn T&D (OPTS 6000) from the 6000 console and turn control of the handler over to T&D processes via 355 GERTS-DHCP.

4.9 SYSTEM INTERFACES

The following system software interfaces will be necessary:

- GECOS III
- File System
- 6000-355 Bootload
- 355 GERTS
- Test and Diagnostics (OPTS 6000)

4.10 ILLUSTRATIONS

Figure 1

Figure 1A

Figure 2

Figure 3

Figure 4

FIGURE 1

DOCUMENT ENTRY SUBSYSTEM & SOFTWARE REQUIREMENTS

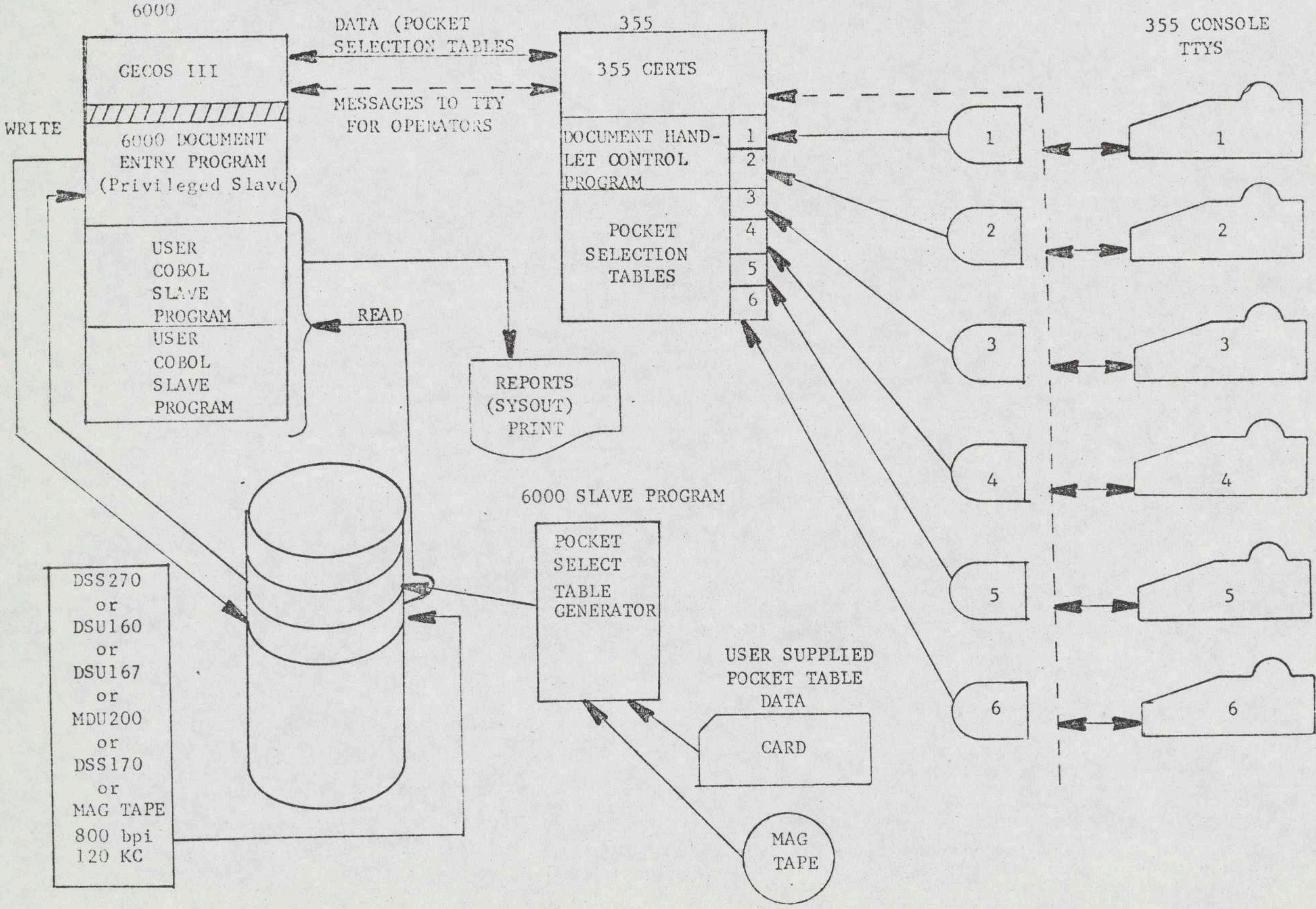


FIGURE 1A

LOADING AND RUNNING THE POCKET SELECT GENERATOR PROGRAM

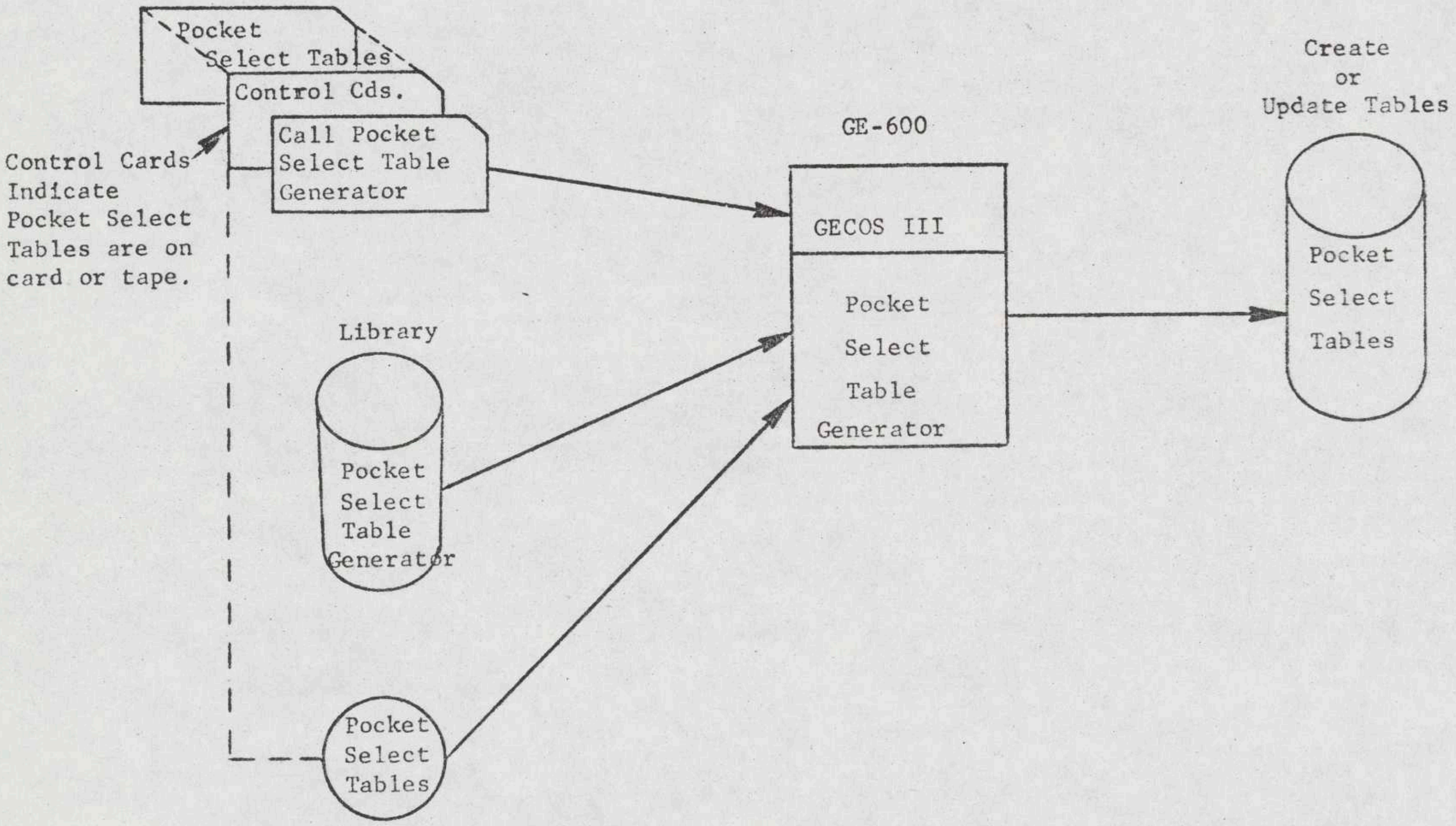


FIGURE 2

LOADING 355 GERTS WITH DOCUMENT HANDLER CONTROL PROGRAM

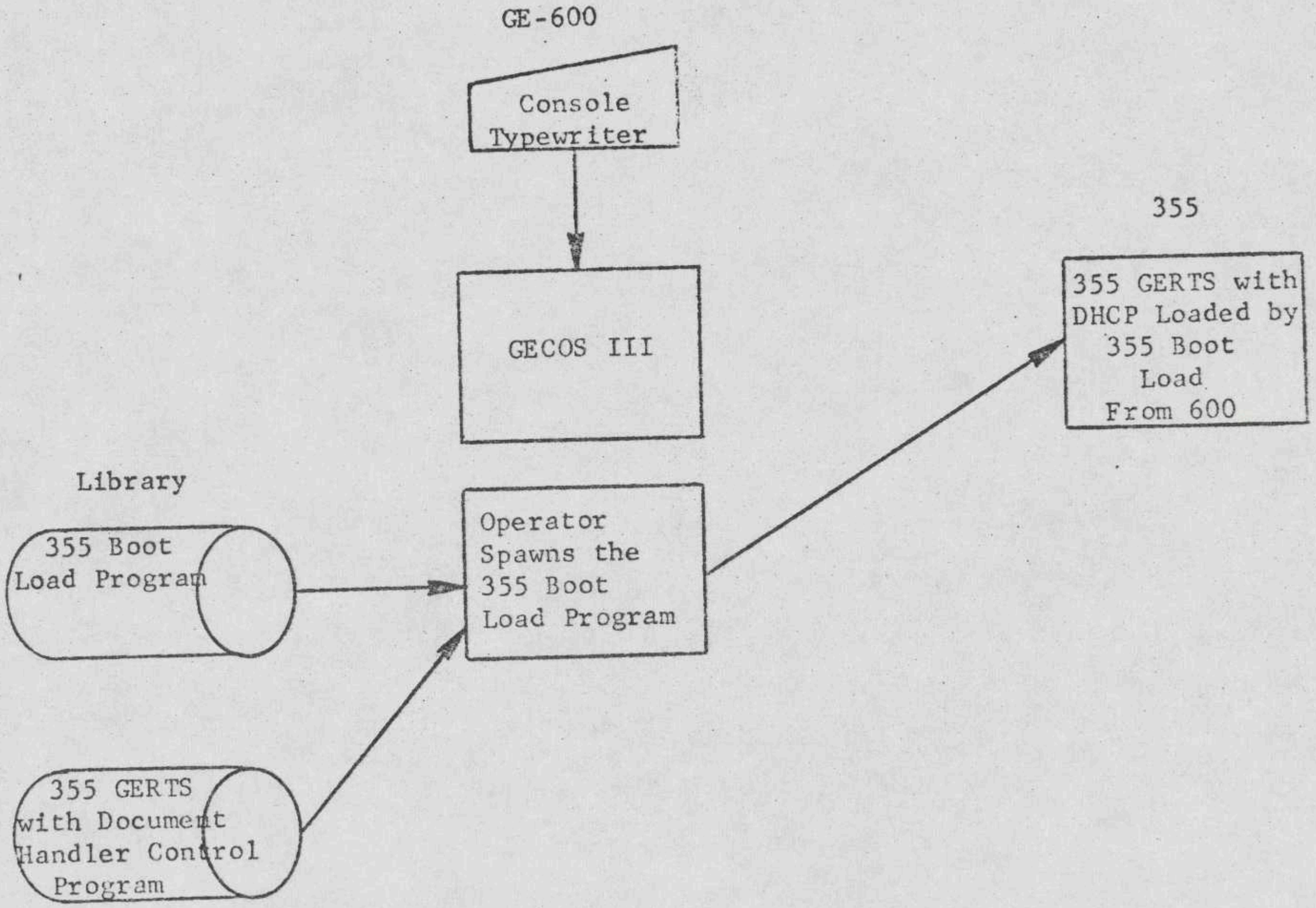


FIGURE 3

LOADING THE 600 DOCUMENT ENTRY PROGRAM

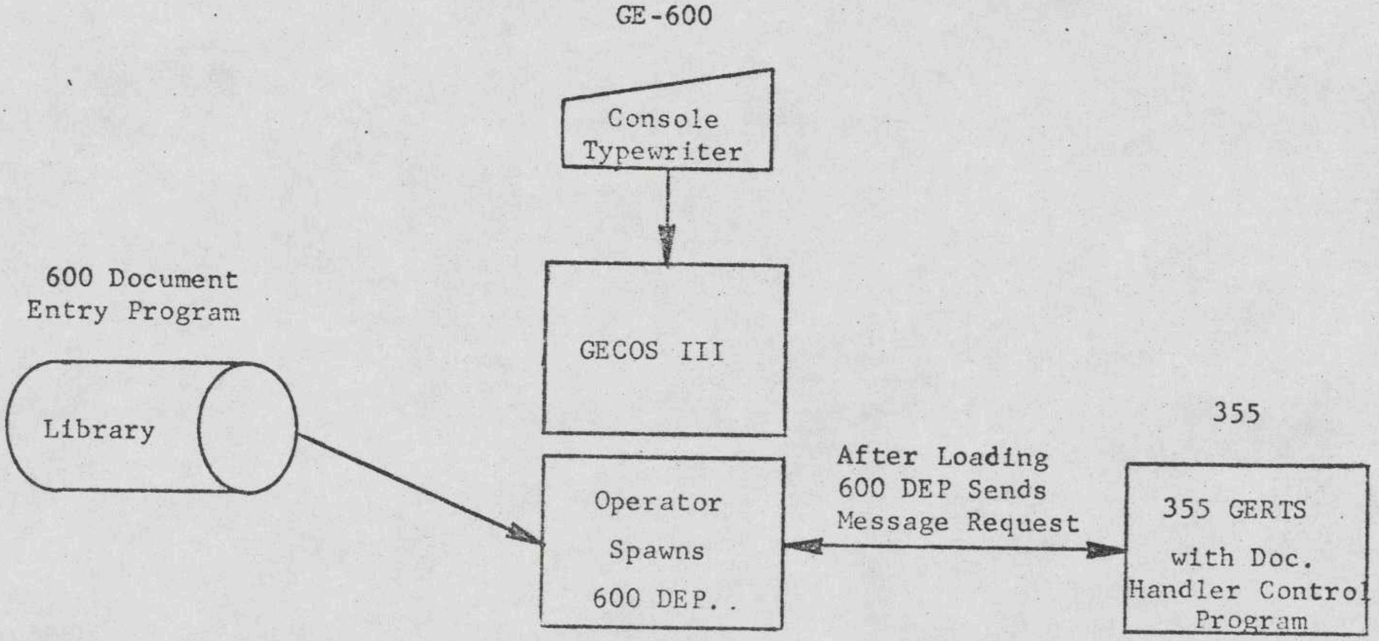
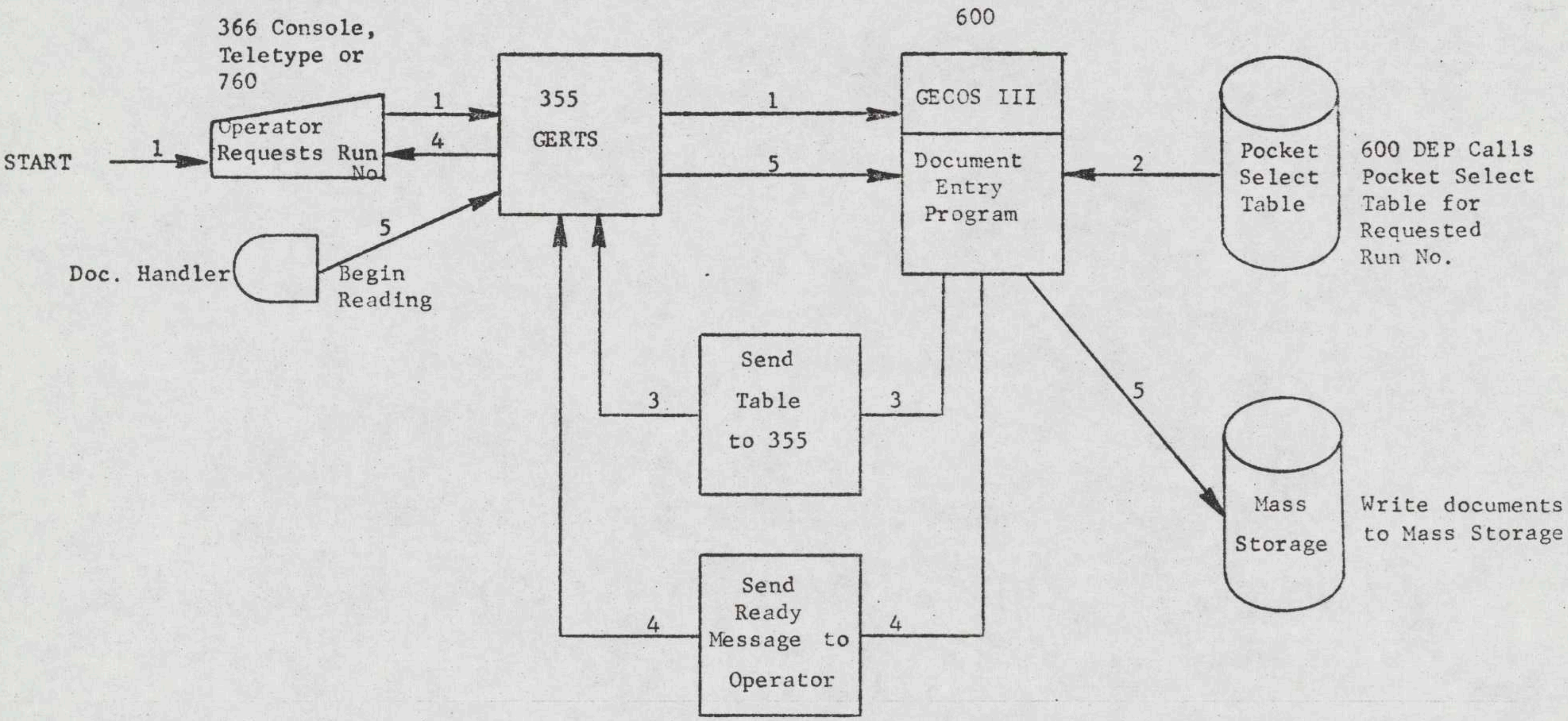


FIGURE 4

INITIATING A RUN ON THE DOCUMENT ENTRY SYSTEM



5.0 TIMING CONSTRAINTS

5.1 GENERAL

Since the Document Entry System will operate in a real time manner but under control of GECOS III, a non-real time operating system; there are some significant timing problems. Assuming that the Document Entry System will be the only real time job in the system the 6000 under control of GECOS III should handle the requirement very well concurrently with other non-real jobs. At the same time skillful and resourceful logic must be employed on the 6000 to minimize the timing problems.

Far more severe and critical are the time problems which involve the 355. While the 355 is quite capable of driving and controlling six document handlers, the added burden of pocket selection requirements raises some serious problems.

5.2 CONTROLLING AND READING THE DRD200 AND MRS200 TO THE 355

A critical timing situation that will occur in the 355 Document Handler Control Program is when six document handlers terminate read at nearly the same time. This will require the 355 to switch read buffers and reissue reads for all six document handlers within a minimum time interval. Otherwise due to the timing variations between reads (0 to an average of 12.6 milliseconds for the MRS200, and 0 to an average of 10 milliseconds for the DRD200) excessive missed reads may occur. Therefore, this area must be carefully programmed to assure that this maximum, worst case, condiction will be handled in the least possible time. If possible, this process should not exceed two or three milliseconds for the six handlers.

The second critical timing area, and probably more severe than the read problem, is pocket selection. Pocket selection must be determined and executed - command sent to the handler - within 48 milliseconds (after read complete for each handler). This means when six handlers are running there will be only an average of 48 milliseconds to make all six pocket selections or 8 milliseconds per handlers. Then subtracted from this will be time required to service any interrupts which may occur. Looking at the problem as a whole, there will be approximately 50 milliseconds to read, pocket select, transfer data to the 6000 and service interrupts for the six document handlers and other devices in service.

The pocket selection requirement will require tables which range up to 8,000,355 words. This will require a table look-up routing (it would be impossible time-wise to do a table search). Then verify, edit, pocket select or reject the documents and transfer the data to the 6000 DEP. While these requirements appear in detail in section 7, they are presented here, in a general way, to point out the magnitude of the job that will be required within the time constraints.

5.3 DATA TRANSFER FROM THE 355 TO THE 6000

Each block transferred will contain at least ten documents. This should not present a particular timing problem. On the basis of ten documents per block and six handlers running; there will be six data transfers to the 6000 every 5000 milliseconds. However, the 6000 Document Entry Program must assure the data requests are issued to the 355 at the rate of one per 500 milliseconds for each document handler and up to six requests (if six handlers are running) within 500 milliseconds.

5.4 GECOS DISPATCHING

The requests for data (MME GEROUT) from the 6000 to the 355 will be courtesy call driven. This means under the maximum load there will be six courtesy call requests and terminates every 500 milliseconds (again assuming that there are ten document per block per handler from the 355). A terminate I/O resulting in GECOS placing the 6000 document Entry Program in the C.C. queue will occur on the average of every 83.3 milliseconds. When the 6000 Document Entry Program is the only job in execution using courtesy call, it will be assured a dispatch on a consistent basis at time intervals not exceeding the maximum timer setting - 64 milliseconds times GECOS overhead of 10% (6.4 milliseconds) - times every other dispatch (2 x 64 x 110% = 140.8 milliseconds). Since the maximum time that can expire before the program will go time critical is 500 milliseconds less the time to process data and issue new data requests (estimated at 30 milliseconds); there should be sufficient time. Even with other jobs running using courtesy call there, randomly, should occur enough relinquishes (considerably short of using the maximum timer setting of 64 milliseconds) to assure adequate dispatching qithin the time constraints. Then too, if absolutely necessary, the timer setting could be reduced to assure more dispatches within the time constraints.

5.5 DATA TRANSFER FROM THE 6000 DOCUMENT ENTRY PROGRAM TO MASS MEDIA AND MAGNETIC TAPE

The 6000 Document Entry Program will delivery (write) blocks to any configured mass storage or magnetic tape subsystem. However, due to time necessary for seek and/or latency (on mass media) and start time (on magnetic tape) plus time to transfer date, the fastest possible storage devices are recommended.

The 6000 Document Entry Program can substantially control the time constraint problem by tailoring the block size to compensate for lack of the presence of the various storage devices on which seek, latency and transfer rates vary. The "link first" capability will be utilized to assure that only one seek and latency time will occur before data transfer takes place. Also, since it assumes multi-users the average seek and latency times apply.

5.5 DATA TRANSFER FROM THE 6000 DOCUMENT ENTRY PROGRAM TO MASS MEDIA AND MAGNETIC TAPE (continued)

The various devices which will be allowed are listed below along with seek, latency, transfer rates and start time for magnetic tape.

- DSS270
  Latency Time - Maximum 50.3 milliseconds
  Latency Time - Average 26.0 milliseconds
  Transfer rate - 211 KC to 320 KC

- DSU160
  Average Access Time - seek plus latency 85.7 milliseconds
  Maximum Access Time - seek plus latency 165. milliseconds
  Transfer Rate - Average - 208 KC

- DSU167
  Average Access Time - seek plus latency 75 milliseconds
  Maximum Access Time - seek plus latency 135 milliseconds
  Transfer Rate Average - 208 KC

- DSS170
  Average Access Time - seek plus latency 67.5 milliseconds
  Transfer rate - 408 KC

- High Speed Magnetic Tape 150 IPS
  Start Time - approximately 10 milliseconds
  Transfer Rate - 120 KC

Any of the above devices in a dedicated environment (the 6000 Document Entry Program is the only program working) afford ample speeds, but in a multi-programming environment (plus system usage) GECOS III I/O can soon get to be a major problem. Therefore, the block sizes should be sufficient in size to restrict write requests to a frequency of one per second. This should allow reasonable time for the real time requirement and assure the system (GECOS III) and other users have sufficient I/O Time.

6.0 SOFTWARE DESIGN REQUIREMENTS

6.1 GENERAL

There will be three separate, and rather complex, programs in the total document entry package. Two of which the 6000 Document Entry System (6000 DEP) and the 355 Document Handler Control Program (355-DHCP) will have major software requirements to assure their efficient operations under control of GECOS III and GERTS 355. The third program, the 6000 Pocket Select Table Generator (6000 - PSTG) will be difficult since it must create all of the sort parameters for the 6000 DEP and the 355 DHCP. It will not be difficult to implement within the GECOS environment since it can run as a normal slave.

The 6000 PSTG will serve to create and maintain the pocket select tables and sort tables and will not be a concern in the real time production environment, other than to generate properly formatted tables containing all necessary control and sort (pocket select) information for the 355 DHCP. Therefore, the major operational problems are involved with the 6000 DES and 355 DHCP which must work together in a real time environment under control of GECOS III.

Since the requirements for the three programs are logically separate and divided between the 6000 and the 355; they are listed below, separately by program and computer (355 or 6000).

7.0 6000 DOCUMENT ENTRY PROGRAM REQUIREMENTS

1. The program will be a privileged slave which will be initially spawned from the 6000 console by the operator by "Spawn DESXX".

1A. It will operate on a similar basis to the Time Sharing System, i.e. once spawned it will remain in execution on a swapped in and swapped out basis until the operator "kills" it or the slave itself determines it is no longer needed and does a "MME GEFINI (end of job). When it is swapped in it will be given an urgency of 62 to insure it will not be swapped out until it requests itself to be swapped out. When the slave determines it no longer has work to do (the job is not complete however) it will enter inhibited code (to prevent an interrupt), lower its urgency to zero (0) and notify the dispatcher to swap it out.

1B. Before requesting swap out the slave will assure it has an outstanding request (MME GEROUT) to the 355 DHCP for a terminal request. An answer to this request will cause it to be swapped back in; so it is essential that one request be outstanding before swapping out.

1C. The 6000 DEP program will be capable of operating as a standalone program but capable of accepting user coded modules (in COBOL) in an easy to implement manner (see item 15 and 15A).

2. It must have the capability for an increased number of I.O.Q.'s. Presently a normal slave is restricted to five (5) I.O.Q.'s; this will be increased to at least one (1) I.O.Q. for each document handler up to a total of six (6). In addition, space for six (6) I.O.Q. will be available for mass media files and one (1) 6000 console typewriter I.O.Q.

The minimum number I.O.Q.'s for the program will be twenty-four (24), (with the extra SSA this can be increased if necessary). The requirement for additional I.O.Q. space will require that this program be provided an additional SSA of 1024 words to held the I.O.Q.'s. To do this requires a logic modification in GECOS III; specifically the peripheral allocator.

3. In order to achieve the requirement listed under #2, a $ Limits card will be used with 9999 indicated as a SYSOUT requirement. This will cause a second SSA to be allocated.

   a. A routine will be written in the DEP to clear up SYSOUT interfaces in the SSA and perform b, c below.

   b. Change the I.O.Q. address pointer at .S NIO to point to the new I.O.Q. address within the second SSA.

c. Change the number of I.O.Q.'s in .S NIO from 5 to 24.

4. It will process data coming from one (1) to six (6) document handlers via 355 GERTS and operating under control of GECOS III within a multiprogramming, multiprocessor environment. This requires that the slave issue MME GEROUTS to 355 GERTS for data coming from 1 - 6 document handlers at regular intervals not exceeding approximately 470 milliseconds, for each handler and process the data within one GECOS dispatch of 64 milliseconds. Each document handler will be treated as a remote terminal within the 6000 DEP and MME GROUT's will be issued for each active handler.

5. It will utilize the GECOS courtesy call feature for all real time I/O, i.e. all MME GEROUTS and the processing of the data will be done within courtesy call. This will assure priority dispatching to the 6000 DEP and that the dispatch will not be lost until processing has been completed within the 64 millisecond time contraint.

6. The courtesy call feature will not be utilized for mass media write files.

7. All non-real time and/or functions which are not time critical will be handled at main level.

7a. The 6000 DEP courtesy call logic will be designed to service all data processes for completed I/O on each C.C. dispatch with the exception of ending courtesy call. Ending courtesy call can only be done for the I/O for which the dispatch occurred.

Whenever a courtesy call dispatch occurs to the 6000 DEP, the 6000 DEP will note which I/O (MME GEROUT) caused the dispatch. Then determine, by testing the status words of the other courtesy call driven I/O's, if there are other completed data requests. Whenever other data requests are complete it will process the data, switch buffers and issue a new data request (MME GEROUT). Each data request done in this way will be flagged to inform the program that whenever the dispatch occurs for that I/O, only an end courtesy call can be issued. When all I/O possible has been completed and flagged the program will issue an end courtesy call for the I/O that caused the current dispatch.

8. Since the 6000 DEP will operate in a multiprogram environment, where other users are using I/O or mass media as well as processor time; it will utilize the .LINKF (link first) feature to minimize the time required to transfer data to mass media. This

will also require the use of master mode in order to build I.O.Q.'s in the slave's SSA.

9. As mentioned there will be other I/O mass media users and since each mass media access has an average seek and/or latency time ranging from a minimum of 28 milliseconds for the 270's to 97.5 milliseconds for the 160's; the 6000 DEP will minimize the number of writes to mass media files to the fullest extent possible. If this is not done, it can be clearly seen that the 355 will transfer data faster than it can be written on mass media through GECOS and the 6000.

10. In view of 9 above data blocks in the 6000 for each document handler file will be sufficiently large to permit at least six (6) seconds to transfer them to mass media without interferring with the slaves' ability to handle data coming from the 355. This means that the block size should be 120 documents or records. Where the user specifies his own block size this will not necessarily apply.

11. It will handle any mix of DRD 200's, MRS 200's, and up to 32 pockets per sorter. This should not be of any particular concern to the 6000 DEP other than status tables, pocket select tables, etc.

12. All operational control messages and interpretation will be provided which enables adequate operator instructions and interface to and from the TTY's for initializing, starting, stopping, and restarting document handlers. This includes transferring sort tables to the 355 for the handlers.

13. Provide for data access from user modules by use of "labeled common".

14. The slave will be written in COBOL and GMAP. User segments will be written in COBOL.

15. It will be constructed in a manner which permits loading and processing of user constructed modules (utilizing labeled common).

16. Careful attention will be given to the size of the basic program. It should be capable of processing all basic requirements (aside from user coded modules) in 30 milliseconds. This supposes that worse case has occurred and there are six data blocks to be processed from six different handlers. This means the 6000 DEP should not have any normal process that exceeds 13,000 instruction executions. Exception processes, such as I/O errors to Mass media are not considered essential within the 13,000 executions restraint. In

this regard, any "MME" other than for I/O can be considered as using one millisecond. .LINKF will require approximately one millisecond. MME's for I/O will require about 2 milliseconds.

17. The 6000 DEP will be capable of accepting job request from TTY's, verify that the job is valid, call in the sort tables and transfer them to the 355 DHCP. The 6000 DEP will verify on status from the 355 DHCP that the job is ready to run and inform the operator via the TTY to start the handler (ready handler).

18. The slave will have the capability to spawn the 355 boot load program by a "MME" and cause the 355 DHCM GERTS package to be loaded.

19. All restart checkpoints will be kept by the slave for all document handlers in use (1 - 6).

20. The slave will keep a resource/status table which will reflect the amount of memory available in the 355, the number of document handlers in the system, their status (active, inactive, test and diagnostics in control, etc.) and when active what pocket select tables are assigned to the handlers. Also the terminal i.e., TTY's console typewriter, that is to be used on operator messages etc. for each document handler.

21. The slave will keep all subtotals and totals that are required by the application other than those that must be kept in the 355 to make pocket selections.

22. The slave will calculate a checksum on all blocks of data coming from the 355 and compare this checksum to one computed by 355 GERTS in the same manner. If the checksums do not agree; a message will be dispatched to the appropriate TTY explaining the error conditions; the 355 system informed to stop the handler from which the data came and a restart initiated for the handler.

23. The slave will keep a total of the number of couments read from each handler and a total for all handlers.

24. The format of each record will include in addition to the contents of the document:, the handler status field, a pocket selected field and a document status field. The following uses will be made, but not limited to these fields.

    a. Handler status field:

        1. The slave will monitor all possible error con-

ditions, late pocket selects, missed reads, blank or garbled field detections, hardware read errors, echo alarms, jams, transfer timing errors, feed alerts, etc. that occur and keep a total which will be monitored against a threshold. Whenever the number of errors equal a given number they will be divided into the number of documents read; when N% is reached the operator will be informed and given an option to continue processing or giving the handler to T&D. Errors will be kept by type for each document handler. When a handler is taken out of production due to excessive errors these errors will be printed out by type and total to aid T&D.

Errors due to late pocket selects, i.e. not necessarily hardware, will indicate the hardware (355) is overloaded and an appropriate reduction of work on the 355 should follow.

b. Pocket selected field:

1. This is required for later data processes.

c. Document status field:

1. This will be used in conjunction with error procedures; such as, the document which followed an error condition has been rejected and will be reread by the 355 DHCP on command from the 6000 DEP but is not an error. However, it is not to be processed by the 6000 DEP at this time.

2. A final stop status - no more data will follow until instructions are issued to the operator and the 355 DHCP to restart the handler. An example is a jam has been detected and a restart point from the 6000 DEP will be required.

In general the use of these fields, particularly the status fields, will be to keep information for restarting, error conditions, etc., which will be impossible within the 355 DHCP due to space and time limitations.

25. It will spawn T&D (OPTS. 6000) whenever requested by the operator or at least inform the operator via the 6000 console that he can spawn T&D (OPTS. 6000) for the malfunctioning handler.

26. When a handler fails during production due to a malfunction and there is another idle handler on line, and upon operator request, the 6000 DEP will transfer, processing via the 355 DHCP to the alternate handler.

27. It will contain roll back and restart capabilities which will allow restarting in the event: a handler or handlers fail, the 355 fails and/or the 6000 fails. This will be done by keeping batch check points on the mass media journal file in such a way that is can be interrogated by the 6000 DEP and determine the most current restart point for each handler. This does not pertain to normal stopping and starting of the document handlers. In those cases, restarting will be handled through document counts following the error stop and an immediate restart procedure and point given to theoperator.

    The means to enter the restart procedure will be provided by a message to the 6000 console typewriter during initialization of the 6000 DEP. A message will inquire whether the run underway is "new" or "restart". If restart, the 6000 DEP will interrogate the journal file (this will be a perm file) to find the most current restart points - initialize the 355 with the GERTS-DHCP package - inform the operators at each document handler the restart point - move the P. S. tables to the 355 and resume processing.

28. The Perm file organization will allow the 6000 DEP "all permissions" i.e. read, write, or append.

29. The 6000 DEP will spawn a reconciliation run whenever requested and there is data for the requesting spawn. The spawned program must not have a real time requirement.

30. All records will be serial numbered coming from the 355 to the 6000 DEP. (This will verify the count agrees at all times. This count will be by document handler (terminal).

31. The PERM file organization will provide onefile for each document handler.

32. Restrictions: The 6000 DEP will be the only real time job on the system when it is in core running. Batch processing can run concurrently with the 6000 DEP. It is recommended, but not prohibited, that the batch jobs running concurrently do not use excessive courtesy calls. If there should be another communication subsystem i.e. a second 355 remote batch is permissible but no real time. However, under no condition will the Time Sharing System cause the 6000 DEP to swap out.

33. When the last batch is received from the 355DHCP for all handlers the 6000 DEP will enter an end of job wrap-up concluding the "MME GEFINI". This will cause GECOS to finish the E.O.J. process and release the core space for other jobs.

34. The 6000 DEP will be resident on the system library so that it can be spawned from the 6000 console typewriter.

35. In the event the 6000 DEP aborts the slave will utilize rollback, restart procedures outlined under 27. In the event the 355 GERTS-DHCP aborts the same procedure applies as outlined under 27.

36. Whenever the 6000 DEP slave has completed processing from a dispatch it will relinquish (MME GERELC).

37. Multi-tape lister is not a requirement of this system.

38. The 6000 DEP will monitor the number of times that document handler stops occurred due to:

    a. Buffers were full in the 6000 and data could not be transferred to mass media (or tape) because of other users or I/O errors thereby causing the 355 to temporarily suspend operation on a handler or handlers.

    b. Dispatching from GECOS fell below the required time constraints which resulted in the 355 to temporarily suspend operations on a handler or handlers, due to no data requests from the 6000 DEP.

    c. Late pocket selects and missed read commands for documents coming to the 355. This would indicate the 355 is overloaded.

39. A threshold limit will be kept on the condition listed under 38a, 38b, and 38c and where these conditions become excessive the following steps will be taken:

    a. The operator will be notified that the work load is excessive and the real time application is seriously degraded and suggesting to him if possible, to kill some of the 6000 jobs (other than the DES). If the operator responds that production is acceptable then the actual performance will be considered an acceptable threshold and production will continue on that basis with the exception that the new threshold (the actual one) will be monitored and if production again decreases - say 10% - the process outlined under b will be repeated.

b. The operator will be informed as to which system, the 6000 or 355, or both, is overloaded. This will allow him to make a better judgement as to what jobs to kill; whenever it is required.

40. The program will be capable of stopping and restarting a run coming from any one of the six handlers on operator request.

41. It will be capable of killing a run coming from any one of the six handlers on operator request. This includes rolling back the file for the job which was killed.

42. At end of job an accounting report will be provided.

43. It will be capable of varying the block and record size for each run according to user supplied data on the pocket select table.

7.1 DOCUMENT STATUS WORDS 355-6000 DEP

Documents will be stored in a standard 355 GERTS block which consists of 32 18-bit words. The first nine words of each block will be made up of status information concerning the data and the status of the handler. The first four words of the eight are reserved for the GERTS 355 system while the last five will be used by the 355 DHCP.

<table>
<tr><th>355 Words</th><th colspan="8">0</th><th>17</th><th>6000 Words</th></tr>
<tr><td>1</td><td colspan="9">Reserved for GERTS</td><td>*. Upper</td></tr>
<tr><td>2</td><td colspan="9">Reserved for GERTS</td><td>*. Lower</td></tr>
<tr><td>3</td><td colspan="9">RCW
No. of Words in Record</td><td>1. Upper</td></tr>
<tr><td>4</td><td>L
D
0</td><td colspan="4">RCW
1 9 10</td><td>355
ST.
14</td><td>DH
ST.
15</td><td>6000
ST.
16</td><td>P
E
17</td><td>1. Lower</td></tr>
<tr><td>5</td><td colspan="4">Handler Status Word
Terminate Read</td><td colspan="5">No. of Characters
in Document</td><td>2. Upper</td></tr>
<tr><td>6</td><td colspan="4">Handler Status Word
PKT Select Status</td><td colspan="5">Pocket Doc.
Put In</td><td>2. Lower</td></tr>
<tr><td>7</td><td colspan="2">355 Scan 0-5
Comm. Status</td><td>6-11
User Error Code</td><td>12
CP</td><td>13
I1</td><td>14
I2</td><td>15
I3</td><td>16
DS</td><td>17</td><td>3. Upper</td></tr>
<tr><td>8</td><td colspan="2">User Message
#1 0-5</td><td>User Message
#2 6-11</td><td colspan="6">Reserved
12 17</td><td>3. Lower</td></tr>
<tr><td>9</td><td colspan="2">Float
0-5</td><td colspan="7">Send Point
6 - 17</td><td>4. Upper</td></tr>
<tr><td></td><td colspan="9">Data
23 18-bit words - 11-1/2 36-bit words</td><td>4. Lower</td></tr>
</table>

Note: all fields in binary except data.

* BLOCK ONLY. 1-4 in front of each record in the block. Upper half of word, in 6000, will contain number of words in the block. The number of words in the block does not include the block word count word.

Cont. on Page 35

7.2 DESCRIPTION OF DOCUMENT STATUS WORDS FOR 355/6000

Word 1. Reserved for GERTS

Word 2. Reserved for GERTS

Word 3. No. of words in each record within block.

Word 4. LD. - Last document indicator. Set to 1 = no more documents will be sent from 355 until handler is started again.

355 ST. - 355 stop indicator. Set to 1 = 355 stopped the 355 to indicate a stop on no more buffers available in 355, etc.

DH ST. - The document handler caused the stop as a result of 1. A manual stop, 2. a jam, 3. no pocket command, 4. Feeder/Pocket Alert, 5. Feed Alert. The handler status fields in word 5 and 6 0-9 will have to be tested to determine the exact reason for the stop.

6000 ST. - The 6000 stop indicator. Set to one indicates the 6000 caused the stop.

P.E. - Parity Error. The handler was not stopped but this document was read in error. However, it may have been read properly and a parity error was indicated. In any case, the 355 DHCP will attempt to process it.

Word 5. bits 0-9. Handler status at terminate read time. 6000 DEP will test this word for approximate error control. See error conditions, page 9 of EPS.

Word 5. bits 10-17. No. of characters read on document.

Word 6. bits 0-9. Handler status at pocket select terminate time. 6000 DEP will test this for appropriate error control. See error conditions, page 9 of EPS.

Word 6. bits 10-17. Pocket no document selected to by 355 DHCP.

Word 7. 355 Scan (355 DHCP) communication to 6000 DEP.

Cont. on Page 36

Codes	Description
17	The handler has been halted through the users sort control. DEP must look at word 8; bits 0-5, and 6-11 to see if a type out is required.
18	A user type out has been requested. The 6000 DEP must get the type out ID no. from word 8, bits 0-5 and 6-11.
21	The End of Sort Run has been given by the user. The 6000 DEP must look at word 8, bits 0-5 and 6-11 to see if a user type out is required. The DEP will type out a system message for end of run------- "*ENDRUN XXX-yy" XXY = Job # yy = handler #

Word 7. bits 6-11. User error code 1 - 63. The DEP will place this in the header of each document.

Word 7. bit 12. Set to 1 = a check point document. The DEP will place an asterisk in the header record in document type field.

Bit 13-15 Set to 1 = a bill separator/or I1-I3 type separator has been manually placed in the given pocket preceding the document. The DEP will set the document type field in the header to 1 if bit 13 = 1, to 2 if bit 14 = 1, to 3 if bit 15 = 1.

Word 7. bit 16 = Document status.
Set to 1 = Formatted document.
Set to 0 = Non-formatted packed on one data.

Word 8. bit 0-5. Message index number to locate a message that the 6000 DEP must type out on the TTY for the handler.

bits 6-11. Second message index to locate a message that the 6000 DEP must type out on the TTY for the handler.

Note: Only two messages per document can be typed out at the users request. Bits 0-5 and 6-11 are used in conjunction with word 7, bits 0-5.

Word 9. bits 0-5. Float code 1-63 will be placed in the document header by the DEP. Field obtained from the FRD-ABA table.

Word 9. bits 6-17. Send point code 001-1000 will be placed in the document header by the DEP. Field obtained from the FRD-ABA table.

7.3 MASS STORAGE FILE REQUIREMENTS

- Rollback Restart

Due to the fact that temporary files are lost when the 6000 fails (a system disaster occurs), the document entry files must be Perm files. This will assure a restart can be initiated through the 6000 DEP in the event of a system disaster. Magnetic tape should not present particular problems as the tapes can simply be dismounted and then remounted during the 6000 DEP restart procedure.

Establishing Perm Files

The user must determine his file requirements as to number of files and amount of space. He must establish file space through the DES FILE SYSTEM procedures.

- Block and Record Control

The blocks and records will conform to all standard "File System" requirements. This includes block and record control fields and formats.

- File Overflow

Since there is always the possibility of exceeding the alloted mass storage file space, the capability will be provided for file overflow. This will be provided by allowing the user to get more space or "pause" until file space becomes available.

- Mirror File

The capability to write a mirror file will be provided. This is necessary in the event of a hardware failure occurring on the prime storage device. The mirror file will be a duplicate of the file or files being created.

7.4 DES FILE SYSTEM

The DES file system will be PERM file system named "DES", i.e., the user Master Catalog name. The DES will contain six major sub catalogs. Such catalogs will be named as follow:

*1. DESCMM - DEP to slave communications.

**2. DESCMN - Slave to DEP communications.

3. SPNXXX - These files will contain all user slaves that the DEP will spawn. XXX will be numeric 001-999. The PSTG will create and write these files. The DEP will read these files with MME GENEWS.

4. TABXXX - These files will contain the assembled sort programs, with the exception of master tables. Master Tables will be filed on the MSTXXX files. The PSTG will create and write the assembled scan records and general tables on these files. XXX will be the user's sort job number, i.e., TAB001-TAB999. The DEP will read these files whenever a sort job is called for from the TTY's.

5. MSTXXX - These files will contain the sort jobs master tables. The PSTG will create and write the assembled master tables on these files. XXX will be the user master table name, i.e., 01-99. The DEP will read these files when cross referenced from the TABXXX files, and include the master table in the requested sort job.

6. DESXXX - The DEP will create and write sorted data to these files. The size and name of these files will be provided on the TABXXX files as part of the user's sort job parameters. The DEP will also release these files when requested by the user.

Other than DESCMM and DESCMN the subfiles will be created and released dynamically through the use of a DES master mode routine. The file building routine must do the following:

a) Build PAT pointers for the files in the SSA.

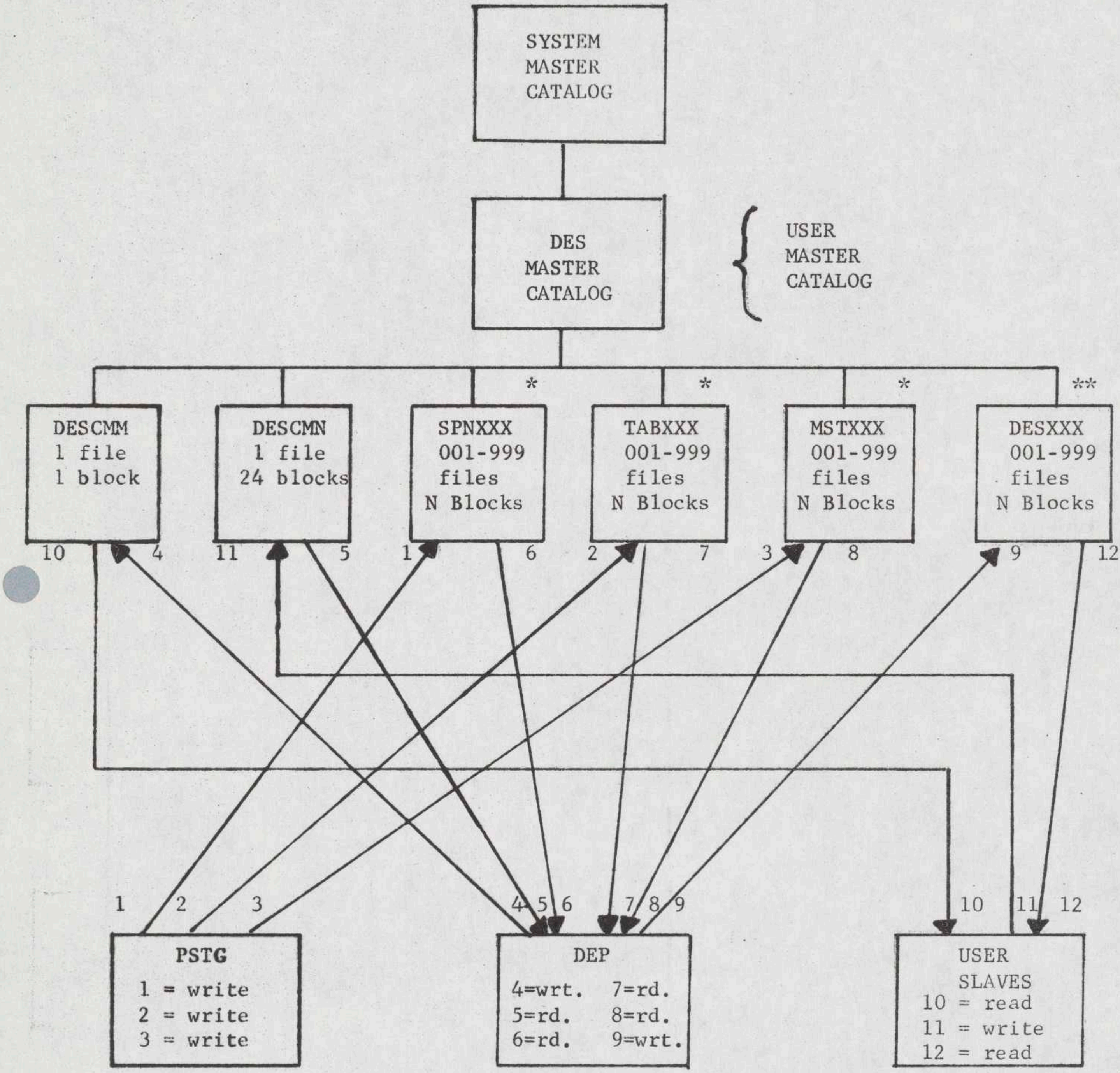
b) Build PAT bodies.

c) Enter file names and spare requirements in the Catalog System.

* DESCMM will not be created through master mode routine. It will be a normal perm file consisting of one block (320 words).

** DESCMN will not be created through the master mode routine. It will be a normal perm file consisting of 24 blocks (24X320 words).

7.5 ILLUSTRATION OF DES FILE SYSTEM



* created by the PSTG using the master mode file create routine

** created by the DEP using the master mode file create routine

Note: XXX = file name with any subname.

7.6 JOB FILE LIBRARY HEADER AND INDEX (TABXXX FILES)

	0 17	18 35
0	JOB NO. (BCD) 001 - 999 or ALFA.	355 Core Req. (Bin) for this Job.
1	Device Type MRS 200 - DRD 200 - H236 (BCD)	
2	(Bin) Output Record Size (in 6000 words 1 - 18	No. of Files (Bin) (Sort-files) 1 - 6
3	Shared or Private files	USER or DES supplied
4	File #1 File code	Seq. or random 1 = seq. 2 = random
5	Min. File Size in Links	Max. File Size in Links
6	File Name for File #1 (only if user supplied)	
7	File #2 File code	Seq. or random
8	Min. File Size in Links	Max. File Size in Links
9	File Name for File #2 Only if User Supplied	
10	File #3 File Code	Seq. or random
11	Min. File Size in Links	Max. File Size in Links
12	File Name for File #3 Only if User Supplied	
13	File #4 File Code	Seq. or random
14	Min. File Size	Max. File Size
15	File Name for File #4 Only if User Supplied	
16	File #5 File Code	Seq. or random 1=seq/2-random
17	Min. File Size in Links	Max. File Size in Links
18	File Name for File #5 Only if User Supplied	
19	File #6 File Code	Seq. or random 1=seq/2=random
20	Min. File Size in Links	Max. File Size in Links
21	File Name for File #6 Only If User Supplied	
22	File Code for DES Comm. File	Seq. or random 1=seq/2=random
23	File Name "DES CMM" (1 LLINK)	
24	File Code for User Comm. File	
25	File Name for User Comm. File - "DESCMN" (24 LLINK)	
26	Zero	

	0 17	18 35
27		
28	Master File Name No. 1. MXX.M99 or MXX	No. of 18 bits words used
29	Master File Name No. 2. MXX	No. of 18 bits words used
30	Master File Name No. 3. MXX	No. of 18 bits words used
31	Master File Name No. 4. MXX	No. of 18 bits words used
32	Master File Name No. 5. MXX	No. of 18 bits words used
33	Master File Name No. 6. MXX	No. of 18 bits words used
34	UA	UA
35	UA	
36	UA	
37	UA	
38	UA	
39	UA	
40	Block No. (320 word) Loc. of Start of SCAN recs.	Starting word in block (36 bit)
41	No. of 18 bit words used for SCAN recs.	Starting 355 Loc of SCAN records
42	Block No. (320 word) Loc of private tables	Starting word in block (36 bit)
43	Block No. (320 word) Loc of start of Private table	Starting word in block (36 bit)
44	No. of 18 bit words used for SCAN recs.	Starting 355 location of tables
45	Block No. (320 word) Loc of SAVE Area	Starting word in block (36 bit)
46	No. of 18 bit words	Starting 355 location of hold area.
47	UA	
48	UA	
49	UA	
50	UA	
51	UA	
52	UA	

	0			35
53	UA			
	0	17	18	35
54	Block No. (320 word) loc of start of pocket tables		Starting word in block (36 bit)	
55	No. of 18 bit words used for pkt tables		Starting 355 location	
56	Block No. (320 word) where SPAWN Job CTL CDS Loc.		Starting word in the block	
57	No. of ctl cds 84 (14 words)			
	S, M or N	SPAWN Code 1, 2	Cond. Code	Condition code = 1 = checkpoint 1 immediate, 3 = END JOB,4=record count
	IDENT # of SPAWN JOB			
	DES File			
	SPAWN JOB # 2			
	"			
	"			
63	"			

64 Assembled program for 355.

7.7 MSTXXX FILES (FRD-ABA MASTER TABLE INDEX)

	0	35
Word 0	RESERVED for I/O	
1	0 No. of Jobs Indexed 1-60 17	18 TABLE TYPE 1 or 2 35
2	Length of Table in 36 bit words	Length of Table in 18 bit words format for 355
3	No. of Arg's in Table-FRD-CITY/STATE-ABAS	Big Item Pkt
4	Table Name "MSTXXX"	
5	RESERVED	
6	* JOB # For 1st Job	HI-DOLLAR
	↓	
54	JOB # For 48th Job	HI-DOLLAR
63	RESERVED	

* Job No.'s correspond to location at PKT.
Fields in segments.

Example: Job located in word 6 bits 0-17 will have its pocket no's. placed in 1st PKT word bits 0-5, Job # in word 6 bits 18-35 will have its pkt no's. in PKT word bits 6--11, etc.

7.8 FRD-ABA TABLE FILE-SEGMENT NO. 1 (FRD NUMBERS)

<table>
<tr><td></td><td>0</td><td></td><td>17</td><td>18 23</td><td></td><td>35</td></tr>
<tr><td>0</td><td colspan="3">* No. of FRD Entries in Table</td><td>Seg. 1</td><td colspan="2">Length of Entry</td></tr>
<tr><td>1</td><td colspan="3">No. of FRD Entries in Seg.</td><td colspan="3">Base Address for Cal. in S.P.</td></tr>
<tr><td>2</td><td colspan="3">FRD Entry No. 1</td><td colspan="3">Pointer to City/State Segment</td></tr>
<tr><td>3</td><td colspan="3">Time Codes Job 1</td><td colspan="3">Time Codes Job 2</td></tr>
<tr><td>4</td><td colspan="3">" " " 3</td><td colspan="3">" " " 4</td></tr>
<tr><td>5</td><td colspan="3">" " " 5</td><td colspan="3">" " " 6</td></tr>
<tr><td>6</td><td colspan="3">Float and S.P. Job 1</td><td colspan="3">Float and S.P. Job 2</td></tr>
<tr><td>7</td><td colspan="3">" " " " 3</td><td colspan="3">" " " " 4</td></tr>
<tr><td>8</td><td colspan="3">" " " " 5</td><td colspan="3">" " " " 6</td></tr>
<tr><td>9</td><td>Job 1 pkt</td><td>Job 2 pkt</td><td>Job 3 pkt</td><td>Job 4 pkt</td><td>Job 5 pkt</td><td>Job 6 pkt</td></tr>
<tr><td>10</td><td colspan="3">No. of FRD Entries in Seg.</td><td colspan="3">Base Address for Cal. S.P.</td></tr>
<tr><td>11</td><td colspan="3">FRD Entry No. 2</td><td colspan="3">Pointer to City/State Segment</td></tr>
<tr><td></td><td colspan="6"></td></tr>
<tr><td></td><td colspan="6">↓</td></tr>
<tr><td></td><td colspan="6"></td></tr>
</table>

* First segment only.

7.9 FRD-ABA TABLE FILE SEGMENT NO. 2 (CITY/STATE CODES)

<table>
<tr><th></th><th colspan="3">0 17</th><th>18 23</th><th colspan="2">35</th></tr>
<tr><td>0</td><td colspan="3">* No. City/State Codes in Table</td><td>Seg. 2</td><td colspan="2">Length of Entry</td></tr>
<tr><td>1</td><td colspan="3">No. City/State Codes in Seg.</td><td colspan="3">Base Address for Cal. S.P.</td></tr>
<tr><td>2</td><td colspan="3">City/StateCode No. 1</td><td colspan="3">Pointer to ABA No's.</td></tr>
<tr><td>3</td><td colspan="3">Time Codes Job 1</td><td colspan="3">Time Codes Job 2</td></tr>
<tr><td>4</td><td colspan="3">" " " 3</td><td colspan="3">" " " 4</td></tr>
<tr><td>5</td><td colspan="3">" " " 5</td><td colspan="3">" " " 6</td></tr>
<tr><td>6</td><td colspan="3">Float S.P. Job 1</td><td colspan="3">Float S.P. Job 2</td></tr>
<tr><td>7</td><td colspan="3">" " " 3</td><td colspan="3">" " " 4</td></tr>
<tr><td>8</td><td colspan="3">" " " 5</td><td colspan="3">" " " 6</td></tr>
<tr><td>9</td><td>Job 1 pkt</td><td>Job 2 pkt</td><td>Job 3 pkt</td><td>Job 4 pkt</td><td>Job 5 pkt</td><td>Job 6 pkt</td></tr>
<tr><td>10</td><td colspan="3">No. City/State Codes in Seg.</td><td colspan="3">Base Address for Cal. S.P.</td></tr>
<tr><td>11</td><td colspan="3">City/State Code No. 1</td><td colspan="3">Pointer to ABA No's.</td></tr>
<tr><td></td><td colspan="3"></td><td colspan="3"></td></tr>
<tr><td></td><td colspan="3"></td><td colspan="3"></td></tr>
</table>

↓ Next Seg.

* First Seg. only.

7.10 FRD-ABA TABLE FILE SEGMENT NO. 3 (ABA NO'S.)

<table>
<tr><td></td><td colspan="3">0 … 17</td><td>18 … 23</td><td colspan="2">… 35</td></tr>
<tr><td>0</td><td colspan="3">* No. ABA No's. in Table</td><td>Seg. 3</td><td colspan="2">Length of Entry</td></tr>
<tr><td>1</td><td colspan="3">No. ABA No's. in Seg.</td><td colspan="3">Base Address for Cal. S.P.</td></tr>
<tr><td>2</td><td colspan="3">ABA Code No. 1</td><td colspan="3">Zero</td></tr>
<tr><td>3</td><td colspan="3">Time Codes Job 1</td><td colspan="3">Time Codes Job 2</td></tr>
<tr><td>4</td><td colspan="3">" " " 3</td><td colspan="3">" " " 4</td></tr>
<tr><td>5</td><td colspan="3">" " " 5</td><td colspan="3">" " " 6</td></tr>
<tr><td>6</td><td colspan="3">Float & S.P. Job 1</td><td colspan="3">Float & S.P. Job 2</td></tr>
<tr><td>7</td><td colspan="3">" " " " 3</td><td colspan="3">" " " " 4</td></tr>
<tr><td>8</td><td colspan="3">" " " " 5</td><td colspan="3">" " " " 6</td></tr>
<tr><td>9</td><td>Job 1 pkt</td><td>Job 2 pkt</td><td>Job 3 pkt</td><td>Job 4 pkt</td><td>Job 5 pkt</td><td>Job 6 pkt</td></tr>
<tr><td>10</td><td colspan="3">No. ABA No's. in Seg.</td><td colspan="3">Base Address for Cal. S.P.</td></tr>
<tr><td>11</td><td colspan="3">ABA Code No. 1</td><td colspan="3">Zero</td></tr>
<tr><td></td><td colspan="6"></td></tr>
<tr><td></td><td colspan="6"></td></tr>
</table>

↓ Next Seg.

* First Seg. only.

7.11 DESXXX FILE TYPES (SORTED FILES)

DEP files will be written in random fashion. There are three types of files the user may choose from for his file needs. These are defined as follows:

M-FILE - a multiplexed random-indexed single job file. From one to six sorters may write on this file so long as the same job number is being run on the sorter. Whenever a sort job is called for using an M-FILE the DEP will determine if a file already exists for the sort job; if so, the DEP will determine the next available write block on the existing file and start the job. The DEP, prior to starting the job, will place the job in the file index. The file index will consist of the first 320 word block on the file, as seen in "file index". Either DEP or the user may name this file depending on the user's choice, see "FILE Control Verbs."

N-FILE - a multiplexed random-indexed multiple job file. From one to six sorters running the same or different jobs may write to this file. Whenever a sort job is called using an N-FILE the DEP will determine if the file index will be updated and the job commenced. Data will be filed on the next available block in the file and continue in a random manner. The user must name this type of file DES 100 - DES 999, also see "FILE Control Verbs." The first block on this type of file (320 words) will contain the job file index.

S-FILE - a single job non-multiplexed file. Only one sorter with one job may access this file. It will contain a job file index. The user must let DEP name this file DES 001 - 099 and DEP must spawn a job to use the file sometimes between sort start and sort end of job.

SORTED INDEX FILE

The first block (320 words) of each Random file, that DES has written will contain a file index. The file index will contain a two word entry for each job on the file. The entry will contain the job number, a unique job snumb, the sorter the job was run on, the starting address in the file of the job, and a pointer to other entries which contain the same job number but run under a different snumb. The maximum number of entries will be 156 per file.

7.12

File Index

0 35

0	Reserved for I/O	
1	0 No. of Job Written to File 17	18 Next Available Entry Location 35
2	Zero	
3	Zero	
4	Zero	

JOB INDEX ENTRY

5	JOB NO. 0 17	SNUMB Sorter # 1-6 18 29 30 35
↓ 316	Relative Block No. of Starting Location	Chain Pointer to Next Entry for Same Job

DES 6000 LOGICAL HEADER AND DATA RECORD

Cont. on Page 49

FIELD DESCRIPTION CODE	A	B		C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
FIELD NAME	SORT JOB NO	SORT JOB SNUMB		YEAR	MONTH	DAY	SORT TIME	RUNST	DOCT.	DOC.ST.	USER.E.C.	NO CHAR IN DOC	PKT #	DOC SEQ NO.	FLOAT	SEND PT CODE	FILE SORTER CKPT SEQ NO.	FILE SORTER BLOCK SEQ. NO.	FILE BLOCK PTR TO NEXT BLOCK	DATA
		DH#	SNUMB																	
LOC & NO CHARACTERS IN FIELD	1 2 3	4	5 6	7	8	9	10 12 11	13	14	15	16	17 18	19	20 22 24 21 23	25	26 27	28 30 29	31 33 32	34 36 35	*
STORAGE MODE	BCD	BCD	BIN	BIN	BIN	BIN	BIN	BCD	BCD	BCD	BIN	BIN	BIN	BIN	BIN	BIN	BIN	BIN	BIN	BCD
6000 Words	1			2				3						4	5			6		*

* S = DATA FROM WORDS 7-18 (37-108) FORMATTED OR PACKED

7.14 DES6000 HEADER FIELD DESCRIPTIONS

A. SORT JOB NO. The number which identifies the sort job called by the operator at the TTY servicing the document handler.

B. SORT JOB SNUMB. Unique number assigned by the DES sort program to identify the run to a particular handler and run. A new SNUMB will be assigned each time the job No. given under A is called. Character four of this field will contain the document handler, 1 - 6, and five and six a unique number within the job under A. SNUMBs are assigned in ascending sequence.

C., D., and E.
YEAR, MONTH and DAY. From the GECOS operating option the day is obtained via MME GETIME then converted to binary. See MME GETIME page 204 CPB 1518B.

F. SORT TIME. The time is obtained from MME GETIME. See CPB 1518B page 204.

G. RUN STATUS. This field will always contain a zero if the sort run is continuing satisfactorily, i.e., no restarts, no aborts etc. However, if the run has had to be restarted or the job has been killed by the operator this field will contain the following codes.

* 1. = A restart has been initiated. The file has been rolled back to the last checkpoint.
NOTE: The user should always retain the block no. which contains the last checkpoint. This can be done by retaining field R from each previous block read.

2. = This run has been killed by the sorter operator.

* Normally the user will not detect a 1 code unless he is processing on a non checkpoint basis because after a roll back restart this field will again contain a zero (0).

NOTE: This field will be set by the TTY messages from the operator "KILL XXX-YY" and "RESTART XXX-YY".

H. DOCUMENT TYPES: The document type field will contain the following codes to indicate the following conditions

DES6000 HEADER FIELD DESCRIPTIONS (continued...)

H. DOCUMENT TYPES (continued...)

Code	Description
1 =	A bill separator, I1 type, was placed in the pocket given in field L prior to this document being read.
2 =	Same as above for I2 type separator.
3 =	Same as above for I3 type separator.
* =	This is a checkpoint restart document.

NOTE: This field is set from the document status words. Word three bits 30 - 33.

bit 30 on = ckepoint = *
bit 31 on = I1 separator = 1
bit 32 on = I2 separator = 2
bit 33 on = I3 separator = 3

I. DOCUMENT STATUS. This field will contain the following codes and meaning.

Code	Description
0	Data contained in 37-108 is in packed mode
1	Data contained in 37-108 is formatted per users sort logic.
2	A double feed has occurred document to garbage in packed mode.

NOTE: This field set from word three bit 34 of document status, or if a double feed from the read status word.

J. USERS ERROR CODE. Will contain a number 1 - 63 which the user has given in the sort logic to indicate an error condition. Zero indicates no error occurred in processing the document.

NOTE: This field is set from word three bits 24-29 of the document status.

DES6000 HEADER FIELD DESCRIPTIONS (continued...)

K. NO CHARACTERS IN DOCUMENT. The number of characters in the document 37-108 in the packed mode. If the document is formatted (37-108) the number of characters read in from the handler.

NOTE: This field is set from the document status words - word bits 10-17.

L. POCKET NUMBER. The pocket number which the document was selected to in the sort logic.

NOTE: This field is set from the document status words, word 3 bits 30-35.

M. DOCUMENT SEQ. NO. Contains the DES document seq number. Each document will contain a seq number beginning with 10 and incremented by 10 until all documents in the given job and SNUMB have been processed.

N. FLOAT. Float code obtained from the FRD-ABA master table - from 1-63.

NOTE: This field obtained by DEP from document status word 4 bits 0-5.

O. SEND POINT CODE. The send point code from 001-1000. This field is obtained from the FRD-ABA master table.

NOTE: This field obtained by DEP from document status word 4 bits 6-17.

P. FILE-SORTER CHECKPOINT SEQ. NO. The checkpoint sequence number for this sorter on this file. Each time a checkpoint is taken for a sorter writing to the file checkpoint seq. no. will be incremented by one.

Q. FILE-SORTER BLOCK SEQ. NO. This field will be incremented by one each time a sorter delivers a 320 word block of data to the file. Each sorter block number will commence with one for each sorter 1-6.

R. FILE BLOCK POINTER TO NEXT BLOCK. The location of the next block of data on the file for the job and sorter. This must be used as the random address for the next read.

S. DATA. Either formatted or packed.

7.15 DEP TO SLAVE COMMUNICATION (DESCMM) *

The DEP will write a management communication file to mass storage which will provide the user sufficient information to properly read DEP created sort files, concurrent with DEP writing these files. The communication file will be a perm file named DESCMM which allows user slaves read permissions. This file will contain one (1) 320 word block of file control information necessary for user file access to DEP created files.

The 320 DESCMM block will contain the following management blocks and control words.

- Word No. 1 will contain "DES6000". If the DEP is not running, this will contain zeroes. This word should always contain DES6000 for spawned jobs.
- Word No. 2: Date MMDDYR
- Word No. 3: Time 00HRMN
- Word No. 4: Message sequence number. Each time DEP writes a new message, this counter will be incremented by one. The first message will start with 1. The user may test this to determine if a later message has been filed.
- Word No. 7 (0-17): Number of DEP sort files in the stack 0-15. If this word is zero, there are no sort files open with data.
- Words 8-187: Fifteen file management blocks. Each block contains 12 words. The user must access these to determine the number of blocks that can be processed and checkpoint sequence number for each sorter.
- Word No. 188, bits 0-17: Number of slave jobs running or holding sort files, 0-12. If this counter is zero, no slave jobs holding or using files.
- Words 189-212: Twelve slave job management blocks. Each block contains two words. The user must test the first word of each block to find his slave programs identifying name or number which was placed on the "SPAWN" control card. The second word bits 0-14, for files sequence numbers 1-15, identify the slaves file management blocks within the stack area, words 8-187.

  Example - bit 0 is on, the file management block starts at word 8 as it is the first file in the stack area.

DEP TO SLAVE COMMUNICATION (DESCMM) (continued...)

Example - bit 1 is on, the file is at location 8 plus 12 as it is the second file in the stack.

Example - bit 2 is on, the file is at location 8. plus 24 as it is the third file in the stack.

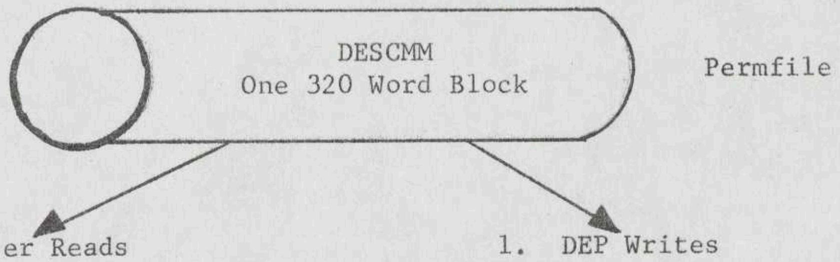
- Word No. 213, bits 0-17: Number of sorters in execution 0-6. If zero, no sorters are running.
- Words 214-273: Six sorter management blocks. Normally the user will not need to reference these blocks. The DEP uses the information in case of restart. Each block contains 10 words.
- Word No. 274, bits 0-17: Number of sort jobs in abort hold status 0-8. If counter equals zero, no sort jobs are in abort hold. Normally the user will not need to reference this data.
- Words 275-306: Contains eight abort hold management blocks. Each block contains four words. Normally the user will not need to reference these blocks.

The user must do the following to properly access the DESCMM file.

- Include a "PERMFIL" card in his object deck set up giving the name DESCMM with read permissions.
- Each time the file is accessed in the slave program, the sequence should be Open, Read, Close. If using GMAP, a 320 word block must be specified, the file read with a MME GEINOS and rewound. These steps are necessary as the DESCMM file is only one block in size and DEP overlaps the previous block on each write.

DEP TO SLAVE COMMUNICATION (DESCMM) (continued...)

Example User Interface



1. User Reads	1. DEP Writes
2. Rewinds	2. Rewinds
3. Reads Again	3. Writes Again

7.16 DESCMM COMMUNICATION AND RESTART MGMT. BLOCK

	0	35
0	Reserved for I/O	
1	Zero or "DES6000"	
2	Date	
3	Time	
4	Message Seq. No.	
5	Zero	
6	Zero	
7	0 No DES Files 17	18 Zero 35
	Fifteen file mgmt. blocks. Each entry = twelve words	
187 188	0 No Slave Jobs Running 17	18 Zero 35
	Twelve slave mgmt. blocks. Each block = 2 words	
212 213	0 No Sorters Running 17	Next available SNUMB
	Six sorter mgmt. blocks. Each block = 10 words	
273 274	0 No Jobs Aborted 17	Zero
	Eight abort mgmt. blocks. Four words each block	
306 307 ↓ 319	Zero	

7.17 SLAVE JOB MANAGEMENT BLOCK

After the user has read the DESCMM file, for the first time he should determine that the DEP is in and running by testing word one (1) of the DESCMM block for the word "DES6000". If DES6000 is contained in word one the DES6000 is running.

The next step is to find the slave program's IDENT (the six digit name placed in the "spawn" control card at sort generation time) in the Slave job management block. When the IDENT is found word 2, bits 0-23, will contain the file locators. See Slave Job Mgmt. Block. After calculating all the file location, test word 4, bits 12-35, of each file mgmt. block to determine if the file contains any data that may be processed. If word four, bits 12-35, of the file mgmt. block is greater than zero, the file may be opened and read.

<table>
<tr><td></td><td>0</td><td></td><td>35</td></tr>
<tr><td>1</td><td colspan="3">SLAVE JOB IDENT
from Spawn Control Card</td></tr>
<tr><td>2</td><td>* File Locators 1-24
bit 0=File 1, bit 1=File 2 etc.</td><td>SPAWN
CTL</td><td>User
Comm
File #</td></tr>
<tr><td></td><td>0 23</td><td>24 29</td><td>30 35</td></tr>
</table>

User placed "END JOB" in word 1 when slave goes to END at job. This will allow respawning on subsequent sort passes where only a single copy at the slave is required.

* Bit 0 = File mgmt. block #1 located (starting) at word 8.

Bit 1 = File mgmt. block #2 located (starting) at word 20.

Bit 2 = File mgmt. block #3 located (starting) at word 32.

Bit 14 = File mgmt. block #15 located (starting) at word 176.

Bits 15-23 not used at present.

7.18 FILE MGMT. BLOCK

The file mgmt. blocks, one for each open file up to fifteen, controls writing and reading of the files. This, along with document header information contained in the files themselves, permit the user programs to read the files and at the same time, properly control file processing. After determining which files belong to the Slave program, as outlined under the "slave job mgmt. blocks" the user should test the file control block, for each file to be read, as follows:

1. Test word 4, bits 12-35 (total no. of blocks written on the file) for non zero. If this field is zero, do not open the file, as no data as yet has been written by DEP.

2. If the file had no blocks written reread DESCMM and repeat step 1., or test another file that may be read by the slave.

3. If word 4, bits 12-35 are non zero the file may be opened.

4. Open the file either as a random file or a sequential file. This should be known and the slave program designed to handle the desired files in either random or sequential fashion. However word 4, bits 0-5, will contain a 1 for sequentially written files, or a 1 for randomly written files. If a random file, a job index will be contained in the first block read.

5. After reading the first data block processing may proceed on a sorter basis or a file basis.

6. On a sorter basis the user should:

   a) compare the header checkpoint sequence number for the given sorter against the corresponding sorter checkpoint sequence number in the file mgmt. block. If the header sequence number is less than the file mgmt. block sequence number, there is another batch of documents following and the batch identified by the file sequence number may be safely processed.

   b) If the header sequence number is equal to the file mgmt. sequence number, DESCMM should be read until a later checkpoint is determined.

   *c) As long as the checkpoint sequence number is satisfactory and the number of blocks written per sorter are greater than the slave has read processing may proceed.

*Normally this type of file will be random and the user must find the next block by getting the points or chain address from characters 31-33 of the doc. header. This will be in the first and last record in the block.

7. On a file basis (non random) processing, reading, of the file may proceed on the basis of reading the file--as long as the DEP block sequence number count of blocks written is greater than the block sequence number found in the block read.

8. When the slave determines that a later message is needed it must read DESCMM. If the control block read is a later message, the message sequence number will have been incremented by DEP (word 4).

9. The DEP will inform the slave when it needs file space by putting the word "REL" in word 12, bits 0-17. The user slave should test this field each time a new message is read from DESCMM.

10. If the slave can release space, the number of 320 word blocks that can be released should be placed in word 12, bits 18-35. The entire message block from DESCMM should then be written to the Slave to DEP communication file.

11. If the Slave does not intend to release any space, at any time, or until end of job,the message "NOL" (no release) should be set to zero and the entire message block from DESCMM written to the slave to DEP communication file.

FILE MGMT. BLOCK

<table>
<tr><td>1</td><td colspan="5">0 Ⓓ USER FILE NAME or IDES FILE NAME 35</td></tr>
<tr><td>2</td><td colspan="2">User File Code
0 Ⓓ 11</td><td>DES File Code
12 Ⓓ 17</td><td>File Seq No.
18 Ⓑ 23</td><td>Slaves Using File
24 Ⓑ 35</td></tr>
<tr><td>3</td><td colspan="3">Current File Size
0 Ⓑ 17</td><td colspan="2">Max File Size
18 Ⓑ 35</td></tr>
<tr><td>4</td><td>File Type
0 Ⓑ 5</td><td>No. Sorters Using File
6 11</td><td colspan="3">Total # Blocks Written to File (Block Seq. No.)
12 Ⓑ 35</td></tr>
<tr><td>5</td><td colspan="3">* No. of Blocks still Available for Writing
0 Ⓑ 17</td><td colspan="2">Next Available Address
18 Ⓑ 35</td></tr>
<tr><td>6</td><td colspan="3">Checkpoint Seq. No. for Sorter #1 Ⓑ 17</td><td colspan="2">18 Ⓑ Last Block Seq. No. Written for Sorter #1</td></tr>
<tr><td>7</td><td colspan="3">Checkpoint Seq. No. for Sorter #2</td><td colspan="2">Last Block Seq. No. Written for Sorter #2</td></tr>
<tr><td>8</td><td colspan="3">Checkpoint Seq. No. for Sorter #1</td><td colspan="2">Last Block Seq. No. Written for Sorter #3</td></tr>
<tr><td>9</td><td colspan="3">Checkpoint Seq. No. for Sorter #4</td><td colspan="2">Last Block Seq. No. Written for Sorter #4</td></tr>
<tr><td>10</td><td colspan="3">Checkpoint Seq. No. for Sorter #5</td><td colspan="2">Last Block Seq. No. Written for Sorter #5</td></tr>
<tr><td>11</td><td colspan="3">Checkpoint Seq. No. for Sorter #6</td><td colspan="2">Last Block Seq. No. Written for Sorter #6</td></tr>
<tr><td>12</td><td colspan="3">Job No. if M file</td><td colspan="2">User to DEP *Communication</td></tr>
</table>

B = binary field

D = BCD field

File type (word 4 bits 0-5) 1 = sequential
2 = random

* includes any released spare by users of file.

7.19 ABORT MGMT. BLOCK

1	SORT JOB # 0 17	JOB SNUMB 18 29	File Seq. No. 30 35
2	BLOCK # CONTAINING LAST CHECKPOINT 0 17	BLOCK # AT 1st BLOCK THIS JOB	
3		NEXT AVAILABLE ADDRESS	
4	SEQ. NO. OF DOC. CONTAINING LAST CHECKPOINT 0 35		

Cont. on Page 62

7.20 SORTER MGMT. BLOCK

0 35

1	Sort Job # 0 17	JOB SNUMB 18 29	File Seq # 30 35
2	Block # Continuing Last Checkpoint 0 17	Block No. of 1st Block This Job 18 35	
3	Sorter Status 0 5		Next Available Address and Chain Address
4	SEQ. NO. OF DOC. CONTAINING LAST CHECKPOINT		
5	CURRENT SEQUENCE NO.		
6	TOTAL # DOCS. PROCESSED THIS JOB BY SNUMG		
7	TOTAL NO. DOCS. PROCESSED ALL JOBS		
8	NO LATE POCKET SELECTS		
9	NO LATE READS		
10	NO FEEDER PKT ALERTS		

Cont. on Page 63

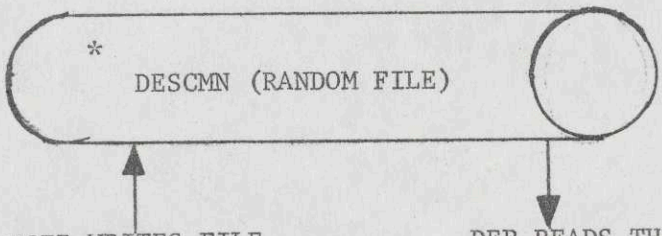
7.21 SLAVE TO DEP COMMUNICATION (DESCMN)

The user slave may communicate to the DEP program by writing to the DESCMN file. The DESCMN file will contain one block (320 words) for each user slave in the slave management area. Each slave will have an assigned block number to write to on the DESCMN file.

The DESCMN will be a random file containing 16 blocks (320 words). The DEP will ensure that each user slave is assigned a unique block address 1-16.

The user must include a $PRMFL card giving the name DESCMN with read and write permissions. This will allow the user to write the necessary communication information to the DEP. Also the user may use the file as a work file for the slave program.

USER TO DEP COMMUNICATION

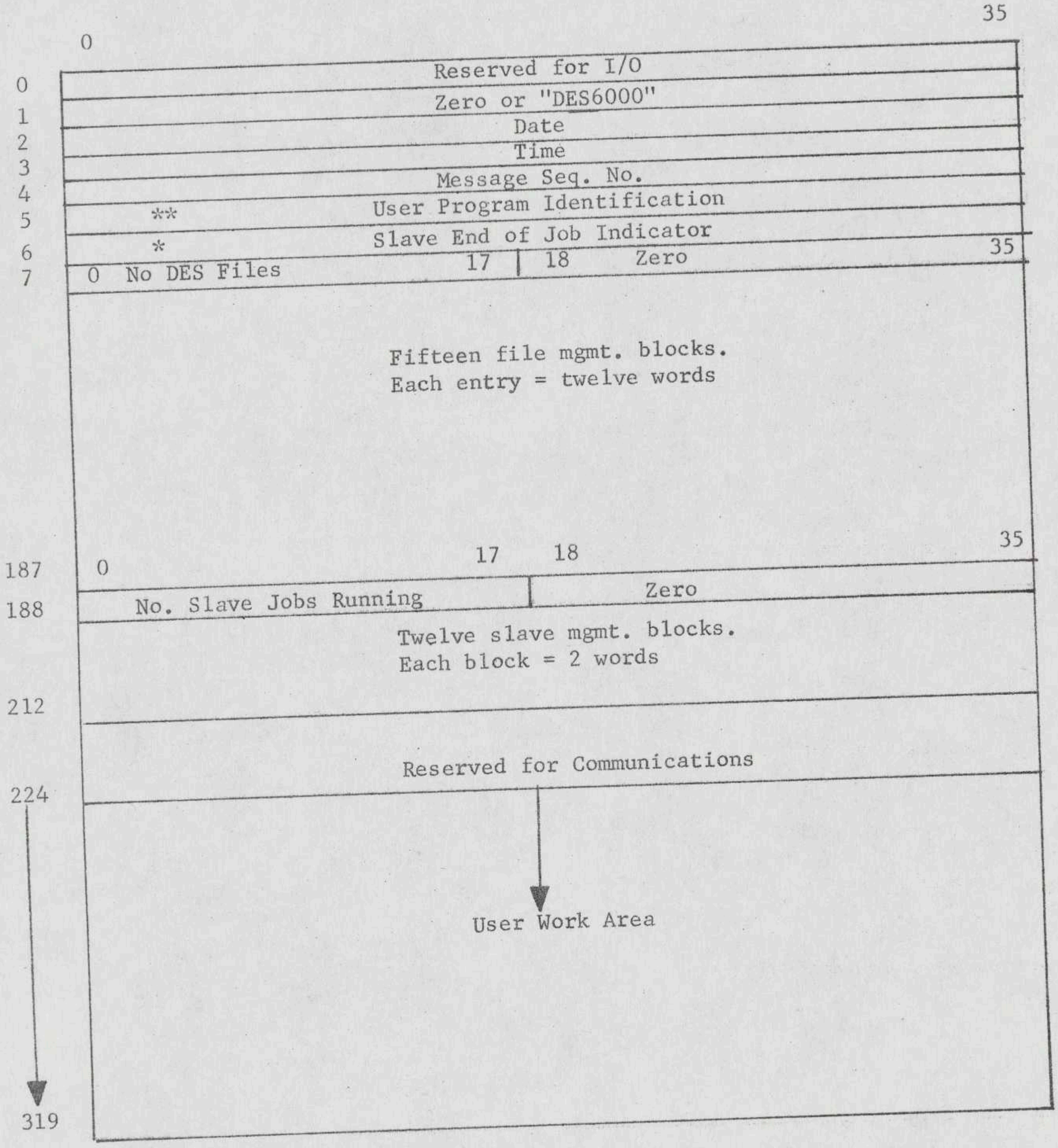


USER WRITES FILE
AND JOB STATUS
INFORMATION TO DEP
TO DESCMN USING
BLOCK NO. ASSIGNED
BY DEP. BLOCK NO
IS GIVEN IN THE
DESCMM FILE UNDER
SLAVE JOB MGMT.
BLOCK, WORD 2, BITS
30-35.

DEP READS THE
FILE TO GET
USER SLAVE FILE
AND JOB INFORMATION.

* User may also read this file.

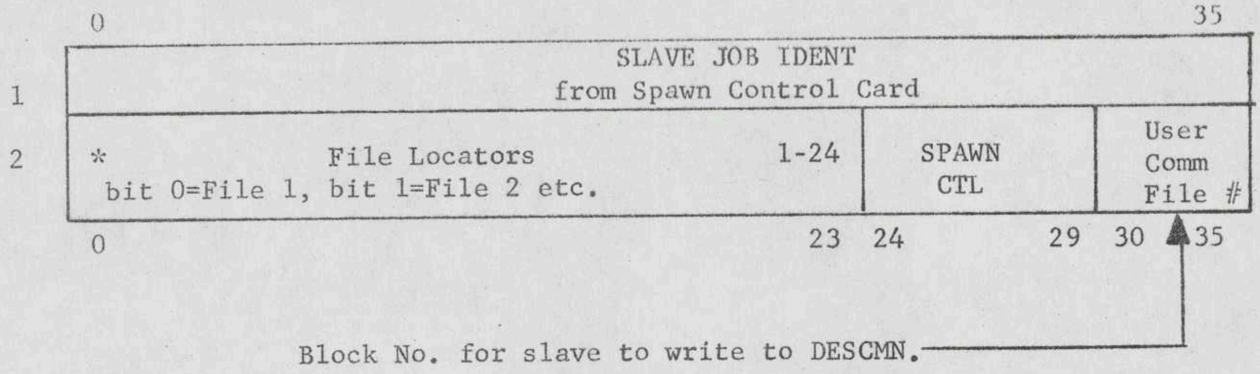
7.22 DESCMN COMMUNICATION - SLAVE TO DEP



* 1 = Slave end of job
2 = Slave abort

** User must place his program ident in this word. Ident must be the same as given in the PSTG SPAWN control card.

7.23 SLAVE JOB MGMT. BLOCK



* Bit 0 = File mgmt. block #1 located (starting) at word 8.

Bit 1 = File mgmt. block #2 located (starting) at word 20.

Bit 2 = File mgmt. block #3 located (starting) at word 32.

Bit 14 = File mgmt. block #15 located (starting) at word 176.

Bits 15-23 not used at present.

7.24 FILE MGMT. BLOCK

<table>
<tr><td>1</td><td colspan="5">0 Ⓓ USER FILE NAME or IDES FILE NAME 35</td></tr>
<tr><td>2</td><td colspan="2">User File Code
0 Ⓓ 11</td><td>DES File Code
12 Ⓓ 17</td><td>File Seq No.
18 Ⓑ 23</td><td>Slaves Using File
24 Ⓑ 35</td></tr>
<tr><td>3</td><td colspan="3">Current File Size
0 Ⓑ 17</td><td colspan="2">Max File Size
18 Ⓑ 35</td></tr>
<tr><td>4</td><td>File Type
0 Ⓑ 5</td><td>No. Sorters Using File
6 11</td><td colspan="3">Total # Blocks Written to File (Block Seq. No.)
12 Ⓑ 35</td></tr>
<tr><td>5</td><td colspan="3">* No. of Blocks still Available for Writing
0 Ⓑ 17</td><td colspan="2">Next Available Address
18 Ⓑ 35</td></tr>
<tr><td>6</td><td colspan="3">Checkpoint Seq. No. for Sorter #1 Ⓑ 17</td><td colspan="2">18 Ⓑ Last Block Seq. No. Written for Sorter #1</td></tr>
<tr><td>7</td><td colspan="3">Checkpoint Seq. No. for Sorter #2</td><td colspan="2">Last Block Seq. No. Written for Sorter #2</td></tr>
<tr><td>8</td><td colspan="3">Checkpoint Seq. No. for Sorter #3</td><td colspan="2">Last Block Seq. No. Written for Sorter #3</td></tr>
<tr><td>9</td><td colspan="3">Checkpoint Seq. No. for Sorter #4</td><td colspan="2">Last Block Seq. No. Written for Sorter #4</td></tr>
<tr><td>10</td><td colspan="3">Checkpoint Seq. No. for Sorter #5</td><td colspan="2">Last Block Seq. No. Written for Sorter #5</td></tr>
<tr><td>11</td><td colspan="3">Checkpoint Seq. No. for Sorter #6</td><td colspan="2">Last Block Seq. No. Written for Sorter #6</td></tr>
<tr><td>12</td><td colspan="3">**</td><td colspan="2">User to DEP *Communication</td></tr>
</table>

B = binary field

D = BCD field

File type (word 4 bits 0-5) 1 = sequential
2 = random

* includes any released spare by users of file.

** contains "RES" or "REF"

RES = release the space given in 18-35 for rewriting.

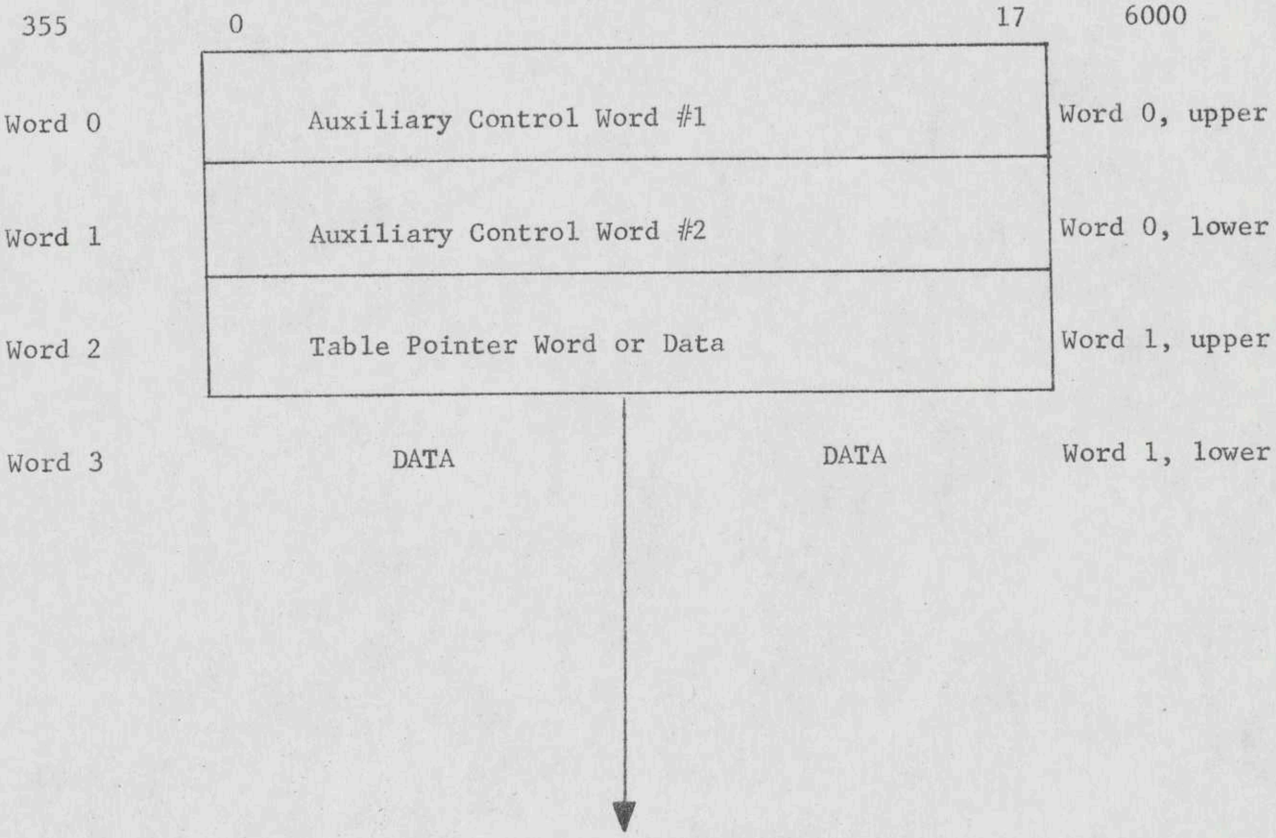
REF = release the file space back to the system. File space given in word 3, bits 0-17 will be released. i.e., file removed from the system.

7.25 INITIALIZATION OF JOBS IN THE 355

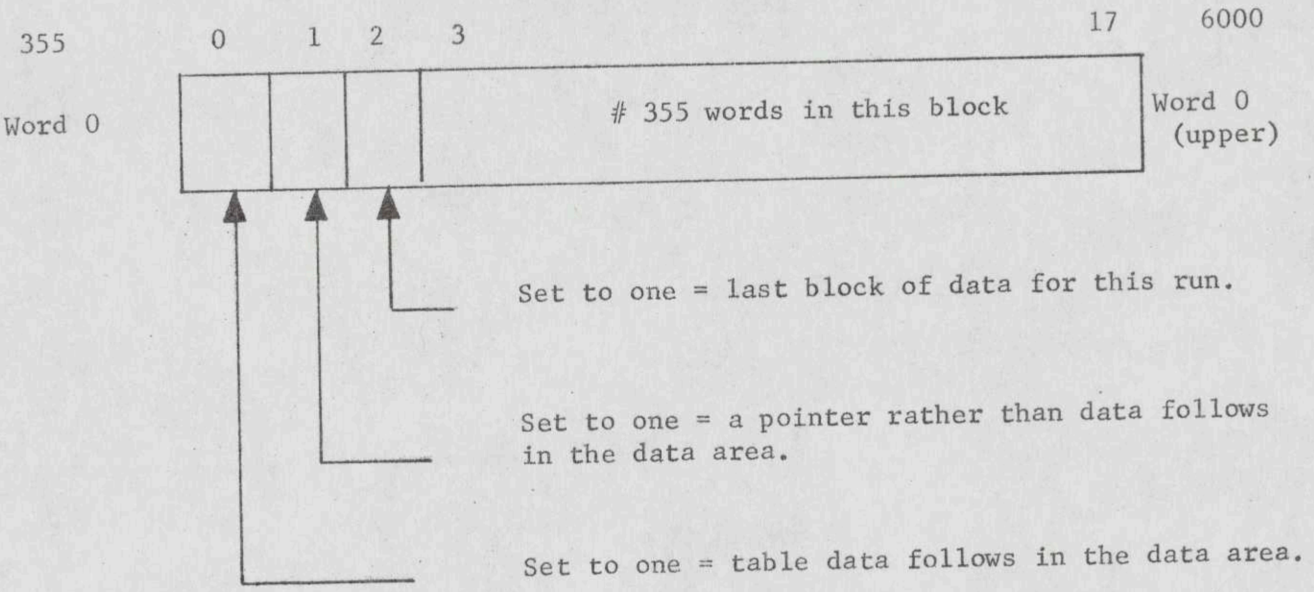
The interface between the 6000 DEP and the 355 DHCP for setting up a job on a handler will be handled through two auxiliary control words, ACW#1 and ACW#2. These control words will inform the 355 DHCP the type of tables and pointers involved and where to place the data within the system control area and/or 355 memory relative to zero. The 355 DHCP will always update the pointers by adding the base address of the system control area.

The messages to the 355 DHCP will be standard Gerts format; however, the 6000 DEP will ensure the first two words of the text as control words, ACW#1 and ACW#2 so that 355 DHCP can interpret the data.

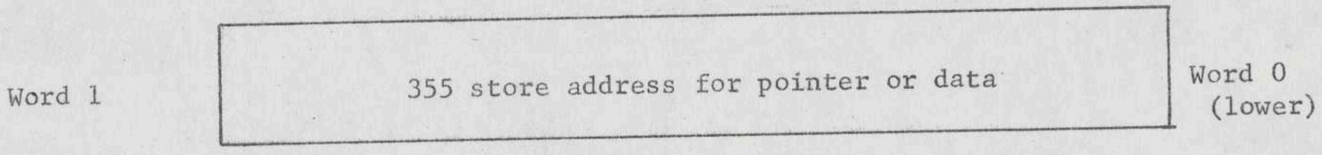
INITIALIZATION TABLE FORMAT



AUXILIARY CONTROL WORD # 1



AUXILIARY CONTROL WORD # 2



7.26 DES MESSAGES

The 6000 Document Entry Program will accept and issue messages from and to the TTY's necessary for operator control, error discovery and recovery, test and diagnostics aids and system and job status. This will require the 6000 DEP to maintain a sizeable message table which will be used to verify, interrogate and respond to incoming messages from the TTY's.

Other messages that will be issued to the TTY's automatically from the 6000 DEP will also be required. These messages will communicate error thresholds, restart conditions, etc.to the operators.

The messages which will be required, their meaning, their action or reaction on the system, are listed by category below.

Operator control messages will provide the means by which an operator via a TTY can initiate a job and control it through "end of Job". This requires the ability to correct error conditions which may occur within the job, restart the job, stop the job, kill the job, etc.

The sign on (initial request from the console teletype or typewriter for the DES6000, procedure will require that the operator request control, dial on, break, etc.depending on the type of TTY or typewriter configured). The 355 GERTS will respond with a request for a program name "Program name?". The operator will then request the "DES".

The "DES" message will be sent to 6000 GECOS via the 355 and the connection established for the terminal to the 6000 DEP direct access program. The 6000 DEP will respond to the requesting terminal with the message "DES6000 RDY". At this point the DES6000 system is ready to accept the various command for setting up the document entry jobs.

a. Program - Name - This message will be issued by the 355 DHCP (GERTS) when initial request is received from the console on TTY terminals.

b. DES - this message will be entered on the TTY to request the 6000 DEP in the 6000.

c. DES6000 RDY - this message will be issued by the 6000 DEP in response to the DES message. It will verify to the user (operator) that the Document Entry System (6000 and 355) is ready to run.

1. ASSIGN yyzz

yy represents the TTY terminal and zz the document handler number. This message permits the operator to assign the TTY (yy) to act as the message control console for one or two or more document handlers. When entered it will cause the 6000 DEP to issue all messages for the jobs running on the handler zz to the assigned TTY (yy).

1a. RDY - zz

This message is issued by the 6000 DEP in response to "assign yyzz". The operator is informed to put the document handler assigned by zz in ready. This will cause the 355 DHCP to respond to the special interrupt caused by putting the handler in ready and connect the document handler terminal to the 6000 DEP. The 6000 DEP will verify that the terminal identification (ID) for the document handler is correct. If correct, message 1B will be issued. If incorrect, message 1F will be issued.

1b. "zz ON yy"

6000 DEP responds to the "assign yyzz with this message if the assignment can be made. This confirms that the document handler is assigned to the TTY designated by yy.

1c. **INVALID** zz ON yy

In the event the document handler zz was previously assigned to another TTY, the 6000 DEP issued this message to indicate the current assignment cannot be made.

1d. REASSIGN yyzz

This message permits the operator to change TTY's for a previously assigned handler.

1e. zz BUSY

This message will be issued when an attempt is made to reassign TTYS to a handler and the handler is in use i.e. running a job. Whenever the handler is not busy and the reassignment can be made zz on yy will be issued. The message "zz busy" may be issued for a variety of other reasons as well.

1f. CK-CONF-RDY-zz

The identification (ID) of the document handler readied does not match the assigned ID (zz) from the TTY. The operator should check the configuration deck and either ready the correct zz or change the assignment to correspond to the configuration.

2. TEST xxxx

This message is entered on the TTY when the operator wishes to run a test job on a particular document handler prior to production or for test diagnostics purposes. The purpose of the test job is to determine if, a handler is operating in a satisfactory manner.

Test xxxx causes the 6000 DEP to verify the test job no (where xxxx is the test no.). Then extract any needed sort control data from the table file for running the test job and transfer it to the 355 DHCP. When the test is ready to run the 6000 DEP will issue a message, "Test xxxx rdy ".

The 6000 DEP program will monitor all data (documents) coming from the document handler on which the test is being run. At end of test a message, E.O.T. xxxx, will be issued to the TTY servicing the document handler. Then totals for the test will be printed which will include the number of documents read followed by error totals by category.

No data will be written to tape or mass storage on test jobs.

2a. TEST xxxx RDY

The 6000 DEP issued this message when the requested test job, denoted by xxxx, is valid and the sort data and sort tables have been transferred to the 355 DHCP, i.e. the job is ready to run.

2b. xxxx INVALID

This message will be issued by the 6000 DEP when a test job, denoted by xxxx, is invalid.

2c. EOT xxxx

The 6000 DEP will issue this message when the test job has finished. It will be followed by error totals as shown under 2d.

2d. TOT. NO. DOC. 9999

Test xxxx errors - If there is no detected errors the Test xxxx errors will be followed by the number 0 (zero). If there are errors, the errors will be listed as follows:

MRS 200 or DRD 200

Test xxxx errors -

Hardware - feed failures -
data alerts -
T/Timing -
S/echo -

Software -

2e. DH-NO AND ZZ - XXX - see 3d and 3e

3. NEW xxxx

The operator will enter this message through the TTY to request a new job (xxxx represents the job no.). The job will be run on the document handler previously assigned to the TTY by the "assign" message. When there are more than one document handlers assigned, to the TTY the 6000 DEP will automatically issue a message requesting the operator to assign a document handler for the job. (D.H. - no.?) if the job number requested is valid, the 6000 DEP will extract the necessary control and sort data from the table file and transfer it to the 355 DHCP. When the 355 DHCP returns a ready to run status for the job, the operator "Job xxxx rdy". In the event the requested job number is invalid the 6000 DEP will issue a message "xxxx invalid" and wait for the operator to enter a new job request: "new xxxx".

3a. ZZ xxxx RDY

After the 6000 DEP has 22 - xxxx verified that the handler and job number, denoted by, is valid and all sort control data sent to the 355 DACP with a return status that the 355 and the disc handler are ready to run the requested job; this message will be issued to the operator.

3b. xxxx INVALID

This message is issued when the requested job no., denoted by xxxx is invalid. The operator must repeat message 3. "Now xxxx" giving corrected job number.

3c. EOJ -zz- xxxx

The message is issued at end of job.

3d. DH - NO?

This message is issued when a job is requested from a TTY to which more than one document handler has been assigned.

The operator must respond by typing the document handler identification (ID) and the job number zz-xxxx, where zz is the document handler and xxxx the job number.

3e. zz - xxxx

The operator types the message in response to SB.

zz - document handler ID

xxxx - job number

3f. EOR zz-xxxx

The documents have all been read but no E.O.J. control document. Job will not be closed until a positive close is given either from the console or a resuming procession with E.O.J. document.

4. START zz

This message is entered by the operator when he is ready to start a run. It must have been preceded by an "assign yyzz" and a "new xxxx" or "test xxxx". When this message is received by the 6000 DEP and is verified to be valid, i.e. the job number is correct and the 355 DHCP is ready to run the job, the 6000 DEP issues a command to the 355 DHCP to commence feeding and reading to the document handler on which the job is loaded.

4a. zz NOT RDY

This message is issued to the operator when the 355 DHCP determines that the handler to which the job is assigned is not in ready. The 355 DHCP returns the not ready status to the 6000 DEP which in turn issues the message. The operator must ready the handler after which message 3a will be issued. ZZ represents the handler number. This should not occur unless the handler drops ready or the operator taxes the handler out of ready inadvertently.

5. STOP zz

When the operator wishes to temporarily, stop a run on a document handler, he will issue the message zz represents the handler ID. This will cause a controlled stop of the handler and all documents following the stop of the handler will be read and pocketed by the 355 DHCP. Normal operation will be resumed by issuing a start zz message.

6. CLOSE zz - xxxx

The operator can bring a job to a normal end of job or test by the use of the close message. Normally this message will be used when all documents have been processed for a job but there was no final control document to cause a normal end of job. ZZ represents the handler ID on which the job was running.

7. KILL zz-xxxx

This message is used when it is necessary to terminate a run along with any files created by it. Data files created by the job will not be saved. When the kill option is used the job will have to be rerun.

8. ABORT zz

This message will be used when it is desired to terminate a run but save the data files created by the job. The job can be restarted at a later time by using the restart xxxx. Processing will resume from the last restart point retained in memory by the 6000 DES. The 6000 DES will not stop the job until an appropriate restart point has been established. This prevents the need to reposition the data file at restart. When the abort zz is issued to the 6000 DES the job number running on zz is placed in a suspense table. When restart xxxx is issued the suspense file is searched for a matching job number and if found the 355 DHCP is reinitiated for the given job and processing resumed from the retained check point.

9. RESTART xxxx-zz

This message is used to restart a job that was previously terminated by an abort zz or through a hardware failure. xxxx is the job to be restarted and zz is the handler on which the job was aborted.

zz-xxxx ABORTED

This message verifies that the requested job has been terminated and the handler on which it was running is available for another job.

NOTE: When a job is aborted and later restarted; it will be possible to restart the job on a different handler provided that: the handler is of the same type and not in use. This is made possible because sort tables and control information are reissued to the 355 DHCP. Also all control totals in the 6000 DEP will be saved after an abort and when the job is restarted carried through to end of job.

10. AUDIT xxxx

Typing the audit xxxx message will cause the 6000 DEP to print on the requesting TTY all the job totals for the job number denoted by xxxx.

10a. WW = Pocket ZZ KL SEP WW

If the condition arises where the number of documents directed to kill pockets prior to encountering a kill separator exceeds the authorized limit, the above message is typed to the operator.

10b. ZZ ATTN

This message reflects a hardware status condition, (full pocket or empty feed hopper), that requires manual intervention.

10c. ZZ JAM

Whenever the transport mechanism is implied to the point of preventing successful reading and pocketing of documents, or an endorser malfunction occurs, the above message is typed to the operator.

10d. ZZ SHUT DOWN

This message is typed to the operator whenever late pocket and late read commands exceed system limitations, or error conditions have occurred which require immediate handler shutdown.

10e. ZZ REQ SEPRS

If the number of Batch Separators are insufficient for proper pocket allocation, the above message is typed to the operator.

10f. ZZ NON-HOMO

Whenever the documents contained within a batch are, according to run types and transaction codes, not applicable to that run, the above message is typed to the operator.

10g.	ZZ $ ERRS	(Dollar amount errors)
	ZZ TR ERRS	(Transaction code errors)
	ZZ ACCT ERRS	(Account Number errors)
	ZZ RTE ERRS	(Transit Number errors)
	ZZ TCD ERRS	(Transposition Check Digit errors)

The system maintains counts of the above errors encountered during processing. Whenever the error count exceeds an allowable limit the system so notifies the operator through one of the above messages.

10h. ZZ CHANALERT
ZZ DATALERT
ZZ CMND REJ
ZZ CHANBSY

The above messages reflect hardware status conditions and will follow the ZZ SHUT DOWN message.

11. ERRORS zz

This message will be provided to assist T&D. Entering this message on the TTY will cause the 6000 DEP to print all hardware and/or possible hardware errors by category on the requesting TTY. Errors will be kept by job and by total jobs for each handler.

12. REVIEW xxxx

This message will be provided to enable an operator to get a printout of the jobs table sort and control data. Entering this message on the TTY will cause the 6000 DEP to printout the control information on the requesting TTY for the job number denoted by xxxx. Review xxxx may be issued prior to a start xxxx to ensure that the table sort parameters are correct.

13. SWITCH zz TO mm

This message allows an operator to switch a run from one handler to another. ZZ represents the from handler and mm represents the to handler.

7.27 SPAWN CONSOLE VERB (FOR DES6000)

The new console verb SPAWN initiates execution of a local batch, remote batch, or time-sharing initiated batch program that has been placed on a utility permanent file. This feature allows placement of frequently run utility programs on a permanent file and initiation of execution of these jobs via console input.

- Building the SPAWN Job File -- Before a job can be placed in the job file, a System Master Catalog (SMC) entry must be created using the FILSYS activity for the user-id OPNSUTIL with a password of UTILIB. A sample directive is shown in the deck setup below:

```
1      8       16
$      SNUMB   sssss
$      IDENT   optional
$      FILSYS  options
$      PRIVITY
CRMAST OPNSUTIL/OPNSUTIL,PASSWORD/UTILIB/SIZE/n/,
          RESOURCES/nnnnnn/
$      ENDJOB
***EOF
```

Next, file space must be created for the job. This can be done immediately following the creation of the SMC entry by including additional directives to the file system. Files can also be created separately as shown below:

```
1      8       16
.....
....
$      FILSYS  options
USERID OPNSUTIL$UTILIB
 FCREAT OPNSUTIL/DESXX,READ,BLOCKS/30,30/      30 Blocks File Size
$      ENDJOB                     (M).(N)
***EOF
```

DES6000 SPAWN
FILE NAME

Notes:

1. The file name used must consist of exactly five characters, because the file name becomes the SNUMB of the spawned job.

2. The values indicated by n,m depend upon the size of the job being placed on the file. An approximation may be obtained by assuming that one block. (1link) is required for every 11 cards. This value is based on binary cards, each including 27 words plus the GEFRC record control word. For BCD cards, less space will be required.

- Placing the SPAWN Job on the File -- The job to be spawned is placed on the permanent file using Bulk Media Conversion (BMC). The rules for the deck setup are the same as for any other job entered into GECOS III except that the $ SNUMB and $ IDENT card must be removed from the job to be placed on the permanent file.

A sample BMC deck setup for placing a job on the permanent file is shown below (the job is the DES6000 program and its necessary control cards):

```
1      8       16
$      SNUMB   sssss
$      IDENT   optional
$      USERID  OPNSUTIL$UTILIB
$      CONVERT options
$      DATA    IN,,COPY
$      LOWLOAD
$      OBJECT
$      DKEND   <------------ DES BINARY DECK            Deck being
$      EXECUTE                                          placed on
$      PRIVITY                                          permanent
$      LIMIT   10,24K,,99999                            file
$      PRMFIL  PUT PERM FILE CARDS HERE
$      ENDJOB
***EOF
$      ENDCOPY
$      PRMFL   OT,W,S,OPNSUTIL/DESXX
$      ENDJOB
***EOF
```

- Using the SPAWN Verb -- The format of a SPAWN request is as follows:

  SPAWN DESXX

A space must be entered between the verb and filename, and filename must consist of exactly five characters.

If the request is accepted by the system, the job is placed in queue to be run.

7.28 GENEWS--SPAWN NEW JOB

The MME GENEWS allows a user program to spawn up to eight independent programs for execution by the system. The program(s) may be described on a temporary file, a permanent file, or in core within the originating program's allocated memory.

Entry

Entry to this routine is made from the fault interrupt vector as a result of the MME GENEWS.

Calling Sequence

```
L        MME      GENEWS
L+1      ZERO     A,B
L+3      ABORT    Return
L+3      Normal   Return
```

Where:

A = pointer to the first word of a 320 word buffer for use by the MME GENEWS processor.

B = pointer to a location containing either:

1. File Code in bits 24 through 35. If job to be spawned resides on a temporary file. Bits 0 through 17 must be zero.
2. Starting location of an input stream in bits 0 through 17 if the job to be spawned resides in core or on a permanent file.

Spawning a Job Residing in Core and/or on Permanent File

The job skeleton is described within core as a BCI string. For jobs residing on permanent files, the BCI input stream normally terminates with a $ SELECT. For job skeletons residing totally within core, the stream terminates with the $ ENDJOB.

To spawn the job on permanent file AB, with catalog name JDOE and password OPEN, the coding could be as follows:

```
         MME       GENEWS
         ZERO      A,B
         TRA       ERROR
          .
          .
      C  BCI       3,$b̸b̸b̸b̸b̸b̸IDENTb̸b̸b̸XXX
         BCI       2,XX,JOHNb̸DOE
         BCI       9,
         BCI       3,$b̸b̸b̸b̸b̸b̸USERIDb̸b̸JDO
         BCI       9,E$OPEN
         BCI       2,
         BCI       3,$b̸b̸b̸b̸b̸b̸SELECTb̸b̸JDO
         BCI       5,E/AB
         BCI       6,
         ZERO      -1,0
      A  BSS       320
      B  ZERO      C,0
```

If the request is not accepted, it will be due to one of the following error conditions:

1. The filename entered at the console was not found in the OPNSUTIL catalog and the job was deleted. The following SPAWN error message appears at the console:

   ```
   GEIN REMOTE, SNUMB sssss JOB DELETED
   ```

   Where: sssss = Filename entered

2. The .MGNEW module is unable to spawn the job, and the job is aborted. The following SPAWN error message appears at the console:

   ```
   *SPAWN ABT, CODE nn
   ```

   Where: nn is one of the following .MGNEW abort codes:

   06 - Illegal seek address (system error)
   07 - No links available (try later)
   11 - SNUMB table full (try later)
   12 - GEPOP queue full (try later)
   13 - Requested SNUMB (filename) already in use (try later)
   14 - Internal spawn error (system error)

- SPAWN Implementation -- The console request to spawn a job is processed by the .MPOPM module. Actual spawning is done by the .MGNEW module.

The SNUMB of the spawned job will be the filename specified in the console message. The $ IDENT and $ SNUMB cards are manufactured for the job from constants coded into the .MPOPM module. The $ IDENT card currently contains the characters "ZZZ,OPNSUTIL" starting in column 16. The variable field of the $ IDENT card may be changed to suit an individual installation's needs by patching or reassembling the .MPOPM module.

Restrictions

1. On temporary file jobs, the $ DUMMY GNEW must be the first card image.

2. On permanent file/in core jobs, the first BCI string must be the $ IDENT card images.

3. All BCI card images must be 14 words long. The GMAP restriction of 9 words for the BCI pseudo-operation must be observed.

4. The end of the BCI input stream is denoted by: ZERO -1,0.

5. If the file code is zero or if either character of the file code is an "*" or a "$" the request will be aborted.

6. The $ SNUMB and activity number of the originating job will appear on the $ SNUMB control card of the spawned job starting in column 36.

7. The $ DUMMY card image for temporary file jobs will be replaced by a valid $ SNUMB card image. In jobs utilizing the BCI input stream, the $ IDENT image will be prefaced by a valid $ SNUMB image.

8. There is no restriction on the types of jobs to be spawned. Any job which run normally may be spawned.

9. A job spawned from a remote station will be returned to that station.

10. In an abort return, the first word of the buffer will contain an abort code (right justified) indicating the reason as follows:

CODE	REASON
1	Eight jobs already spawned by this program.
2	File/stream pointer outside calling program.
3	Illegal file code for temporary file.
4	File code not in PAT.
5	Temporary file device not mass storage.
6	Illegal seek address on device.
7	No links available on device.
10	First temporary file block not $ DUMMY card image.
11	SNUMB table full.
12	GEPOP queue full.
13	Requested SNUMB is already in CRSNB table.

- See GECOS MANUAL CPB1518B for complete information concerning the MME GENEWS SPAWN NEW JOB feature.

7.29 INTERFACE COMMANDS

The commands (MMC GEROUTS) required to allow the 6000 DEP to communicate to the 355 DHCP are as follows:

1. Start Handler (op code 21)

<u>Example</u>:

VFD Instruction Record Pointer Op Code Line ID

Word 0	MME GEROUT
Word 1	VFD 18/Record Pointer, 06/21,H12/00
Word 2	Zero status, courtesy call address pointer

Word 1	Record pointer to word 0 of the data block.
	Operation code (Octal 21-start handler)
	Line ID (2 character line identification

Word 2	Pointer to word at which to store status. Pointer to location at which courtesy call routine starts.

Data Block

Word 0	Zero 2, Input Buffer Address
Word 1	Zero 4,0
Word 2	Zero 1,0

Word 0	The 2 represents the number of words to be transmitted to the 355 DHCP. Input Buffer is the starting location in the 6000 DEP to which the 355 DHCP should send its data.
Word 1	Number of characters to be transmitted to the 355 DHCP. Right half of Word 1 not used.
Word 2	The 1 represents a special code to the 355 DHCP to start the handler.

2. Continue Reading (Op Code 21)

Word 0	MME GEROUT
Word 1	VFD 18/Record Pointer, 06/22, H12/00
Word 2	Zero Status, Courtesy call address pointer

Word 1	Pointer to input buffer in the 6000 DEP to which DHCP should send its data.
Word 1	Operation Code (Octal 22 - continue handler reading)
	Line ID (2 character line identified)
Word 2	Pointer to word at which to store status.
	Pointer to location at which courtesy call routine starts.

3. Stop Handler (Op Code 30)

Word 0	MME GEROUT
Word 1	VFD 18/0, 06/30,H12/00
Word 2	Status 0

Word 1	First 18 bits = 0
	Operation Code (Octal 30 - stop handler)

	Line ID (2 character line identified)
Word 2	Pointer to word at which to store status.
	Pointer to location at which courtesy call routine starts.

3. Stop Handler (Op Code 30)

Word 0	MME GEROUT
Word 1	VFD 18/0, 06/30,H12/00
Word 2	Status 0

Word 1	First 18 bits = 0
	Operation Code (Octal 30 - stop handler)
	Line ID (2 character line identifier)
Word 2	Pointer to word at which to store status in left half or word. Zero in the right half of word.

It should be noted line ID for document handlers will always be assigned an ID of one through six. Teletype line IDs will be handled as in the normal GERTS package, except that the ID will be assigned a line ID of seven or greater. For further information refer to the 625/6000 GERTS Programming reference manual, CPB-1558.

8.0 THE 6000 POCKET SELECT TABLE GENERATOR (6000 PSTG)

1. The 6000 PSTG will be a batch program with the capability to construct various tables which contain control information and document select (sort) tables for the 355 DHCM. These tables will be a PERM file stored on a mass media device. The table generator program must be able to create and up-date (file maintain) the file including the following:

    a. Create and/or add a new table or tables from data on magnetic tape or punched card.

    b. Delete a previously constructed table or tables from the file.

    c. Make changes to the tables within the file, from data on punched card or magnetic tape by adding, deleting, or replacing items within the table or tables.

2. The 6000 PSTG must be developed in close coordination with the 355 DHCP to ensure that:

    a. The table contains all the necessary information that the 355 DHCP requires.

    b. The format of the tables are always in agreement with the coded logic of the 355 DHCP.

3. The Pocket Select Tables will be constructed from user furnished data either on punched card or magnetic tape.

4. The format and language of the input record for the 6000 PSTG punched card or magnetic tape will be determined and described for the user. It must be adequately explained to ensure ease of implementation. Section shows the input language. There will be three user input forms:

    a. Master Transit Table

    b. General Table

    c. Scan Logic

8.1 PSTG FILE REQUIREMENTS

The DES file system will be PERM file system named "DES", i.e., the User Master Catalog name. The DES will contain six major subcatalogs. Such catalogs will be named as follow:

*1. DESCMM - DEP to slave communications.

**2. DESCMN - Slave to DEP communications.

3. SPNXXX - These files will contain all user slaves that the DEP will spawn. XXX will be numeric 001-999. The PSTG will create and write these files. The DEP will read these files with MME GENEWS.

4. TABXXX - These files will contain the assembled sort programs, with the exception of master tables. Master Tables will be filed on the MSTXXX files. The PSTG will create and write the assembled scan records and general tables on these files. XXX will be the user's sort job number, i.e., TAB001-TAB999. The DEP will read these files whenever a sort job is called for from the TTY's

5. MSTXXX - These files will contain the sort jobs master tables. The PSTG will create and write the assembled master tables on these files. XXX will be the user master table name, i.e., M01-M99. The DEP will read these files when cross referenced from the TABXXX files, and include the master table in the requested sort job.

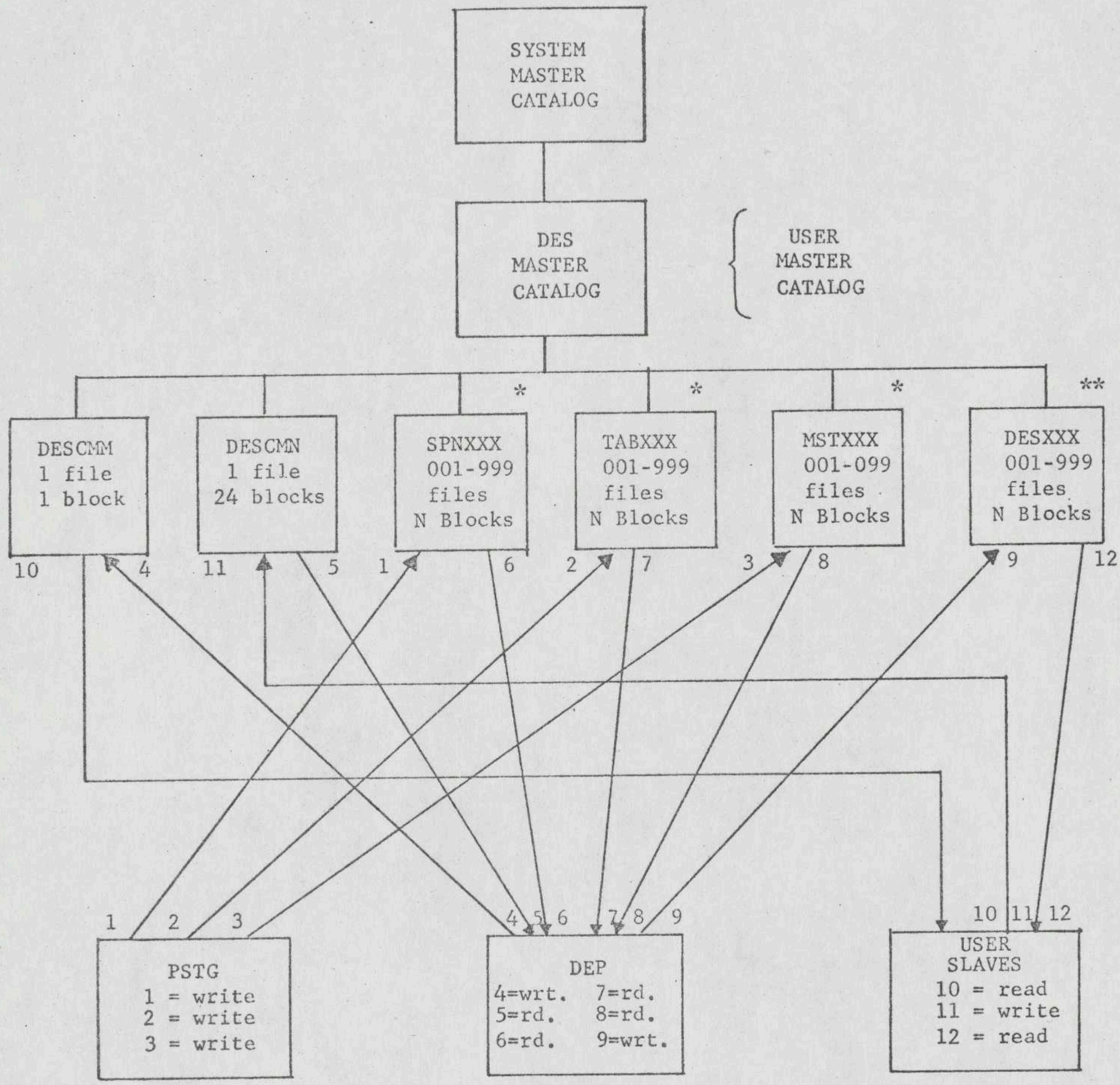
6. DESXXX - The DEP will create and write sorted data to these files. The size and name of these files will be provided on the TABXXX files as part of the user's sort job parameters. The DEP will also release these files when requested by the user.

Other than DESCMM and DESCMN the subfiles will be created and released dynamically through the use of a DES master mode routine. The file building routine must do the following:

a. Build PAT pointers for the files in the SSA.
b. Build PAT bodies.
c. Enter file names and spare requirements in the Catalog System.

*DESCMM will not be created through master mode routine. It will be a normal perm file consisting of one block (320 words).

**DESCMN will not be created through the master mode routine. It will be a normal perm file consisting of 24 blocks (24x320 words).



* created by the PSTG using the master mode file create routine.

** created by the DEP using the master mode file create routine.

Note: XXX = file name with any subname.

Cont. on Page 88

8.2 TABXXX - JOB FILE LIBRARY HEADER AND INDEX

	0 17	18 35
0	JOB NO. (BCD) 001 - 999 or ALFA.	355 Core req. (Bin) for this Job.
1	Device Type MRS 200 - DRD 200 - H236 (BCD)	
2	(Bin) Output Record Size (in 600 words) 1 - 18	No. of Files (Bin) (Sort-files) 1 - 6
3	Shared or Private files	USER or DES supplied
4	File #1 File code	Seq. or random 1 = seq. 2 = random
5	Min. File Size in Links	Max. File Size in Links
6	File Name for File #1 (only if user supplied)	
7	File #2 File Code	Seq. or random
8	Min. File Size in Links	Max. File Size in Links
9	File Name for File #2 Only if User Supplied	
10	File #3 File Code	Seq. or Random
11	Min. File Size in Links	Max. File Size in Links
12	File Name for File #3 Only If User Supplied	
13	File #4 File Code	Seq. or Random
14	Min. File Size	Max. File Size
15	File Name for File #4 Only if User Supplied	
16	File #5 File Code	Seq. or Random 1=seq/2=random
17	Min. File Size in Links	Max .File Size in Links
18	File Name for File #5 Only if User Supplied	
19	File #6 File Code	Seq. or Random 1=seq/2=random
20	Min. File Size in Links	Max. File Size in Links
21	File Name for File #6 Only If User Supplied	
22	File Code for DES Comm. File	Seq. or random 1 = seq. 2 = random
23	File Name "DES CMM"	(1 LLINK)
24	File Code for User Comm. File	
25	File Name for User Comm. File - "DESCMN" - (24 LLINK)	
26	Zero	

8.2 TABXXX - JOB FILE LIBRARY HEADER AND INDEX (continued)

	0 17	18 35
27		
28	Master File Name No. 1. MXX M99 or MXX	No. of 18 bit words used
29	Master File Name No. 2. MXX	No. of 18 bits words used
30	Master File Name No. 3. MXX	No. of 18 bit words used
31	Master file Name No. 4 MXX	No. of 18 bit words used
32	Master File Name No. 5 MXX	No. of 18 bit words
33	Master File Names No. 6-MXX	No. of 18 bit words
34	UA	UA
35	UA	
36	UA	
37	UA	
38	UA	
39	UA	
40	Block No. (320 word) Loc. of Start of SCAN recs.	Starting word in block (36 bit)
41	No. of 18 bit words used for SCAN recs.	Starting 355 Loc. of SCAN records
42	Block No. (320 word) Loc. of private tables	Starting word in block (36 bit)
43	Block No. (320 word) loc. of start of private table	Starting word in block (36 bit)
44	No. of 18 bit words used for scan recs.	starting 355 location of tables
45	Block No. (320 word) Loc. of SAVE Area	Starting word in block (36 bit)
46	No. of 18 bit words	Starting 355 location of hold area.
47	UA	
48	UA	
49	UA	

8.2 TABXXX - JOB FILE LIBRARY HEADER AND INDEX

0 17 18 35

50	UA	
51	UA	
52	UA	
53	UA	
54	Block No. (320 word) loc. of start of pocket tables	Starting word in block (36 bit)
55	No. of 18 bit words used for pkt. tables	Starting 355 location
56	Block No. (320 word) where SPAWN Job CTL CDS Loc.	Starting word in the block
57	No. of ctl cds 84 (14 words)	
	S, M or SPAWN N Cond. Code 1, 2 Code	Condition code = 1 = checkpoint 1 immediate, 3 = END JOB, 4 = record count
	IDENT # of SPAWN JOB	
	DES File	
	SPAWN JOB #2	
	"	
	"	
63	"	
64	Assembled program for 355	

8.3 MSTXX FILES (FRD-ABA MASTER TABLE INDEX)

	0	35
Word 0	RESERVED for I/O	
1	0 No. of Jobs Indexed 1-60 17	18 TABLE TYPE 1 or 2 35
2	Length of Table in 36 bit words	Length of Table in 18 bit words format for 355
3	No. of Arg's in Table-FRD-CITY/STATE-ABAS	Big Item Pkt.
4	Table Name "MSTXXX"	
5	RESERVED	
6	* JOB # for 1st Job	HI - Dollar
54	JOB # for 48th Job	HI - Dollar
63	RESERVED	

*Job No.'s correspond to location of PKT. Fields in segments.

EXAMPLE: Job located in word 6 bits 0-17 will have its pocket no.'s placed in 1st PKT word bits 0-5, Job # in word 6 bits 18-35 will have its pkt no.'s in PKT word bits 6--11, etc.

8.4 FRD-ABA TABLE FILE-SEGMENT NO. 1 (FRD Numbers)

<table>
<tr><th></th><th colspan="3">0 17</th><th>18 23</th><th colspan="2">35</th></tr>
<tr><td>0</td><td colspan="3">*NO FRD ENTRIES IN TABLE</td><td>SEG. 1</td><td colspan="2">LENGTH OF ENTRY</td></tr>
<tr><td>1</td><td colspan="3">NO FRD ENTRIES IN SEG.</td><td colspan="3">BASE ADDRESS FOR CAL. IN S.P.</td></tr>
<tr><td>2</td><td colspan="3">FRD ENTRY NO. 1</td><td colspan="3">POINTER TO CITY STATE SEGMENT.</td></tr>
<tr><td>3</td><td colspan="3">TIME CODES JOB 1</td><td colspan="3">TIME CODES JOB 2</td></tr>
<tr><td>4</td><td colspan="3">" " " 3</td><td colspan="3">" " " 4</td></tr>
<tr><td>5</td><td colspan="3">" " " 5</td><td colspan="3">" " " 6</td></tr>
<tr><td>6</td><td colspan="3">FLOAT AND S.P. JOB 1</td><td colspan="3">FLOAT AND S.P. JOB 2</td></tr>
<tr><td>7</td><td colspan="3">" " " " 3</td><td colspan="3">" " " " 4</td></tr>
<tr><td>8</td><td colspan="3">" " " " 5</td><td colspan="3">" " " " 6</td></tr>
<tr><td>9</td><td>Job 1 pkt</td><td>Job 2 pkt</td><td>Job 3 pkt</td><td>Job 4 pkt</td><td>Job 5 pkt</td><td>Job 6 pkt</td></tr>
<tr><td>10</td><td colspan="3">NO. FRD ENTRIES IN SEG.</td><td colspan="3">BASE ADDRESS FOR CAL. S.P.</td></tr>
<tr><td>11</td><td colspan="3">FRD ENTRY NO. 2</td><td colspan="3">POINTER TO CITY/STATE SEG.</td></tr>
<tr><td></td><td colspan="3"></td><td colspan="3"></td></tr>
<tr><td></td><td colspan="3"></td><td colspan="3"></td></tr>
<tr><td></td><td colspan="3"></td><td colspan="3"></td></tr>
</table>

*First Segment only.

8.5 FRD-ABA TABLE FILE SEGMENT NO. 2 (CITY/STATE CODES)

<table>
<tr><td></td><td>0</td><td></td><td>17</td><td>18 23</td><td></td><td>35</td></tr>
<tr><td>0</td><td colspan="3">NO. CITY/STATE CODES IN TABLE</td><td>SEG. #2</td><td colspan="2">LENGTH OF ENTRY</td></tr>
<tr><td>1</td><td colspan="3">NO. CITY/STATE CODES IN SEG.</td><td colspan="3">BASE ADDRESS FOR CAL. S.P.</td></tr>
<tr><td>2</td><td colspan="3">CITY/STATE CODE # 1</td><td colspan="3">POINTER TO ABA NO's.</td></tr>
<tr><td>3</td><td colspan="3">TIME CODES JOB #1</td><td colspan="3">TIME CODES JOB #2</td></tr>
<tr><td>4</td><td colspan="3">TIME CODES JOB #3</td><td colspan="3">TIME CODES JOB #4</td></tr>
<tr><td>5</td><td colspan="3">TIME CODES JOB #5</td><td colspan="3">TIME CODES JOB #6</td></tr>
<tr><td>6</td><td colspan="3">FLOAT S.P. JOB #1</td><td colspan="3">FLOAT S.P. JOB #2</td></tr>
<tr><td>7</td><td colspan="3">" " " #3</td><td colspan="3">" " " #4</td></tr>
<tr><td>8</td><td colspan="3">" " " #5</td><td colspan="3">" " " #6</td></tr>
<tr><td>9</td><td>Pkt # Job 1</td><td>Pkt # Job 2</td><td>Pkt # Job 3</td><td>Pkt # Job 4</td><td>Pkt # Job 5</td><td>Pkt # Job 6</td></tr>
<tr><td>10</td><td colspan="3">NO. CITY/STATE CODES IN SEG.</td><td colspan="3">BASE ADDRESS FOR CAL. S.P.</td></tr>
<tr><td>11</td><td colspan="3">CITY STATE CODE #1</td><td colspan="3">POINTER TO ABA NO's.</td></tr>
<tr><td></td><td colspan="6"></td></tr>
<tr><td></td><td colspan="6"></td></tr>
</table>

NEXT
SEG.

*First Seg. only.

8.6 FRD-ABA TABLE FILE SEGMENT NO. 3 (ABA NO's.)

<table>
<tr><th></th><th colspan="3">0 17</th><th>18 23</th><th colspan="2">35</th></tr>
<tr><td>0</td><td colspan="3">*NO. ABA NO's. IN TABLE</td><td>SEG. 3</td><td colspan="2">LENGTH OF ENTRY</td></tr>
<tr><td>1</td><td colspan="3">NO. ABA NO's. IN SEG.</td><td colspan="3">BASE ADDRESS FOR CAL. S.P.</td></tr>
<tr><td>2</td><td colspan="3">ABA CODE #1</td><td colspan="3">ZERO</td></tr>
<tr><td>3</td><td colspan="3">TIME CODES JOB #1</td><td colspan="3">TIME CODES JOB #2</td></tr>
<tr><td>4</td><td colspan="3">" " " #3</td><td colspan="3">" " " #4</td></tr>
<tr><td>5</td><td colspan="3">" " " #5</td><td colspan="3">" " " #6</td></tr>
<tr><td>6</td><td colspan="3">FLOAT & S.P. JOB #1</td><td colspan="3">FLOAT & S.P. JOB #2</td></tr>
<tr><td>7</td><td colspan="3">" " " #3</td><td colspan="3">" " " #4</td></tr>
<tr><td>8</td><td colspan="3">" " " #5</td><td colspan="3">" " " #6</td></tr>
<tr><td>9</td><td>Pkt #
Job 1</td><td>Pkt #
Job 2</td><td>Pkt #
Job 3</td><td>Pkt #
Job 4</td><td>Pkt #
Job 5</td><td>Pkt #
Job 6</td></tr>
<tr><td>10</td><td colspan="3">NO ABA NO's. IN SEG.</td><td colspan="3">BASE ADDRESS FOR COL. S.P.</td></tr>
<tr><td>11</td><td colspan="3">ABA CODE #1</td><td colspan="3">ZERO</td></tr>
<tr><td></td><td colspan="6"></td></tr>
<tr><td></td><td colspan="6"></td></tr>
</table>

Next Seg.

*First Seg. only.

8.7 SPNXXX FILE FORMAT

The SPNXXX files will be formated according to the GEload requirements. See CPB 1518B, MME GENEWS and the SPAWN REQUIREMENTS. Also see the CONVERT utility package for writing spawn job files.

8.8 6000 POCKET SELECT TABLE GENERATOR INPUT FORMS

The input parameters to the PSTG will be in the form of a decision chart. The chart will be a preprinted form listing all the options and/or tests available which can be made on a field within a document. The user can simply check the tests he wants on a field and document to verify and sort it.

Sort tables to which the fields within the document are to be sorted will be named and constructed either during or before generating the test or sort logic. If the sort tables are generated prior to the sort logic then the PSTG must be able to address the table from storage when it is required during the generation of sort logic.

The PSTG will generate, from the input decision form and table data, scan tables in the form shown under 8.12 "SCAN TABLES". These tables along with the run number for which they are used will be stored on a "run parameter" perm file. As jobs are requested during production the 6000 DEP will read the run parameters from this file and send them to the 355 DHCP.

8.9 MASTER TRANSIT TABLE USER INPUT FORM

A. Master Ident - Identifies the table as a master table. The first digit must be "M". The next two digits may be any number or letter.

B. Sequence Number - Sequence numbers may start with 6001 and extend to 9999. Each entry should contain a sequence number and must be in ascending sequence.

C., D. and E. FRB - ABA - The Federal Reserve District, City State and American Banking Association codes must be provided by the users in this field. The user may give the FRB only or the FRB and City State or the FRB, City State, and the ABA number.

F. Multiple ABA's - When the user wishes to tie several ABA's to the same send point and/or pocket number, an asterisk must be placed in this field. Succeeding ABA's can then be placed in columns 17-80 and separated by dashes. However, if this is utilized, a previous line must have defined all other required control input, such as run number, send point, pocket number, etc.

8.9 MASTER TRANSIT TABLE USER INPUT FORM (continued)

G. Prime Run Number - This field must contain the run number of the first run (first sort pass or prime run) which requires the table for sorting.

H. Prime Pocket Number - The pocket number which is to be used on the first run (prime run) is entered here. This corresponds to the run number entered in field G. Depending on the sort in use, up to 32 pockets, 1 - 32, may be indicated.

I. Alternate Pocket Number - An alternate pocket, 1 - 9, may be entered here for the prime run. This field is used in conjunction with field G and H. If no alternate pocket, this field will be left blank.

J. Send Point Code - The send point code may be any value from 000 - 999, or if no send point code, this field may be left blank. However, if sorting is to take place by send point, this field must contain a send point code.

K. Float - The float code, or number of days float, for each send point may be entered here. If no float, this field may be left blank.

L. Secondary Run Number - This field may contain the run number of the second run (sort pass) which requires the table for sorting.

M. Pocket Number - The pocket number which is to be used on the second run is entered here. Depending on the sorter in use, up to 32 pockets 1 - 32 may be indicated.

N. Alternate Pocket Number - An alternate pocket, 1 - 9, may be entered here for the second run. If no alternate pocket, this field will be left blank.
O,P,Q Same as L,M,N for the third run, which requires the table.
R,S,T Same as L,M,N for the fourth run, which requires the table.
U,V,W Same as L,M,N for the fifth run, which requires the table.

X. Big Items - Amount - The amount in hundreds, thousands, and ten thousands which is to be considered a big item amount for the ABA's or send point may be entered here. If no big item amount, this field will be left blank.

Y. Big Item Pocket Number - The pocket number to which big items are to be selected may be entered here. Item which are equal to or greater than the amount given in field Y will be selected in this pocket.

8.9 MASTER TRANSIT TABLE USER INPUT FORM (continued)

Z. Send Points - Collection Time Codes - There are 24 time periods which the user may designate as valid for items going to the various send points. The user may indicate in this field which time periods are valid for the send point given in field J. If at the time of processing the time of day falls within the designated times, the document will be pocketed in the normal pocket. If the time of day is outside of the range of times given in the timetable, the document will be tested for a big dollar item. If the document is a big dollar item, it will be selected to the big item pocket. When no big item value or pocket is given, documents will be selected to their normal pockets.

The user must provide the corresponding collection timetable if he chooses to use this feature. See the Miscellaneous Tables Input writeup fields F, I, J and K.

8.10 PSTG MASTER TABLE FILE

	CARD COLS			
A	1-3	MASTER IDENT.		
B	4-7	SEQ. NUMBER		
C	8-9	DIST.	FRB	FRB-ABA
D	10-11	CITY/STATE		
E	12-15		ABA	
F	16	MULTIPLE ABA'S		
G	17-19	PRIME RUN NUMBER		
H	20-21	PRIME POCKET NO.		
I	22	ALT PRT		
J	23-25	SEND PT. CODE		
K	26-27	FLOAT		
L	28-30	RUN NO.	1	SECONDARY RUN NO'S. AND POCKETS
M	31-32	PRT NO.		
N	33	ALT PKT		
O	34-36	RUN NO.	2	
P	37-38	PKT NO.		
Q	39	ALT PKT		
R	40-42	RUN NO.	3	
S	43-44	PKT NO.		
T	45	ALT PKT		
U	46-48	RUN NO.	4	
V	49-50	PKT NO.		
W	51	ALT PKT		
	52	10,000	AMT.	BIG ITEMS
X	53	1,000		
Y	54	100		
	55-56	PKT No.		
	57	T 1		SEND POINTS - COLLECTION TIME CODES -
	58	T 2		
	59	T 3		
	60	T 4		
	61	T 5		
	62	T 6		
	63	T 7		
	64	T 8		
	65	T 9		
	66	T 10		
Z	67	T 11		
	68	T 12		
	69	T 13		
	70	T 14		
	71	T 15		
	72	T 16		
	73	T 17		
	74	T 18		
	75	T 19		
	76	T 20		
	77	T 21		
	78	T 22		
	79	T 23		
	80	T 24		

PSTG builds MASTER File from this USER INPUT.

See:
- o FRD Segment
- o CITY/STATE Segment
- o ABA Segment

8.11 GENERAL TABLES USER INPUT FORM

A. Ident

The ident will be preprinted to "GT", which indicates the input is a general table.

B. Run Number

The run number or job number for which the table is used must be given in this field.

C. Sequence Number

Sequence numbers may start with 0001 and extend to 9999. Each entry should contain a sequence number and must be in ascending order.

D. Table Name

The table name must appear on the first entry. Subsequent entries in the same table need not contain the name.

E. Sort Indicator

An "S" must be placed in this field if the table is a sort table, i.e., will be used to select pockets for documents. If the table or data is not for sorting purposes, this field will be left blank.

If an "S" is placed in this field, field "N" must contain the pocket number.

F. Table Type

There are six table types that the generator will recognize and one of the names listed below must be placed in this field.

TCDT: Transposition Check Digit Table

If the user specified a TCD other than G10, G11, N10, N11 or alternate two's, this field must contain TCDT.

TRLT: Transaction Code Transliteration Table

This field must contain TRLT if a transliteration table is to be generated.

8.11 GENERAL TABLES USER INPUT FORM (continued)

TIME: Timetable

The timetable is used along with the Master Transit Table Send Points - Collection Time Codes. The timetable may consist of 24 entries 01-24. The codes must be given in 26-27 and the time of day in 28-35.

PKTT: Pocket Table

The user may construct his own pocket table given item counts beginning in Col. 28 and pocket number and type in fields "M" and "N".

MISC: Miscellaneous Table

This applies to all tables, either sort or non-sort, which consist of an argument only or an argument and a pocket number.

CNST: Constant

This indicates the table is a constant only.

G. TCD ID

This field is used in conjunction with field "F". If field "F" contains TCDT, then this field must contain TCD identification number 0-9. This identification number must be referred to, for special TCD, in field "V" of the scan record input formats by a "T" followed by the ID number 0-9.

H. Length of Argument in Characters

When a single argument is given, the length of it must be specified in characters in this field. If this field contains a comma in Col. 24, the field length will be determined by the number of digits between commas in the data entered in Cols. 28 - 76.

I. TCD Division or Time Codes

Depending on Field "F" (TCDT or TIME) this field will contain the TCD divisor 01-99 or the time of day codes of the times appearing in fields "J" and "K".

J. and K. Collection Times

These fields are used in conjunction with fields "F" and "I". If "F" contains TIME, and "I" a code 01-24, then the times of day 0000 - 2400 must be entered in these fields.

8.11 GENERAL TABLES USER INPUT FORM (continued)

L. Table Data

Other than "J" and "K" above, table data will begin in Col. 28 and may continue through 76. A single argument may be given with the character count in field "H", or multiple arguments may be given, separated by dashes through Col. 76.

M. Pocket Type

This field used in conjunction with Field "F". If Field "F" contains PKTT, then pocket type may contain the following codes:

K = Kill pocket

R = Rerun pocket

N. Pocket Number

This field contains pocket number for the sort table or pocket table. It is used in conjunction with fields "E" and "F".

O. Alternate Pocket

An alternate pocket, 1 - 9, for the prime pocket in field "N" may be given here. If no alternate pocket, this field may be left blank.

GENERAL TABLES USER INPUT FORM - DES 6000

USER INPUT

										28 TABLE DATA →				
										COLLECTION TIMES TIME OF DAY				
	IDENT	RUN NO.	SEQ. NUMBER	TABLE NAME	SORT IND.	TABLE TYPE	TCD ID. NO.	LENGTH OF ARG. IN CHARS.	TCD DIV. OR TIME CODES 01-24	FROM 00 2400	TO 00 2400		PKT TYPE	Pckt No.
COLS	1-2	3-5	6-9	10-17	18	19-22	23	24-25	26-27	28-31	32-35	36 76	77	78-79
	A	B	C	D	E	F	G	H	I	J	K	L	M	N

PSTG GENERATES INPUT ON THIS FORM ONTO THE TABXXX FILES, ALONG WITH THE SCAN RECORDS.

8.12 VERBS (SCAN RECORDS)

Action Verbs provide the user with complete logic flexibility for building sort runs. Validation and pocket selection of documents, including control documents, can be readily processed with the variety of action verbs offered. These verbs, along with the pocket tables and general tables, provide the user an easy to implement sort package which should cover nearly any kind of sort requirement. The action verbs by category are as follows:

1. Field/Subfield Identification Verbs

   - GETFIELD
   - GETSUBF
   - VALIDATE*

2. Test/Validation Verbs

   - CKFORMAT
   - CHECKTCD
   - TEST
   - SEARCH
   - DIGITEST
   - VALIDATE*

3. Logic Control Verbs

   - ENDTEST
   - SETSW
   - RESETSW
   - TESTSW
   - TRANSFER

4. Document Handler Control Verbs

   - STARTRUN
   - ENDRUN
   - TYPEOUT
   - HALTDH

5. Special Purpose Verbs

   - MASK
   - CKPOINT
   - SVEFLD

6. Device and Pocket Control Verbs

   - DEVICE
   - SETVALUE

*Combines GETFIELD and CKFORMAT verbs.

8.12 VERBS (SCAN RECORDS) (continued)

7. File Control and Job Control Verbs

- o FILECTL
- o *SPAWNJOB

8. SLAVE Program to DEP Communication

*SPAWNJOB is provided through the use of the SPAWN control card see PSTG assembly control cards.

The PSTG will generate these verbs, from the user input forms into given scan record formats and file them onto the TABXXX file.

SCAN records are referenced from the User Input Forms shown on page

See User Section 13.0

VERBS	PAGE
GETFIELD	220
GETSUBF	221
VALIDATE*	222
CKFORMAT	223
CHECKTCD	224
TEST	225
SEARCH	226
DIGITEST	227
VALIDATE*	
ENDTEST	228
SETSW	229
RESETSW	230
TESTSW	231
TRANSFER	232
STARTRUN	233
ENDRUN	234
TYPEOUT	235
HALTDH	236
MASK	237
CKPOINT	238
SVEFLD	239
DEVICE	240
SETVALUE	241
FILECTL	242
*SPAWNJOB	

Legion

Letters and numbers below words =
corresponding field from users scan input forms.
* = PSTG computed or tested

% = General Table

C = Master Tables

GETFIELD

Test Type 01

Word	0 5	6 17
0	Length of Scan Record	Transfer Index
	*	Z and 2

	0 17
1	Control Word
	G, I, J, K, L, Z and 2

	0 5	6 11	12 17
2	Test Type 01	Field ID #	Field Location in document IF
	E	H and *	Bit 7 of CW = 1

	0 5	6 11	12 17
3	Error Code	Accept Pocket	Error Pocket
	3	Y	1

	0 5	6 11	12 17
4	1st Starting Que	2nd Starting Que or Skip Count	Min. Field Length
	I	J	N

	0 5	6 11	12 17
5	1st Ending Que	2nd Ending Que or Skip Count	Max. Field Length
	K	L	M

	0	1	2	3	5	6 11	12 17
6	A	B	C	D		Zero	Scale Count by Digit
	P	P	J	L		4	R

GETFIELD

A set to 1 = Scaling required.

A set to 0 = No scaling required.

B set to 1 = Right hand scaling required.

B set to 0 = Left hand scaling required.

C set to 1 = Word 4 bits 6-11 are a que skip count.

D set to 1 = Word 5 bits 6-11 are a que skip count.

Legion

Letters and numbers below words =
corresponding field from users scan input forms.

* = PSTG computed or tested

% = General Table

C = Master Tables

GET SUBF

Test Type 03

Word 0

0 5	6 17
Length of Scan Record	Transfer Index
*	Z and 2

1

0 17
Control Word
G, Z and 2

2

0 5	6 11	12 17
Test Type 03	Que Field ID Number	Subfield within Que Field ID #
E	H and *	S and *

3

0 5	6 11	12 17
Error Code	Accept Pocket	Error Pocket
3	Y	I

* 4

0 5	6 11	12 17
Beginning Loc. of Subfield in Que field	No. of digits in subfield	UA
Q	R	

*Word 4 bits 0-5 contain starting location of subfield within the que field. The starting location within the que field is given left to right with the first position (after the que) equal to zero.

VALIDATE

The VALIDATE verb will consist of two other verbs; GETFIELD and CKFORMAT. The PSTG will recognize the VALIDATE verb and construct two verbs, the GETFIELD and CKFORMAT, to enable the 355 DHCP to execute the required logic.

The PSTG will generate the GETFIELD verb first followed by the CKFORMAT verb. The error pocket, if one, will be inserted in both verbs. The success pocket if one, will be inserted in the CKFORMAT verb only.

If an XFER or success is given it will be placed in the CKFORMAT verb only. An XFER on failure will be placed in both verbs.

Bits 12-15 of the control card will be set as follows for the GETFIELD and the CKFORMAT.

VALIDATE

VALIDATE verb has XRER on success.

GETFIELD verb control word

bit 12 = 1
bit 13 = 0
bit 14 = 1 if no ENDTEST after VALIDATE
bit 14 = 0 if ENDTEST follows VALIDATE
bit 15 = 0

CKFORMAT verb control word

bit 12 = 1
bit 13 = 1
bit 14 = 1 if no ENDTEST after VALIDATE
bit 14 = 0 if ENDTEST follows VALIDATE
bit 15 = 0

VALIDATE

VALIDATE verb has XFER on failure

GETFIELD verb control word

bit 12 = 1

bit 13 = 0

bit 14 = 1

bit 15 = 1

CKFORMAT verb control word

bit 12 = 0 if ENDTEST follows VALIDATE

bit 12 = 1 if no ENDTEST follows VALIDATE

bit 13 = 0

bit 14 = 1

bit 15 = 1

VALIDATE

VALIDATE has no XFER

GETFIELD verb control word

bit 12 = 1

bit 13 = 0

bit 14 = 0

bit 15 = 0

CKFORMAT verb control word

bit 12 = 0 if ENDTEST follows VALIDATE

bit 12 = 1 if no ENDTEST follows VALIDATE

Bit 13 ∓ 0

bit 14 = 0

bit 15 = 0

CKFORMAT

TEST TYPE - 04

		0 5	6 17
Word	0	Length of Scan Record	Transfer Index
		*	Z and 2

		0 17
	1	Control Word
		G, Z and 2

		0 5	6 11	12 17
	2	Test Type 04	Field ID #	
		E	H	

	3	Error Code	Accept Pocket	Error Pocket
		3	Y	1

	4	MAX digit count before a dash + or -	MIN digit count before a dash,+ or -	D	E	F	G	H	I
		T	T	*	G	G	U		
*Y	5	MAX digit count after a dash + or -	MIN digit count after a dash, + or -	E	E	F	G	H	I
		T	T	*	G	G	U		

D = Set to 1 end of Format Test
e = 1 = A dash IS expected after MIN or MAX digit count
F = 1 = a + or - is expected after MIN or MAX digit count
e and F = 0 = NO dash or + or - expected after MIN or MAX count
In this case the count of digits and no dash is acceptable.
G = 1 = accept a non-numeric in place of a dash (error signal for MRS 200, Oct. 13).
*Y repeated for each dash, + or -.
I = UA. H = UA.

CHECKTCD

* TEST TYPES 08, 09, 10

Word 0	0-5	Length of SCAN record	*
	6-17	Transfer Index	Z and 2
1	0-17	Control Word	Z and 2
2	0-5	Test Type *08 - 10	# and, V, W.
	6-11	Field ID No. To be tested	H and *
	12-17	02-63 Divisor	V
3		Error Code	3
		Accept Pocket	Y
		Error Pocket	1

* Test Type 08 = Alternate 2's TCD
Test Type 09 = Natural TCD
Test Type 10 = Mod 10 Geometric TCD

The Pocket Select Generator will determine the TCD type from the users input form and give the 355 DHCP the proper verb type including the divisor.

The 355 DHCP will handle these TCD types through standard TCD routines per TCD section of the EPS.

CHECKTCD (SPECIAL)

Test Type 11

	0 5	6	17
Word 0	Length of SCAN Record	Transfer Index	
	*	Z and 2	
	0		17
1	Control Word		
	Z and 2		
	0 5	6 11	12 17
2	Test Type 13	Field ID No. to be tested	UA
	E, and V, W	H and *	
3	Error Code	Accept Pocket	Error Pocket
	3	Y	1
4	Pointer to TCD Table		
	4, 5 and %		

The user must supply the necessary TCD divisor remainder and multipliers. The 355 DHCP program will do a table load-up, multiplying each in the field by the multiplier in the table, divide this result by the given divisor and check the given remainder.

Legion

Letters and numbers below words = corresponding field from users scan input forms.

* = PSTG computed or tested.

% = General Table

C = Master Tables

TEST

TEST TYPE 05

WORD	0 – 5	6 – 17	
0	Length of SCAN Record	Transfer Index	
	*	Z and 2	
	0 – 17		
1	Control Word		
	C, 5, Z and 2, X.		
	0 – 5	6 – 11	12 – 17
*2	Test Type 05	Field ID #	Bit 8 = 1 Field ID # at 2nd Compare Field
	E	H and *	5 and *
3	Error Code	Accept Pocket	Error Pocket
	3	Y	1
**4	Pointer to the Table or Constant		
	4, 5*, % or @		

*Bit 8 control will be set to 1 if a field-to-field compare is required and 12-17 of word 2 will contain the 2nd compare field.

**Word four (4) will contain the pointer to the table or constant if bit 8 of control word = 0. If bit 8 is set to 1 word 4 will be set to zero. The 355 DHCP will set a pointer from the field ID found in the document to locate the same.

Legion

Letters and numbers below words = corresponding field from users scan input forms.

* = PSTG computed or tested

% = General Table

@ = Master Tables

SEARCH

Test Type 07

	0 5	6 17
Word 0	Length of Scan Record	Transfer Index
	*	Z and 2

1	Control Word
	@, Z and 2 and X

	0 5	6 11	12 17
2	Test Type 07	Field ID #	Pocket Field 0, 1 or 2
	E	H	Dep. at load time, zero.

	0 5	6 11,	12 17
3	Error Code	Accept Pocket	Error Pocket
	3	Y	1

	0 17
4	Pointer to Table
	%, @ and *, 4 and 5

Field Segment Control

	0		17
** 5	A 0 5	B 6 11	C 12 17
	@	@	@

*Bits 4, and 5 of the control are used as follows:

SEARCH

Test Type 07

Bits	Use
4	Set to 0 = BCD Table Set to 1 = Bin Table
5	Set to 0 = Segmented Table Set to 1 = Non-segmented Table

**A, B, and C contain number of digits in each segment of the field and table.

Legion

Letters and numbers below words = corresponding field from users scan input forms.

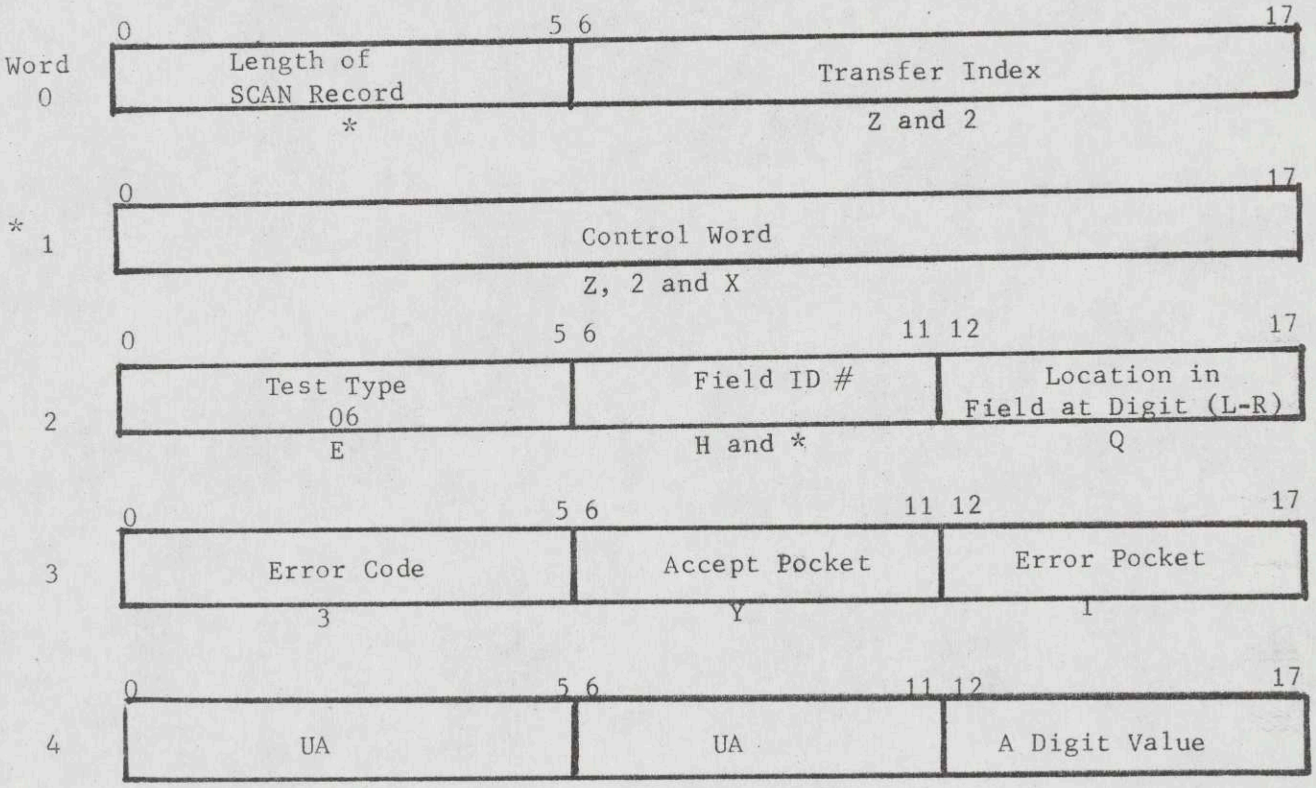
* = PSTG computed or tester

% = General Table

C = Master Tables

DIGIT TEST

Test Type 06



ENDTEST

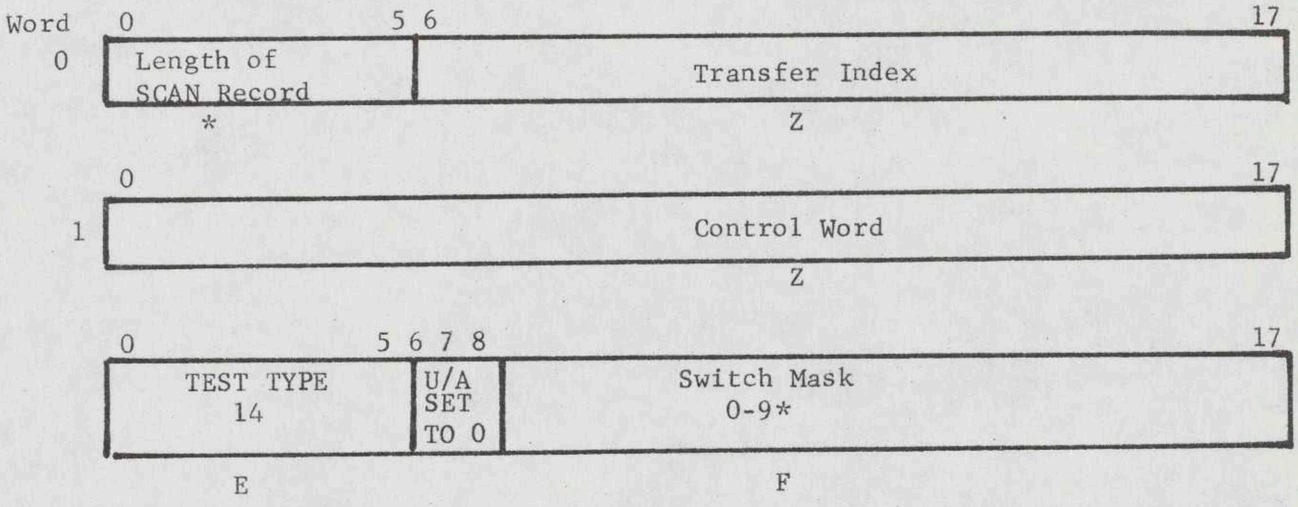
Test Type 12

0	Length of SCAN record	UA
1	bits 12, 13, 14, 15 = 0	
2	Test Type = 12	UA

The ENDTEST verb signals the DHCP that all testing has been (for a document) completed and to pocket the document.

SETSW

TEST TYPE 14



*The PSTG will convert the users switch settings 0-9 to 8-17.

The 355 DHCP will use 8-17 as a mask to set corresponding bit or bits in the switch word in the system control area. There will be a switch word in the system control area for each handler 1-6.

RESETSW

TEST TYPE 15

Word	0 5	6 7	8 17
0	Length of Scan Record *	Transfer Index Z	
1	Control Word Z		
	Test Type 15 E	U/A SET TO 0	Switch Mask 0-9 * F

*The PSTG will convert the users switch settings 0-9 to 8-17. The 355 DHCP will use 8-17 as a mask to set the corresponding bit or bits in the switch word in the system Control area off (set to 0). There will be a switch word in the system control area for each handler 1-6.

TESTSW

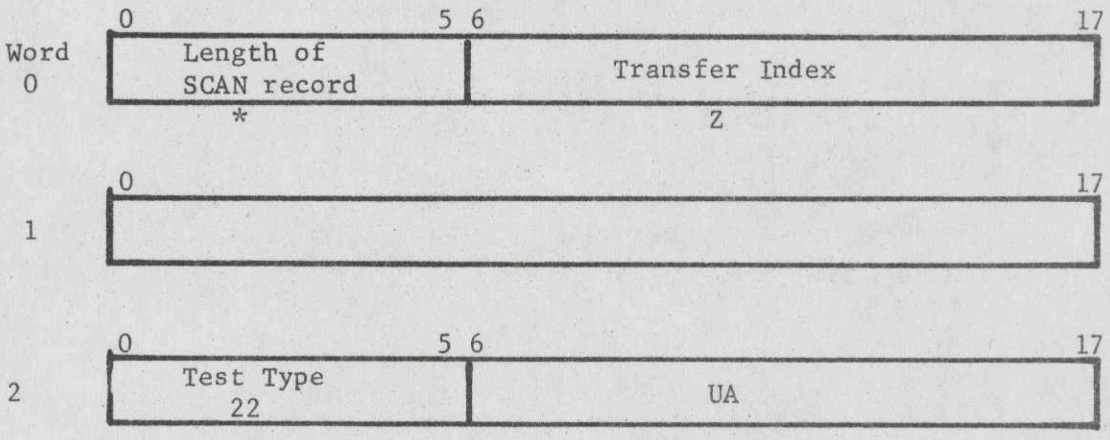
TEST TYPE 16

Word			
0	0-5	Length of Scan Record	*
	6-17	Transfer Index	Z and 2
1	0-17	Control Word	Z and 2
2	0-5	Test Type 16	E
	6-7	U/A SET TO 0	
	8-17	Switch Mask 0-9 *	F
3	0-5	Error Code	3
	6-11	Accept Pocket	Y
	12-17	Error Pocket	1

*The PSTG will convert the users switch settings 0-9 to 8-17. The 355 DHCP will use 8-17 as a mask to test the corresponding bit or bits in the switch word in the system control area. There will be one switch word in the System Control area for each handler 1-6. The switch is considered to be on (Set to 1) when both corresponding bits are set to 1.

TRANSFER

TEST TYPE 22

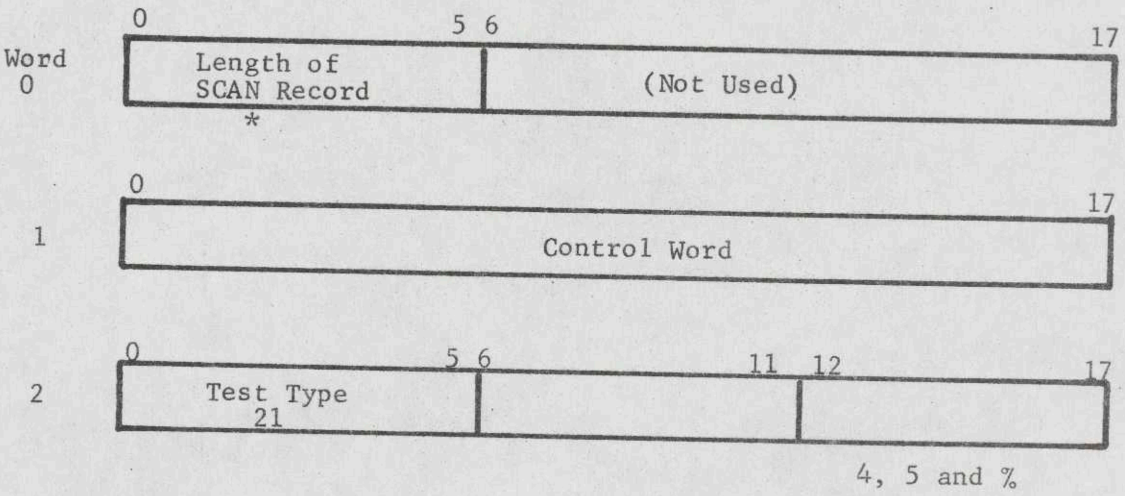


This verb will cause the 355 DHCP to execute an unconditional transfer to the location given in word 0, bits 6-17

NOTE: If a verb, (other than the Transfer verb) has both success and failure transfer specified then a transfer verb will be generated for the success transfer by the PSTG.

ENDRUN

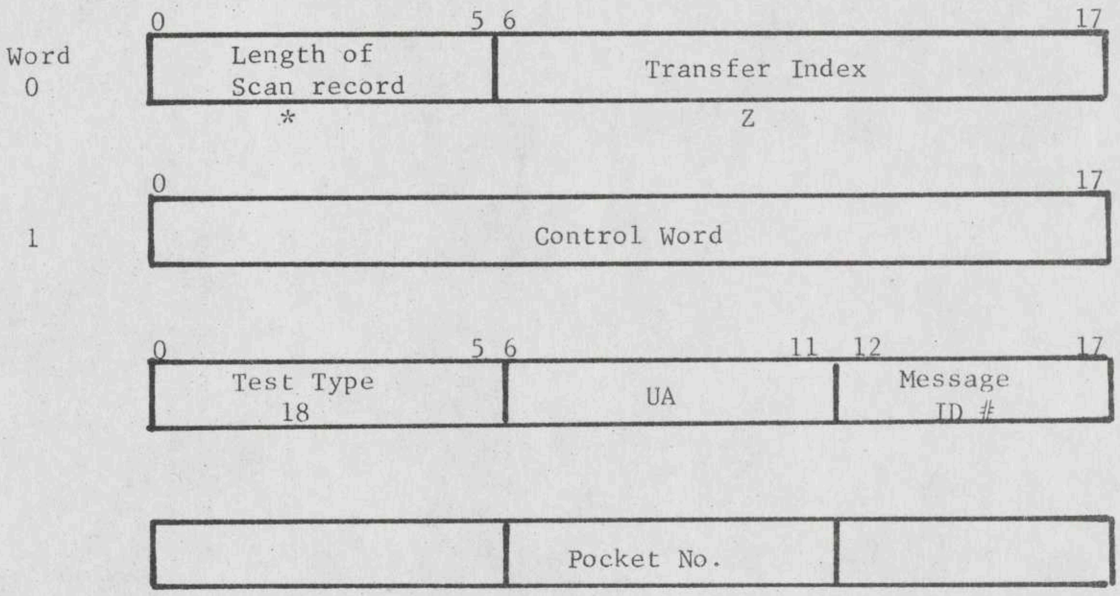
TEST TYPE 21



When the ENDRUN verb is detected in the 355 DHCP, the test verb will be stored in the document status word (word 6-bits 0-5) and the document handler stopped. All following documents will be rejected (after the stop).

TYPEOUT

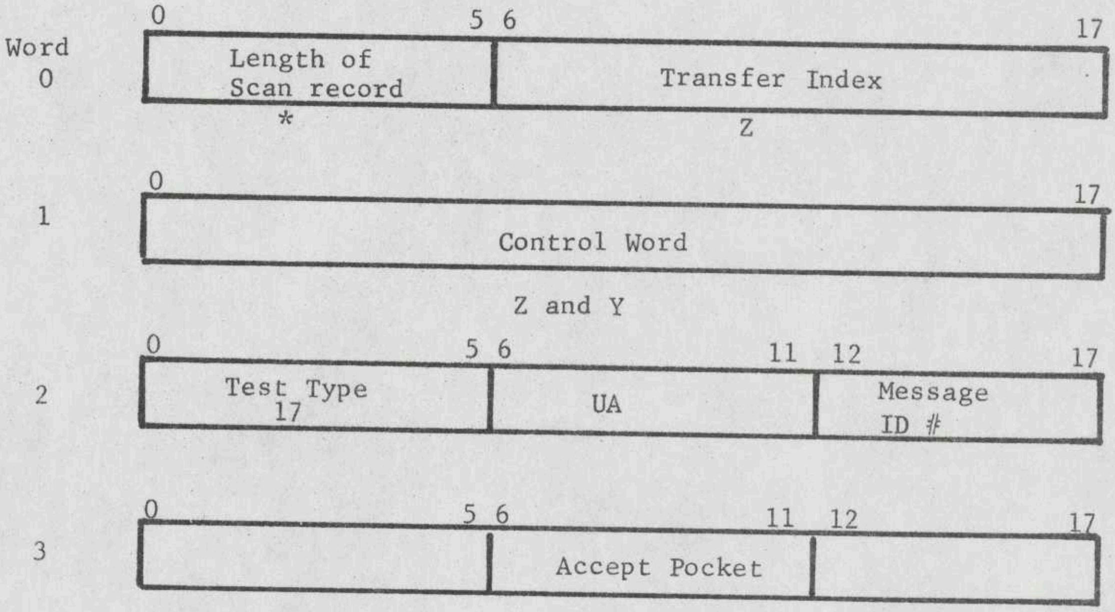
Test Type 18



On a "typeout" the 355 DHCP will place the test type (18) in the comm status word (word 7 bits 0-5) and the message ID no. in the document status word no. 7, bits 0-5.

HALTDH

Test Type 17



On a HALTDH the 355 DHCP will issue a controlled stop to the document handler, put the document in the given pocket (word 3 bits 6-11), and place the type (17) in the comm status word (word 7 bits 0-5) and the message ID# in word 7, bits 0-5, of the document status word. All documents which follow the HALTDH will be sent to the pocket no. given in word 3 bits 6-11 or the last pocket received for the document.

Legion

Letters and numbers below words = corresponding field from users scan input forms.

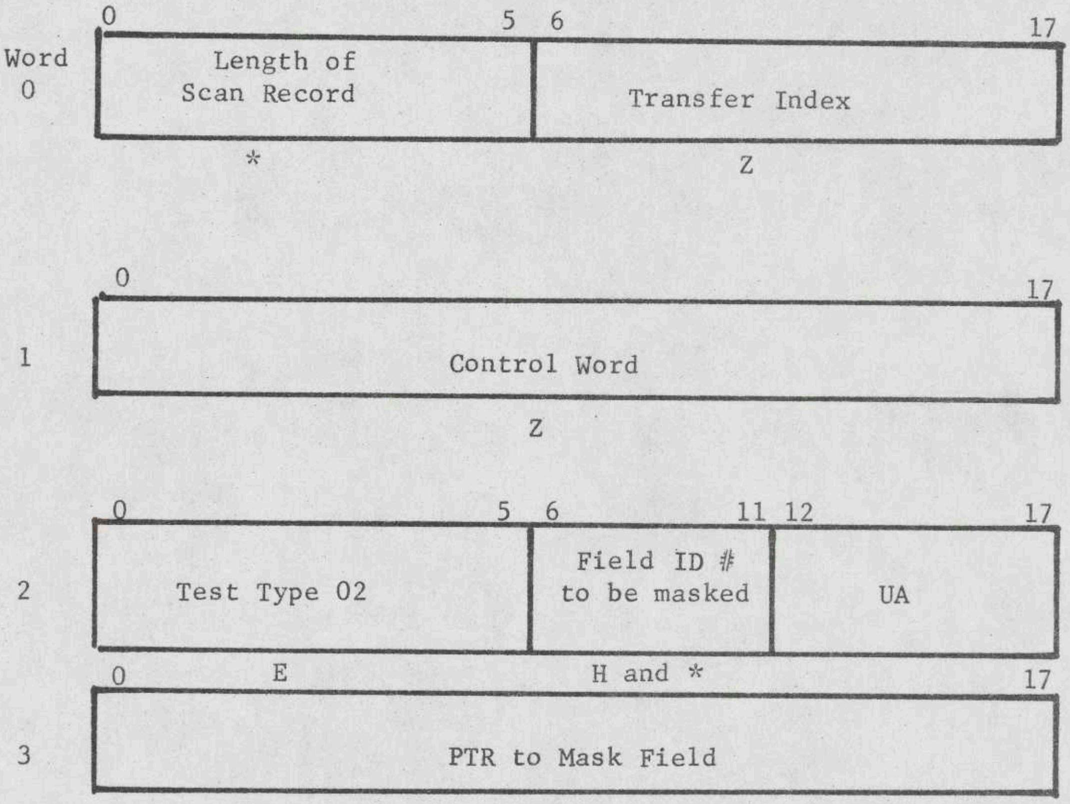
* = PSTG computed or tested

% = General Table

C = Master Tables

MASK

Test Type 02



The mask verb is used to zero non-significant high order positions in a field. The field given in word 2 bits 6-11 is the field to be masked. The field given in word 3 bits 0-17 is the field from which the mask will be derived. The high order position of the field given in word 3 bits 0-17 will be searched until a non-zero digit is found. The no. leading zeros found must be applied to the same high order positions of the field given in word 2 bits 6-11.

CKPOINT

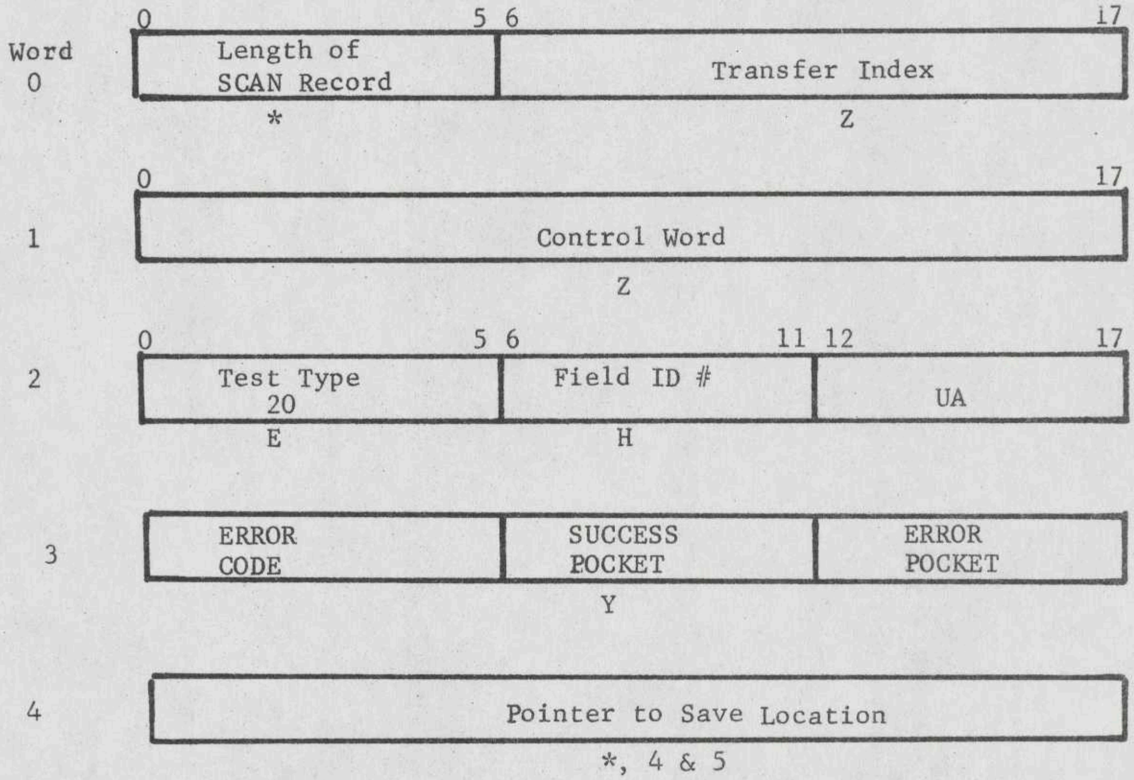
Test Type 19

Word	0 5	6 11	12 17
0	Length of Scan record *	Transfer Index Z	
1	Control Word		
2	Test Type 19	UA	Contains an * PSTG places an *
3		Pocket No. Y	

The 355 DHCP will place the verb type in the status word (word 6 bits 0-5).

SAVEFIELD

TEST TYPE 20



DEVICE

The DEVICE verb does not require a scan record. It till be used to identify the device type (sorter type) 6000 PSTG that the sort job is programmed for. The PSTG will use this information for diagnostic purposes. At the present time there are two device types:

MRS 200 - 12 Pockets

DRD 200 - 2 Pockets

*H236 - up to 32 pockets

*Not included in present system.

8.13 SETVALUE and Pocket Select Control Tables

The SETVALUE verb will not require a scan record. It will be used by user to set up pocket control information about control documents and pocket item counts. The PSTG will use this information to construct the pocket select control tables.

The SETVALUE verb will be used in conjunction with three pocket control codes: MI-M3 will be for defining multi pocket separators, I1-I3 will be used for a kill type separator to be pocketed depending on the I pockets item count, and C which will be used to control the count of items that all pockets will contain before a full pocket condition is reached.

The 355 DHCP will, before pocketing any document, determine if the document is an M, or I type document by testing the accept pocket value for a value of octal 50-55. An octal 50-52 will equal an I type document and an octal 53-55 will equal an M type document.

The SETVALUE used with a "C" control code sets the full pocket count in word 1 (by the PSTG) for each pocket. The 355 DHCP will keep a program count for each pocket, using word 11, and compare this count to the full pocket count. When the actual count exceeds the full pocket count the 355 DHCP will set bit 1 of word 0 on for that pocket, turn on the full pocket light and stop the handler. The reason code will be put in the documents status word to inform the 6000 DEP the reason for the stop.

When the operator on the device empties the full pocket, the full pocket light will turn off and depressing the operate button will re-start the handler. The 355 DHCP will always test bit 1 whenever a full pocket is indicated by count and, if on, interrogate the full pocket status bit of the devices status word. If the status bit is no longer on, bit 14 will be set to zero and the program count for the pocket, word 11, set to one and normal processing resumed.

SET VALUE

ACTION VERB

SCAN FORM 10-17 18 ____________

SETVALUE C (00-11,300)

The PSTG will set the full pocket count at 300 for all pockets. The count of 300 will be put in word 1 of the pocket select control table for all pockets.

8.13 SETVALUE and Pocket Select Control Tables (continued)

SET VALUE

ACTION VERB

SCAN FORM 10-17 18 __________

SETVALUE I1(02,200.220)(03,200,300)

The PSTG will put the minimum count given in word 1 and the maximum count in word 4 of pocket select control table for pockets two and three. Bit 12 of word 0 will be set for 1 for pockets two and three on the pocket select control table. The PSTG will make I1=50, I2=51, I3=52.

When an M type document is presented to the 355 DHCP for pocketing, the 355 DHCP will:

a. Determine if it is the first M type document received in the series.

b. If it is the first M type document in the series,set up to ensure that the following documents are the same M type until all pockets in sue for the device have received one.

SET VALUE

ACTION VERB

SCAN FORM 10 - 17 18 __________

SET VALUE M1 (00-11)

PSTG will set bit 12 and 17 of word 0 of the pocket select table to 1 for all pockets 0-11. In the event of a 32-pocket device and the M(00-31) is used, then all pockets 00-31 will be indicated as being in use by setting bit 12 & 17 on for each pocket. PSTG will make M1=53.

ACTION VERB

SCAN FORM 10 - 17 18 __________

SET VALUE M2 (00,6,7)

8.13 SETVALUE and Pocket Select Control Tables (continued)

PSTG will set bit 12 and 17 of word 0 of the pocket select table to 1 for pockets 0, 6 and 7. PSTG will make M2=54.

PSTG will make M3 = 55 and will set bit 14 on of word 0 of the pocket select table.

SET VALUE

ACTION VERB

SCAN FORM 10 - 17 18 ___________

SETVALUE C(00-11,300)

The PSTG will set the full pocket count at 300 for all pockets. The count of 300 will be put in word 1 of the pocket select control table for all pockets.

ACTION VERB

SCAN FORM 10 - 17 18 ___________

SETVALUE I1(02,200.220)(03,200,300)

The PSTG will put the minimum count given in word 1 and the maximum count in word 4 of pocket select control table for pockets two and three. Bit 12 of word 0 will be set to 1 for pockets two and three in the pocket select control table. The PSTG will make I1=50, I1-51, I3=52.

When an M type document is presented to the 355 DHCP for pocketing, the 355 will:

a. Determine if it is the first M type document received in the series.
b. If it is the first M type document in the series, set up to ensure that the following documents are the same M type until all pockets in use for the device have received one.

8.13 SETVALUE and Pocket Select Control Tables (continued)

c. If it is not the first in a series of M type documents, place it in the next pocket in use which requires one, or if there are no more pockets requiring one, reject it.

d. When the first non-M type document is received following a series of M type documents, ensure each pocket in use received an M type document. If each pocket in use for the device has received an M type document, resume normal processing until next M type document is received repeat steps a - d over. If all pockets did not receive an M type document in the series, issue a controlled stop, reject documents in transit -- do not process these (must be re-read) and place the M type value in the document status communicator word (word 7 bits 0--5), octal 53, 54 or 55. This will inform the 6000 DEP a type out is required and what message to type.

When an I type document is presented to the DHCP for pocketing, the 355 DHCP will:

a. Determine which pocket requires an I type separator by testing bits 12 - 14 of word zero of the pocket select table.

b. If the I type document is required for the pocket, test the min and max counts in the pocket table against the program count.

If the count is within the min and max range put the document in the pocket and reset the program count for the I separator to 0. Also set the appropriate bit for the I separator received in word 7 bits 12 - 17 of the document's status words.

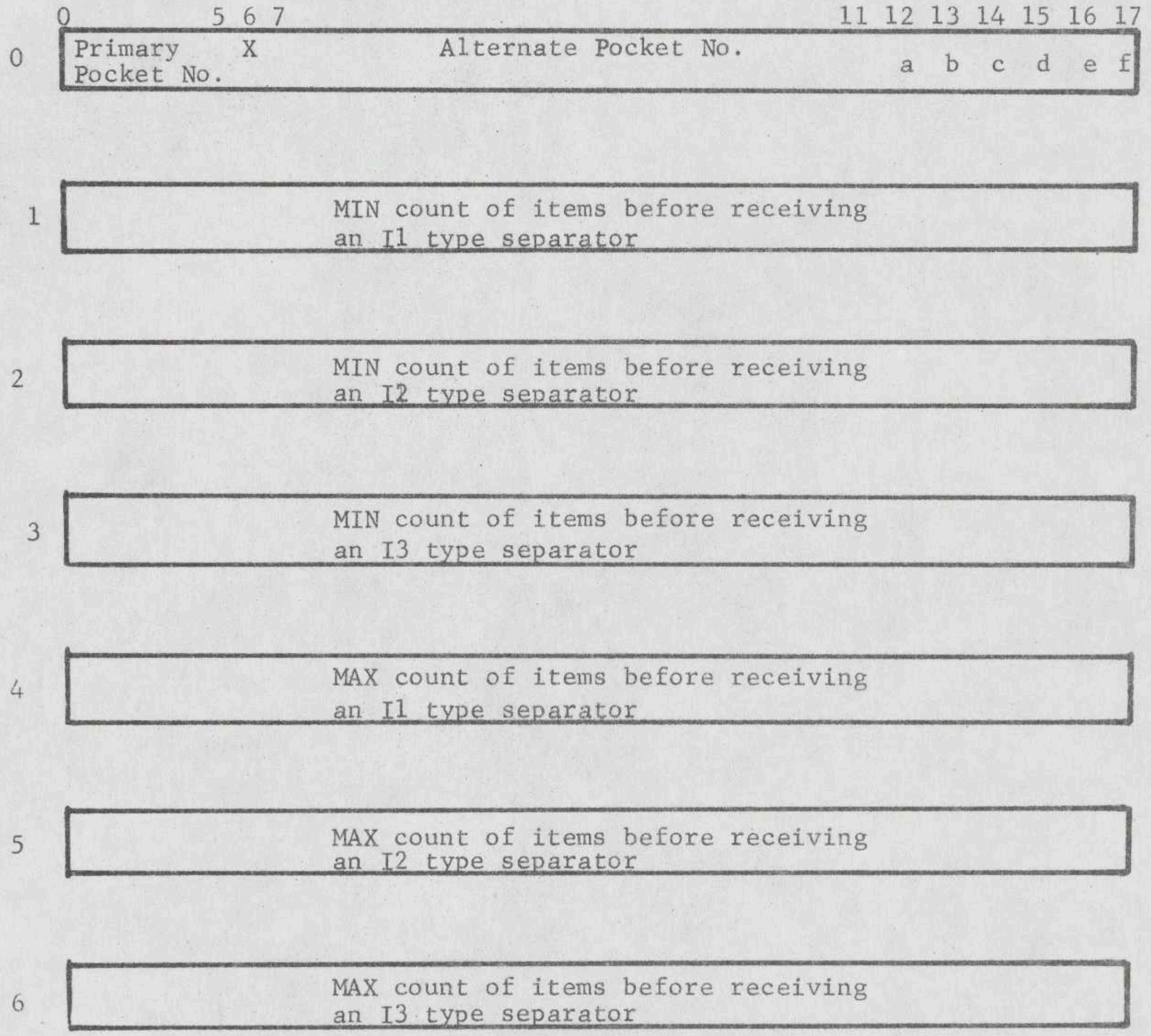
c. In the event no pocket requires the I document, it will be rejected.

d. When an I type pocket exceeds the max. count allowed and has not received an I type document, the 355 DHCP will pocket the item document and stop the handler (controlled stop). An octal 50 will be put in the documents status word bits 0 - 5.

e. When an I pocket has more items than the maximum count allows, step d. will be repeated until an I document is received. After an I document is received it will be pocketed, the count in word 4 reset to 0 and normal processing resumed.

Pocket Select Control Tables

Format and Usage



0 Primary Pocket No. X Alternate Pocket No.

1 MIN count of items before receiving an I1 type separator

2 MIN count of items before receiving an I2 type separator

3 MIN count of items before receiving an I3 type separator

4 MAX count of items before receiving an I1 type separator

5 MAX count of items before receiving an I2 type separator

6 MAX count of items before receiving an I3 type separator

Pocket Select Control Table

Format and Usage

7 Number of items counted by program since receiving an I1 separator

8 Number of items counted by program since receiving an I2 separator

9 Number of items counted by program since receiving an I3 separator

10 Number of items to be put in pocket before turning on full light Pkt.

11 Program count of number of items in pocket since turning off full light Pkt.

a = 1 = I1 separators in use for this pocket
b = 1 = I2 separators in use for this pocket
c = 1 = I3 separators in use for this pocket
d = 1 = pocket not in use
d = 0 = pocket in use
x = 1 = no alternate pocket in use
x = 0 = alternate pocket in use
f = to be used by the 355 DHCP set to 1 when pocket full. It will be set back to zero when full pocket light turned off.
e = pocket will require M type separator.

8.14 SCAN RECORD - CONTROL WORD USE

BITS	USE
0	Set to 1 = Format the field in the outlet location given in word 6 bits 5-11 of the GETFIELD verb set to 0 = no formatting.
1-3	100 = Numeric field 010 = Numeric field with dashes 001 = Numeric field which can contain plus, minus, or dash symbols. 110 reserved for 101 later use. These 111 will be used for alpha and alpha-numeric fields.
4	Bit 4 will indicate table type, BCD or BIN, for the TEST and SEARCH Verbs. 0 = BCD 1 = BIN
5	Bit 5 will indicate if a table is segmented or non-segmented. It will be used in the TEST and SEARCH verbs. 0 = non-segmented 1 = segmented table
6	Unassigned.
7	Set to 1 = no ques delimit field. The field will be located by the starting location given in word 2, bits 12-17, of the GETFIELD verb. Set to 0 = ques delimit the field
8	This bit is used in conjunction with the TEST verb. It will be set to 1 if the test to be made is between fields rather than a field to a constant or table. Set to 0 = The field is to be tested against a constant or table. Set to 1 = The field whose ID # is given in word 2, bits 6-11 of the TEST verb is to be compared against the field whose ID # appears in 12-17.
9-11	These bits will be used to indicate the test condition wanted - IE - Equal, less than, greater than, or a range. 000 = Equal 100 = Less than 001 = Greater than 010 = Unequal 101 = Range condition Field > 1st factor Field < 2nd factor

8.14 SCAN RECORD - CONTROL WORD USE (continued)

BITS	USE
12	This bit will be used to control success scan logic. Set to 0 = Stop scan logic on a successful test and pocket the document. Set to 1 = Continue testing if the test was successful.
13	This bit will be used in conjunction with bit 12. It will indicate whether to transfer on a successful test or continue sequentially to the next scan record. Set to 0 = No transfer continue to next sequential scan record. Set to 1 = Transfer to the scan record whose location is given in word 0 bits 6 = 17 of the scan record.
14	This bit will be tried to control unsuccessful (failure) scan logic. Set to 0 = Stop and pocket the document if the test has failed. Set to 1 = Continue to another scan record if the test has failed.
15	This bit is used in conjunction with bit 14. It will indicate a transfer on failure or continue to the next sequential scan record. Set to 0 = Continue to the next sequential scan record. Set to 1 = Transfer to the scan record whose location is given in word 0, bits 6 - 17 of the scan record.
16-17	Unassigned

Use of bits 12, 13, 14 and 15 of the control word.

The PSTG must set these bits appropriately so that the 355 DHCP can execute the scan records in proper sequence. These bits are used as follow:

- bits 12 and 13 control the success path of the scan records.
- bit 12 = 0 = stop testing on the success path and pocket the document. This will normally be set to 1.
- bit 12 = 1 = continue to another scan record for more testing.
- bit 13 = 0 = continue to next sequential scan record on success.
- bit 13 = 1 = transfer, use index in word 0 bits 6-17 to find next scan record on success.
- bits 14 and 15 control the failure patch of the scan records.

8.14 SCAN RECORD - CONTROL WORD USE (continued)

- o bit 14 = 0 = stop testing on the failure path and pocket the document.
- o bit 14 = 1 = continue to another scan record for more testing.
- o bit 15 = 0 = continue to the next sequential scan record.
- o bit 15 = 1 = transfer, use index in word 0, bits 6 - 17 to find next scan record.

8.15 PSTG CONTROL CARDS

There will be eight input control cards to the PSTG. These control cards control the assembly of scan records, tables, and assembly updates. Their description and use are as follow:

- INIT

1 - 6	42 - 47
"INIT"	"DES6000"

Signals the PSTG to initialize for a 6000 assembly.

- JOB

1 - 6	8 - 10	13 - 24
"JOB"	Job No. 001-999, or alfa or alfa-numeric job numbers	"DATE"

Provides the generator with the library identification number for filing the assembly. The TTY operator will also use this number to call the Job into execution.

- TABLE

1 - 6	13 - 24
"TABLE"	"ASSEMBLE" or "UPDATE"

Signals the generator that a table, either general type or master type, is to be assembled or updated. The ADD and DEL control cards are used with the "UPDATE" control word.

8.15 PSTG CONTROL CARDS (continued)

- SCAN

1 - 6	13 - 24
"SCAN"	"ASSEMBLE" or "UPDATE"

Signals the generator that SCAN tables follow and that either an "assembly" or assembly "update" is to be processed. The ADD and DEL cards are used with the "update" control word.

- SPAWN

1 - 6	12 - 18	20	24		30 - 37
SPAWN	IDENT	SYSTEM CONTROL	COPIES	SPAWN CODE	CONDITION CODE

Signals the generator that a user 6000 slave is to be placed in the library for the DEP to spawn, or the slave job whose IDENT is given will come through the job stream and use the sorted file or files created by the sort job. The object deck, for spawning must be set up for running under "spawn" as outlined in GECOS manual CPB 1518 B.

IDENT: Unique number of (alfa or alfa numeric) which identifies the 6000 slave job. This identifier will be put in the DEP slave control stack - see slave job management block.

SYSTEM CONTROL: Used to indicate respawning jobs after system disaster:
1 = respawn
2 = no respawn

COPIES: Single or multiple copies of the job may or may not be spawned:

S = single only
M = multiple
*N = Null - no spawn identifies a slave job which will come through the job stream.

8.15 PSTG CONTROL CARDS (continued)

SPAWN CODE: If multiple copies are to be spawned, "M" in Col. 24, given condition for spawning multiple copies:

1 = one copy for each sorter running the sort job.
2 = one copy each time the job is called from a sorter, i.e., each time the job is started.

CONDITION CODE: The condition under which this job is to be spawned.

- "CHECKPT" - spawn as soon as there is a sort checkpoint.
- "IMMED" - spawn as soon as the sort job starts.
- "ENDJOB" - spawn at the end of the sort job.
- Spawn this job on a document count.

  00000001 - 99999999

*N - the job is not to be spawned, e.e., no object deck follows the SPAWN card. The SPAWN card in this case identifies a slave program that will access DEP created files, but will come through the normal job stream.

NOTE: There may be more than one "spawn" card for NULL, "N" spawn jobs.

8.15 PSTG CONTROL CARDS (continued)

- DEL

1 - 6	8 - 13	15 - 20
DEL	Starting Seq. No. to be deleted	Ending Seq. No. to be deleted

The "DEL" control card signals the generator to delete the given sequence numbers. The "DEL" card may be followed by sequence cards replacing those numbers deleted and cards to be merged in (added) to the source program. The DEL control card must follow either a TABLE or SCAN control card. The cards to be deleted, added, or changed must be punched in standard scan or table form - see input forms.

- ADD

1 - 6	8 - 13
ADD	Starting sequence No. where following cards are to be added.

The "ADD" control card signals the generator to add the following cards to the source deck after the number given in 8-13 in the ADD card. The ADD control card must follow a "TABLE" or "SCAN" control card. Cards to be added must be punched in standard scan or table form - see input forms.

- END

1 - 6
"END"

Signals the generator the final source card for the given assembly (job) has been read.

8.15 PSTG CONTROL CARDS (continued)

o ***END

```
1 - 6

***END
```

Signals the generator that all source jobs for assembly in the stack have been read in.

```
1        8-14       16
$       Program     PSTG
```

First card of deck which calls PSTG into core.

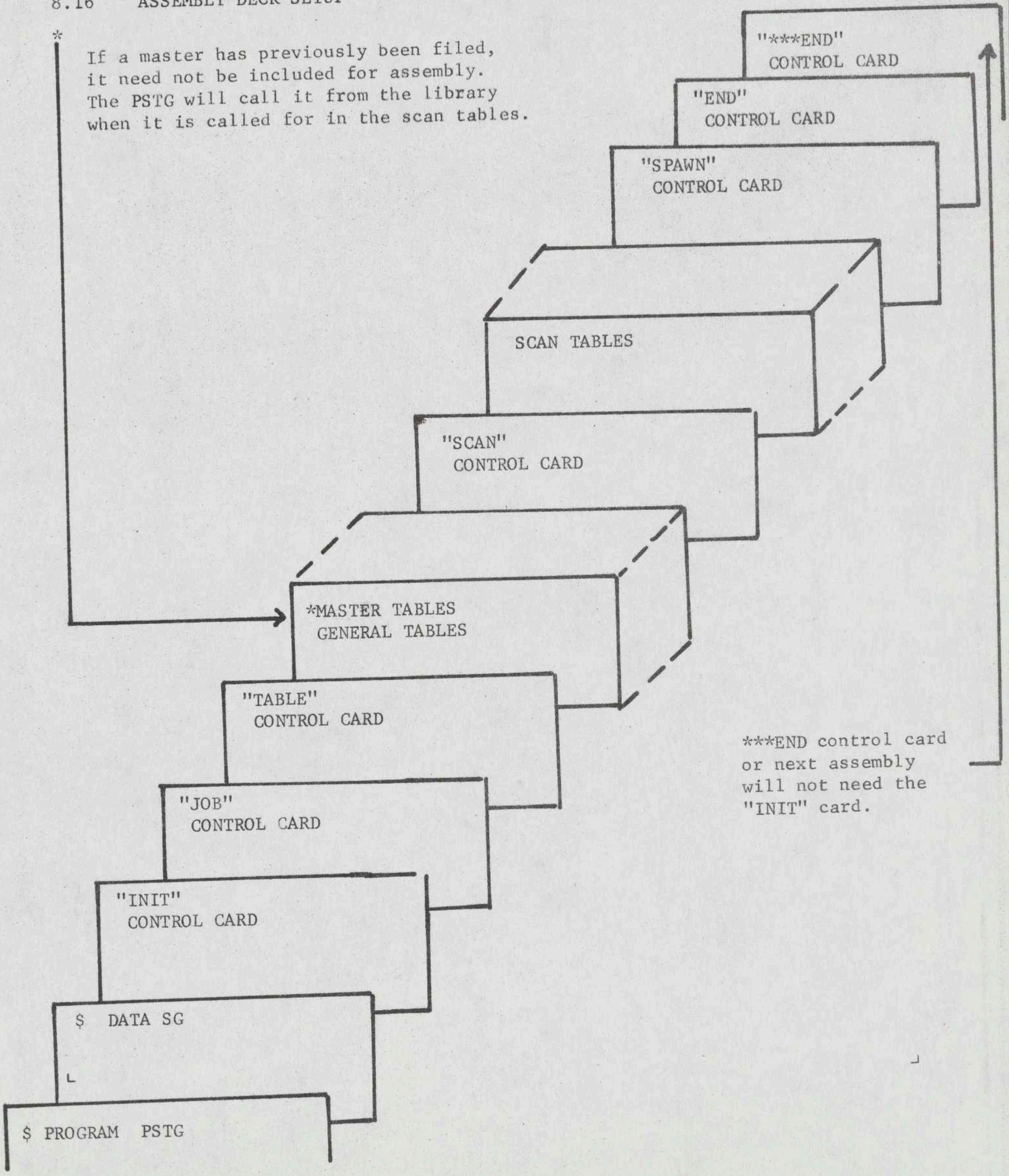
```
1        8-14       16

$        DATA       SG
```

Second card of deck setup preceding PSTG control cards and source program.

Cont. on Page 146

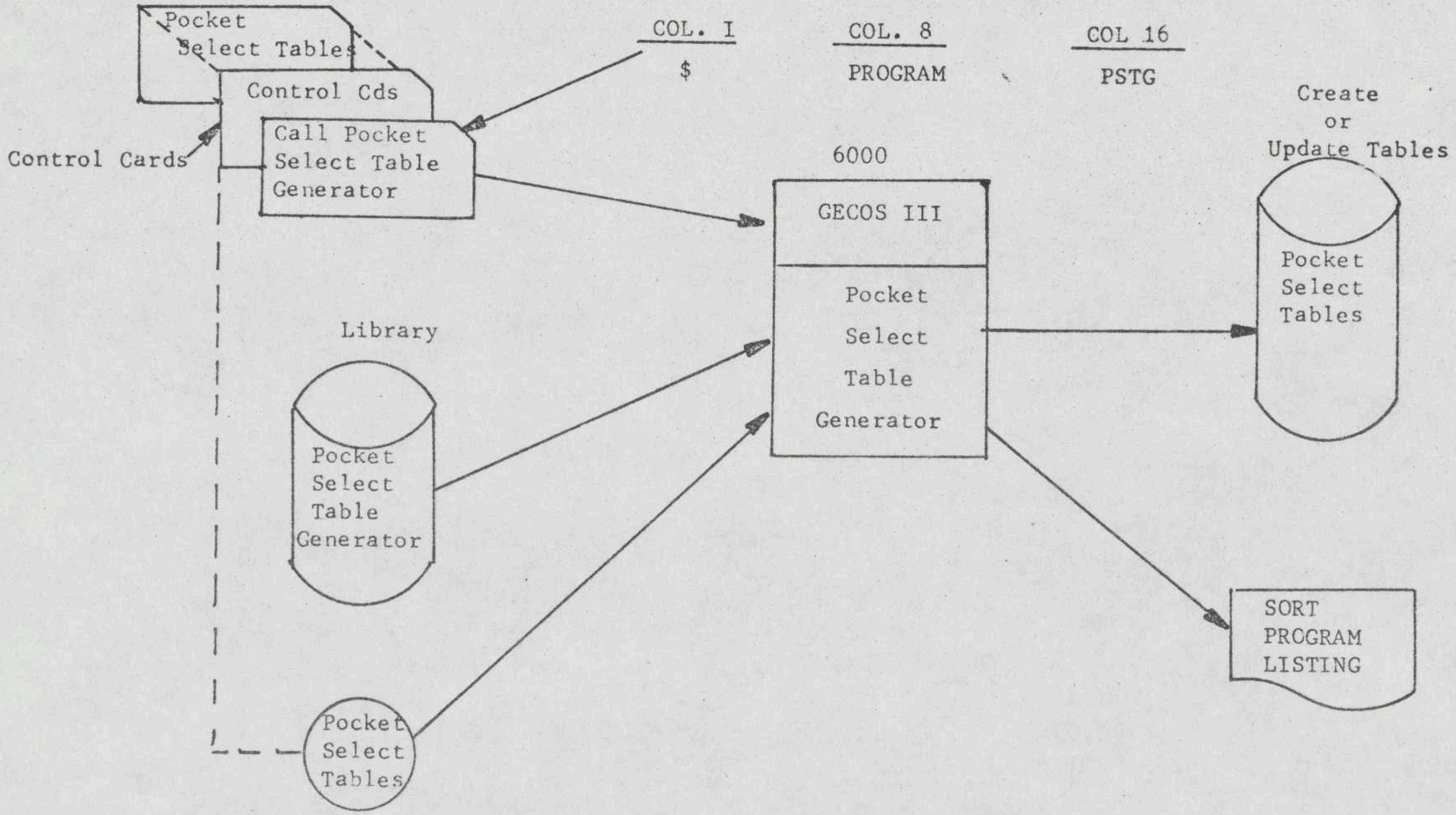
8.16 ASSEMBLY DECK SETUP



*If a master has previously been filed, it need not be included for assembly. The PSTG will call it from the library when it is called for in the scan tables.

***END control card or next assembly will not need the "INIT" card.

8.17 LOADING AND RUNNING THE POCKET SELECT GENERATOR PROGRAM



9.0 355 DOCUMENT HANDLER CONTROL PROGRAM (355 DHCP)

The software for the 355 will be restricted, as much as possible, to the mechanical requirements of driving the MRS200 and the DRD200 document handlers, and performing the necessary logic required to make pocket selections. Therefore, whenever possible, totals, subtotals, and restart procedures, and communications to the TTY's, etc. will be performed by the 6000 DEP. This is necessitated by the shortage of core in the 355 and to some extent the critical timing requirements of the document handlers, particularly when running a maximum of six. The major software requirements that are absolutely necessary for the 355 follows:

1. The 355 document handler control module will utilize the existing 355 GERTS package as much as possible. This will include:
   a. The Fault processor "GSSP".
   b. The main module (executive) which handles dispatches, etc. "GMAN"
   c. The inter-computer module which handles transfer of data and messages between the 355 and the 6000 "GICM".
   d. Whenever there is a High Speed Line Adapter or a Single Line Adapter the TTY's module will be utilized "GTTY" in place of the console module listed under 3.
2. A design of operating six document handlers is required utilizing 32K.
3. The 355 DHCP will require a console module interfacing with the console adapter. The 355 console module will be capable of cummunicating with from one (1) to six (6) 355 console typewriters processing standard GERTS messages formats.
4. One console typewriter will serve as a master console, as well as a control console for a document handler or handlers.
5. The 355 DHCP will initiate, on command from the 6000 DEP, start feeds, and stop feeds on the document handlers.
6. For the MRS200 and DRD200 (2 pockets) document handlers the 355 DHCP will determine which pocket each document is to be selected through table data, and required logic, and issue that select within 48 milliseconds for from 1 - 6 document handlers. Worst case will be six selects in 48 milliseconds.

9.0 355 DOCUMENT HANDLER CONTROL PROGRAM (355 DHCP) continued

7. The pocket that the document was pocketed in will be placed in the record for transmission to the 6000 DEP.

8. The block size will be a standard GERTS block of 320 words (36 bit words) which will contain at least 10 documents.

9. The 355 DHCP will put all hardware status, pertinent to the document handlers, in every document (record) and this will be transmitted to the 6000 DEP as part of the record.

10. The 355 DHCP will be capable of starting, stopping, and restarting the document handlers on request from the 6000 DEP for normal stop - start conditions includes but is not restricted to the following situation:

    a. In the event there is no request for data on a given handler from the 6000 DEP, and its buffers are full in the 355; stop the handler read and pocket the in-transit documents. This will require a three record overflow area or a three record lead time within a normal block.

    b. When the condition under a arrives, and upon receiving and servicing a data request from the 6000 DEP for the stopped handler, resume normal operation without any operator interference.

    c. Whenever this type of stop occurs the proper code will be placed in the record to inform the 6000 DES that the handler had to be stopped, the reason, that the data is correct and to process it.

    d. On a hardware read error, stop the handler, issue a read for the next document in route, reject the document in error, place a status in the record that informs the 6000 DEP slave not to process this document, it must be reread.

    e. After the document or documents in transit following the read error have been read and pocketed, send the block (even if short) to the 6000 DEP and wait for a restart message from the 6000 DEP.

11. Sequence no. each record going from the 355 to the 6000.

12. The 355 DHCP will be capable of handling all standard GERTS transmissions from the 6000 DEP for the teletypes or 355 console typewriters.

9.0 355 DOCUMENT HANDLER CONTROL PROGRAM (355 DHCP) continued

13. The 355 DHCP will not need to keep error threshold counts; this will be done in the 6000 DEP.

14. When the 355 detects an error that requires a restart point (such as a jam); the 355 will stop the handler, send a message to the 6000 DEP informing it a restart is needed for the particular handler and wait for the 6000 DEP perform the restart procedure and issue a restart message to the 355DHCP.

15. When error conditions reach an excessive point (this will be determined in the 6000) and upon request from the 6000 DEP; stop the handler transfer all remaining data in core from the handler to the 6000 DEP slave. Inform the 6000 DEP that the handler has been stopped, and no more data will follow from that handler.

16. It must be capable of running document handlers for other 6000 slaves specifically T&D (OPTS 6000).

17. It must accept, store, and link the pocket select tables to each document handler module for the 6000 DEP, and Test and Diagnostics (OPTS 6000).

18. It must be capable, upon request from the 6000 DEP, of transferring a job from a malfunctioning handler to another handler. This will be done from the 6000 DEP, but the 355 DHCP must receive and store the table, link the table to the proper module, etc. while continuing production on the other handlers.

19. After a terminate read status from a document handler, the next read for that handler must be given immediately because of the timing variation between reads can vary to a minimum of 00.0 milliseconds.

20. The 355 DHCP must operate any mix of DRD 200s and MRS 200s.

21. Beside containing the contents of the document, a record will contain for control purposes the following:

    a. Handler status field pertinent to the document handlers status for each transaction.

    b. The pocket the document was selected in.

    c. A document status field which informs the 600 DEP of various conditions some of which will be:

9.0 355 DOCUMENT HANDLER CONTROL PROGRAM (355 DHCP) continued

1. Normal read and select; processable data.

2. The handler was stopped on this record due to an error. A restart message will be required from the 6000 DEP.

3. No data record, an irrecoverable error has occurred on this handler. It must be restarted from checkpoint from the 6000 DEP.

4. This is the last record that will be transmitted from this handler due to a normal (software recoverable) error stop condition issue restart instructions to the operator and 355 DHCP.

5. The data is processable but the handler was stopped due to no data request from the 6000 DEP. No operator action is required to restart the handler. It requires a MME (data request) from the 6000 DEP and a message to the 355 DHCP to restart the handler.

22. It must determine field length is correct for all fields on a document.

23. It must determine that record (document) length is correct, i.e. is all characters specified were read in.

24. Perform check digit equation and test results.

25. Read and interrogate four modes of encoding, COC5, MICR mark sense and special mark sense.

26. Make various data tests such as:

- dollar amount limit
- blank field detection
- compare various field to user specified limits

27. Be capable of overiding error conditions when user specifies this option in the table data.

28. Check for invalid characters.

29. Include any special reading and tests (if any) for on T&D. This will be for 6000 OPTS if required.

30. Be capable of servicing document handlers for other 6000 slaves in addition to the 6000 DEP. This will definitely be necessary for 6000 OPTS.

9.1 355 SYSTEM CONFIGURATION AND START-UP (Bootload)

The document handlers will be assigned a permanent identification label (ID) ranging in value from 01 to 06 depending on the number of handlers in the system. The ID number will be placed on the document handlers and will be a permanent ID by which the operator communicates to the Document Entry System Handler assignments by TTY and job number. The TTYs may also have a permanent ID. However, it is only required that a unique ID be assigned to each document handler.

The use of permanent IDs for the document handlers will be necessary for two reasons. First, the operator must identify which handler he plans to run a job on and to which TTY the handler will be assigned. Second, the DES must have a positive way by which to verify the operator's selection. The only way this can be accomplished is through the use of a permanent ID and configuring the system at boot (start-up) time utilizing configuration cards.

The configuration cards will assign the permanent ID numbers (document handlers) to their configured channels on the 355. Figure A shows the configuration requirements.

Use of configuration cards will allow channel assignment flexibility as far as the document handlers are concerned. The document handler can be coupled to any channel so long as a corresponding configuration card is provided at boot (start-up) time. As shown in Figure A the configuration card will assign the document handler by ID to its physical channel.

The configuration card will include the following information about the handler and the channel.

a. The handler physical ID number and channel configured to.

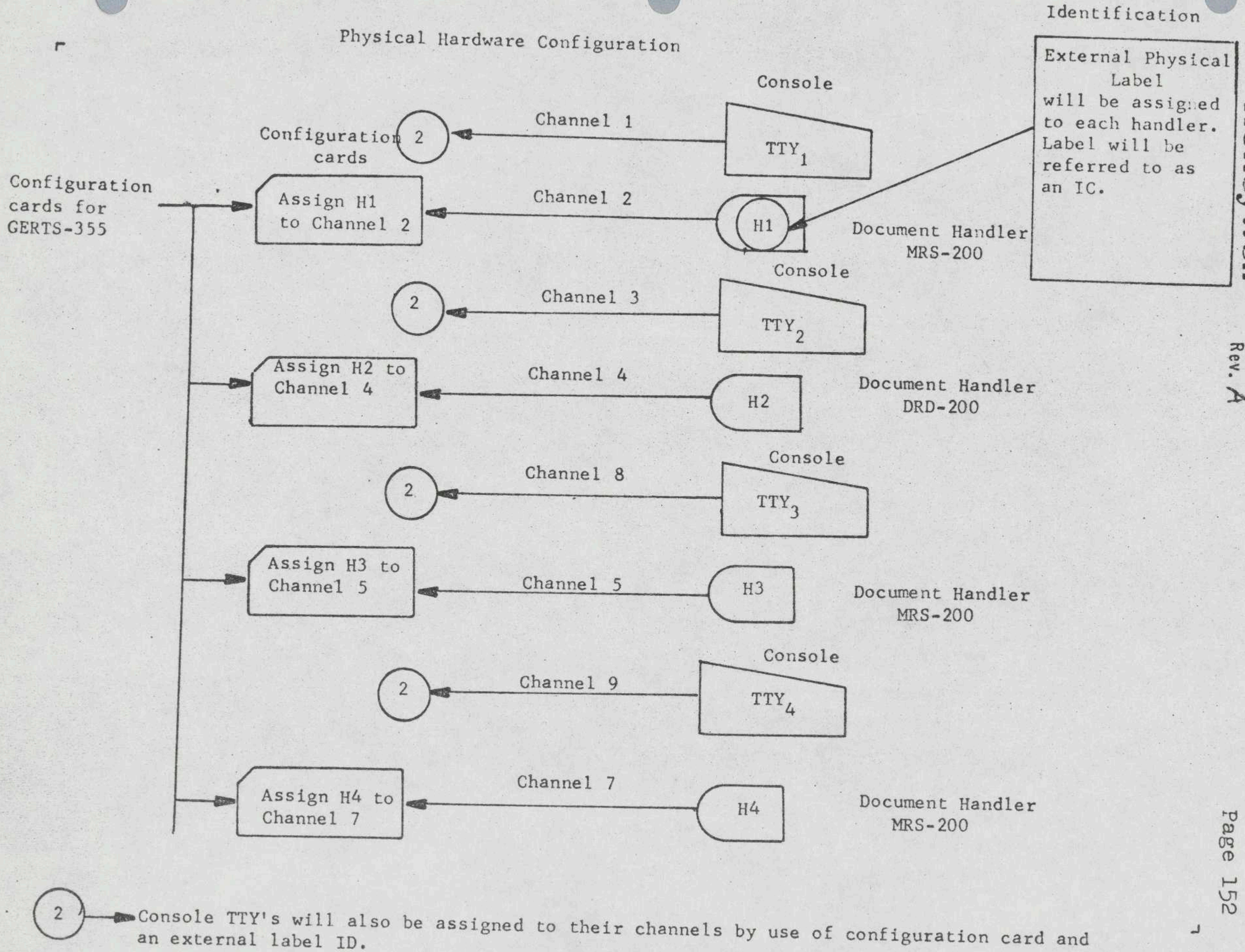
b. The type of device
   1. MRS200
   2. DRD200
   3. B9134*

c. The type of read head the device is equipped with
   1. MICR
   2. COC5
   3. Mark Sense
   4. OCR*

This information must be available to T&D when requested by a T&D program.

* This will be defined in a later EPS update.

Physical Hardware Configuration



Console TTY's will also be assigned to their channels by use of configuration card and an external label ID.

FIGURE A

9.2 SYSTEM CONTROL AREA

The System Control Area will contain all necessary pointers to enable the 355 DHCP to process data coming from six handlers, determine the pocket selection and monitor and control the physical requirements of the pockets on the sorters, etc.

The System Control Area consists of 64 words. It will be updated from the 6000 DEP prior to running each job requested. It consists of the following table pointers and control words.

9.3 SCAN TABLE POINTERS

There are six scan table pointers, one for each document handler, 01 - 06. After a document has read, the document handler ID, 01 - 06, is used to index to the handlers corresponding scan table pointer. The scan table in turn directs the 355 DHCP through the necessary steps required to sort the document read on the handler.

SYSTEM CONTROL AREA

	0 17
0	Not used
1	Pointer to scan tables handler #1
2	" " " " " #2
3.	" " " " " #3
4	" " " " " #4
5	" " " " " #5
6	" " " " " #6
7	Not used
8	Pointer to Pocket Tables Handler #1
9	" " " " " #2
10	" " " ", " #3
11	" " " " " #4
12	" " " " " #5
13	" " " " " #6
14	Not used
15	Switch word for handler #1 bits 8-17
16	" " " " #2 " "
17	" " " " #3 " "
18	" " " " #4 " "
19	" " " " #5 " "
20	" " " " #6 " "
21	
63	Available for other use

SYSTEM CONTROL AREA

Scan Table Pointers

Decimal	
0	US = unassigned
1	Pointer to handler #1 scan tables
2	Pointer to handler #2 scan tables
3	Pointer to handler #3 scan tables
4	Pointer to handler #4 scan tables
5	Pointer to handler #5 scan tables
6	Pointer to handler #6 scan tables
7	UA = unassigned

9.4 POCKET SELECT CONTROL TABLE POINTERS

There are six pocket select control table pointers and six pocket select control tables. One for each handler in the system 01 - 06. The pocket select control table pointers point to each handler, 01 - 06, control tables.

9.5 POCKET SELECT CONTROL TABLES

There are six pocket select control tables one for each handler 01 - 06. However, the number of pocket select tables residing in the 355 at any given time will depend on the number of handlers in use. There will be one table for each handler in use.

These control tables allow the 355 DHCP to monitor and control the number of documents in each pocket, determine when a kill separator is needed, etc. Prior to going to the control tables the 355 DHCP must have selected a pocket for the document.

The pocket control tables may consist of 2 - 32 pocket devices. For each device, each pocket is defined as to number of documents before a kill separator, number of pockets before switching to an alternate pocket, etc.

Cont. on Page 158

Pocket Select Control Table Pointers

Decimal	
8	Pointer to Handler #1 Pocket Select Control Table
9	Pointer to Handler #2 Pocket Select Control Table
10	Pointer to Handler #3 Pocket Select Control Table
11	Pointer to Handler #4 Pocket Select Control Table
12	Pointer to Handler #5 Pocket Select Control Table
13	Pointer to Handler #6 Pocket Select Control Table
14	UA = unassigned

9.6 DHCP SORT VERBS (SCAN TABLES)

Action Verbs provide the user with complete logic flexibility for building sort runs. Validation and pocket selection of documents, including control documents, can be readily processed with the variety of action verbs offered. These verbs, along with the pocket tables and general tables, provide the user an easy to implement sort package which shoild cover nearly any kind of sort requirement. The action verbs by category are as follows:

1. Field/Subfield Identification Verbs

   . GETFIELD
   . GETSUBF
   . VALIDATE *

2. Test/Validation Verbs

   . CKFORMAT
   . CHECKTCD
   . TEST
   . SEARCH
   . DIGITEST
   . VALIDATE *

3. Logic Control Verbs

   . ENDTEST
   . SETSW
   . RESETSW
   . TESTSW
   . TRANSFER

4. Document Handler Control Verbs

   . STARTRUN
   . ENDRUN
   . TYPEOUT
   . HALTDH

5. Special Purpose Verbs

   . MASK
   . CKPOINT
   . SAVEFLD

* Combines GETFIELD and CKFORMAT verbs.

9.6 DHCP SORT VERBS (SCAN TABLES) continued

6. Device and Pocket Control Verbs

   - DEVICE
   - SETVALUE

7. File Control and Job Control Verbs

   - FILECTL
   - *SPAWNJOB

8. SLAVE Program to DEP Communication

* SPAWNJOB is provided through the use of the SPAWN control card see PSTG assembly control cards.

The DHCP will include interpretive routines to execute each one of the given verbs with the exception of those which are for file and spawn control. The interpretive routines will use the given scan records to determine the field and sort characteristics for processing the documents.

The Scan Tables (sort verbs) will contain the test indicators and field identifiers for processing the documents coming from the handlers. Each document handler in use, 01 - 06 will have a set of scan records.

Legion

Letters and numbers below words = corresponding field from users scan input forms.

* = PSTG computed or tested

% = General Table

C = Master Tables

GETFIELD

Test Type 01

word 0

0 5	6 17
Length of Scan Record	Transfer Index
	Z and 2

1

0 17
Control Word
G, I, J, K, L, Z and 2

0 5	6 11	12 17
Test Type 01	Field ID #	Field Location in document IF
E	H and *	Bit 7 of CW = 1

3

0 5	6 11	12 17
Error Code	Accept Pocket	Error Pocket
3	Y	1

4

0 5	6 11	12 17
1st Starting Que	2nd Starting Que or Skip Count	Min. Field Length
I	J	N

5

0 5	6 11	12 17
1st Ending Que	2nd Ending Que or Skip Count	Max. Field Length
K	L	M

6

0	1	2	3 5		6 11	12 17
A	B	C	D		Zero	Scale Count by Digit
P	P	J	L		4	R

GETFIELD

A set to 1 = Scaling required.

A set to 0 = No scaling required.

B set to 1 = Right hand scaling required.

B set to 0 = Left hand scaling required.

C set to 1 = Word 4 bits 6-11 are a queue skip count.

D set to 1 = Word 5 bits 6-11 are a queue skip count.

SCAN records are referenced from the User Input Forms shown on page

See User Section 13.0

VERBS	PAGE
GETFIELD	220
GETSUBF	221
VALIDATE*	222
CKFORMAT	223
CHECKTCD	224
TEST	225
SEARCH	226
DIGITEST	227
VALIDATE*	
ENDTEST	228
SETSW	229
RESETSW	230
TESTSW	231
TRANSFER	232
STARTRUN	233
ENDRUN	234
TYPEOUT	235
HALTDH	236
MASK	237
CKPOINT	238
SVEFLD	239
DEVICE	240
SETVALUE	241
FILECTL	242
*SPAWNJOB	

Legion

Letters and numbers below words = corresponding field from users scan input forms.

* = PSTG computed or tested.

% = General Table

C = Master Tables

GET SUBF

Test Type 03

	0 5	6 17
Word 0	Length of Scan Record	Transfer Index
	*	Z and 2

	0 17
1	Control Word
	G, Z and 2

	0 5	6 11	12 17
2	Test Type 03	Que Field ID Number	Subfield within Que Field ID #
	E	H and *	S and *

	0 5	6 11	12 17
3	Error Code	Accept Pocket	Error Pocket
	3	Y	1

	0 5	6 11	12 17
* 4	Beginning Loc. of Subfield in Que field	No. of digits in subfield	UA
	Q	R	

* Word 4 bits 0-5 contain starting location of subfield within the que field. The starting location within the que field is given left to right with the first position (after the que) equal to zero.

VALIDATE

The VALIDATE verb will consist of two other verbs; GETFIELD and CKFORMAT.

The PSTG will recognize the VALIDATE verb and construct two verbs, the GETFIELD and CKFORMAT, to enable the 355 DHCP to execute the required logic.

The PSTG will generate the GETFIELD verb first followed by the CKFORMAT verb. The error pocket, if one, will be inserted in both verbs. The success pocket if one will be inserted in the CKFORMAT verb only.

If an XFER or success is given it will be placed in the CKFORMAT verb only. An XFER on failure will be placed in both verbs.

Bits 12-15 of the control card will be set as follows for the GETFIELD and the CKFORMAT.

VALIDATE

VALIDATE verb has XRER on success

GETFIELD verb control word

bit 12 = 1
bit 12 = 0
bit 14 = 1 if no ENDTEST after VALIDATE
bit 14 = 0 if ENDTEST follows VALIDATE
bit 15 = 0

CKFORMAT verb control word

bit 12 = 1
bit 13 = 1
bit 14 = 1 if no ENDTEST after VALIDATE
bit 14 = 0 if ENDTEST follows VALIDATE
bit 15 = 0

VALIDATE

VALIDATE verb has XFER on failure

GETFIELD verb control word

bit 12 = 1
bit 13 = 0
bit 14 = 1
bit 15 = 1

CKFORMAT verb control word

bit 12 = 0 if ENDTEST follows VALIDATE
bit 12 = 1 if no ENDTEST follows VALIDATE
bit 13 = 0
bit 14 = 1
bit 15 = 1

VALIDATE

VALIDATE has no XFER

GETFIELD verb control word

bit 12 = 1
bit 13 = 0
bit 14 = 0
bit 15 = 0

CKFORMAT verb control word

bit 12 = 0 if ENDTEST follows VALIDATE
bit 12 = 1 if no ENDTEST follows VALIDATE
bit 13 = 0
bit 14 = 0
bit 15 = 0

CKFORMAT

TEST TYPE - 04

Word 0

	0 5	6 17
Word 0	Length of Scan Record	Transfer Index
	*	Z and 2

	0 17
1	Control Word
	G, Z and 2

	0 5	6 11	12 17
2	Test Type	Field ID #	
	E	H	

3	Error Code	Accept Pocket	Error Pocket
	3	Y	1

	4	MAX digit count before a dash +or-	MIN digit count before a dash +n-	D	E	F	G	H	I
		T	T	*	G	G	U		
*Y	5	MAX digit count after a dash + or -	MIN digit count after a dash +or-	E	E	F	G	H	I
		T	T	*	G	G	U		

D = Set to 1 end of Format Test

e = 1 = A dash IS expected after MIN or MAX digit count

F = 1 = a + or - is expected after MIN or MAX digit count

e and F = 0 = NO dash or + or - expected after MIN or MAX count. In this case the count of digits & no cash is acceptable.

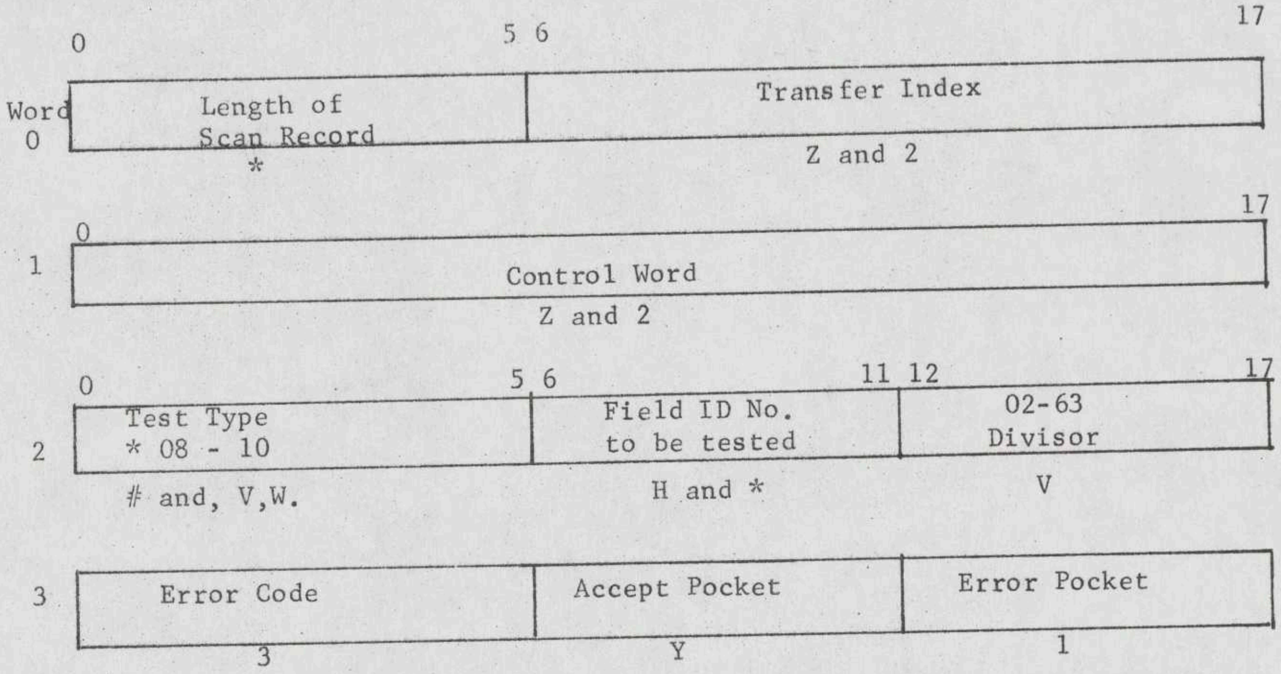
G = 1 = accept a non-numeric in place of a dash (error signal for MRS 200, Oct. 13)

* Y repeated for each dash, + or -.

I = UA. H = UA

CHECKTCD

* TEST TYPES 08, 09, 10



* Test Type 08 = Alternate 2's TCD
Test Type 09 = Natural TCD
Test Type 10 = Mod 10 Geometric TCD

The Pocket Select Generator will determine the TCD type from the users input form and give the 355 DHCP the proper verb type including the divisor.

The 355 DHCP will handle these TCD types through standard TCD routines, per, TCD section of the EPS.

CHECKTCD (SPECIAL)

Test Type 11

Word 0	0–5	Length of Scan record	*
	6–17	Transfer Index	Z and 2
1	0–17	Control Word	Z and 2
2	0–5	Test Type	E, and V, W.
	6–11	Field ID No. To be tested	H and *
	12–17	UA	
3		Error code	3
		Accept Pocket	Y
		Error Pocket	1
4		Pointer to TCD Table	4, 5 and %

The user must supply the necessary TCD divisor remainder and multipliers. The 355 DHCP program will do a table load-up, multiplying each in the field by the multiplier in the table, divide this result by the given divisor and check the given remainder.

Legion

Letters and numbers below words = corresponding field from users scan input forms.

* = PSTG computed or tested.
% = General Table
C = Master Tables

TEST
TEST TYPE 05

Word	0 – 5	6 – 17
0	Length of Scan Record	Transfer Index
	*	Z and 2

Word	0 – 17
1	Control Word
	C, %, Z and 2, X.

Word	0 – 5	6 – 11	12 – 17
*2	Test Type 05	Field ID #	Bit 8 = 1 Field ID # at 2nd Compare Field
	E	H and *	5 and *

Word			
3	Error Code	Accept Pocket	Error Pocket
	3	Y	1

Word	
**4	Pointer to the Table or Constant
	4, 5*, % or @

* Bit 8 control will be set to 1 if a field to field compare is required and 12-17 of word 2 will contain the 2nd compare field.

** Word four (4) will contain the pointer to the table or constant if bit 8 of control word = 0. If bit 8 is set to 1 word 4 will be set to zero. The 355 DHCP will set a pointer from the field ID found in the document to locate the same.

Legion

Letters and numbers below words = corresponding field from users scan input forms.

* = PSTG computed or tested.

% = General Table

@ = Master Tables

SEARCH

Test Type 07

Word 0	0–5	Length of Scan Record	*
	6–17	Transfer Index	Z and 2
1	0–17	Control Word	@, Z and 2 and X
2	0–5	Test type 07	E
	6–11	Field ID #	H
	12–17	Pocket Field 0, 1 or 2	Dep. at load time, zero.
3	0–5	Error Code	3
	6–11	Accept Pocket	Y
	12–17	Error Pocket	1
4	0–17	Pointer to Table	%, @ and *, 4 and 5

Field Segment Control

** 5	0–5	A	@
	6–11	B	@
	12–17	C	@

*Bits 4, and 5 of the control are used as follows:

SEARCH
Test type 07

Bits	Use
4	Set to 0 = BCD Table Set to 1 - Bin Table
5	Set to 0 = Segmented Table Set to 1 = Non-segmented Table

** A, B, and C contain number of digits in each segment of the field and table

Legion

Letters and numbers below words = corresponding field from users scan input forms.

* = PSTG computed or tested.

% = General Table

C = Master Tables

Digit Test

Test Type 06

	0 5	6 17
Word 0	Length of Scan Record	Transfer Index
	*	Z and 2

	0 17
*1	Control Word
	z, 2 and X

	0 5	6 11	12 17
2	Test Type 06	Field ID #	Location in Field at Digit (L-R)
	E	H and *	Q

	0 5	6 11	12 17
3	Error Code	Accept Pocket	Error Pocket
	3	Y	1

	0 5	6 11	12 17
4	UA	UA	A Digit Value

ENDTEST

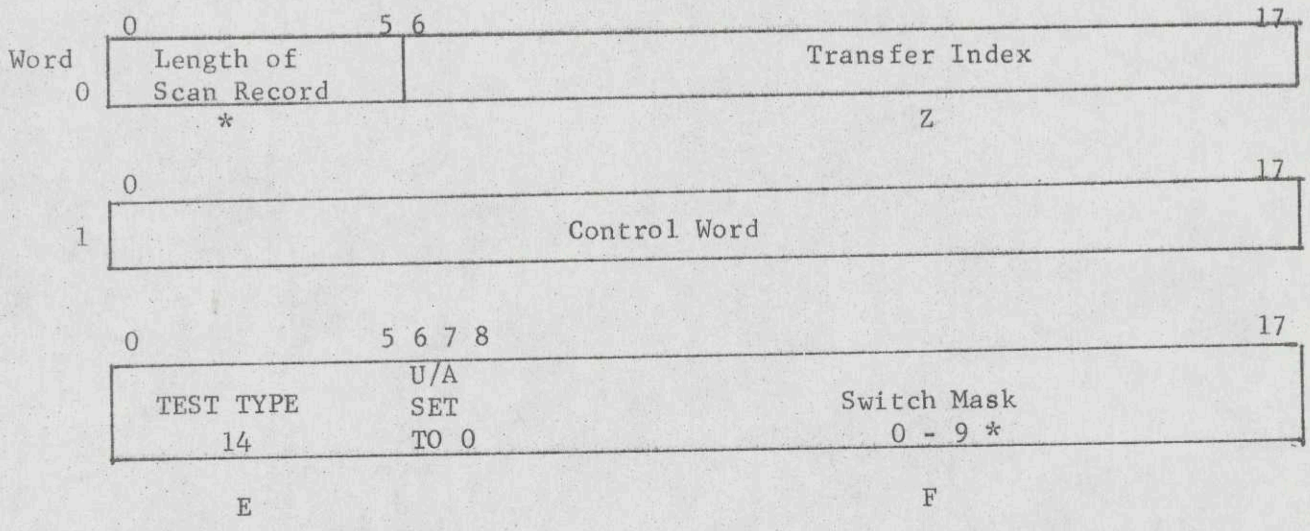
Test type 12

0	Length of Scan Record	UA
1	bits 12, 13, 14, 15 = 0	
2	Test type = 12	UA

The ENDTEST verb signals the DHCP that all testing has been (for a document) completed and to pocket the document.

SETSW

TEST TYPE 14



* The PSTG will convert the users switch settings 0-9 to 8-17.

The 355 DHCP will use 8-17 as a mask to set corresponding bit or bits in the switch word in the system control area. There will be a switch word in the system control area for each handler 1-6.

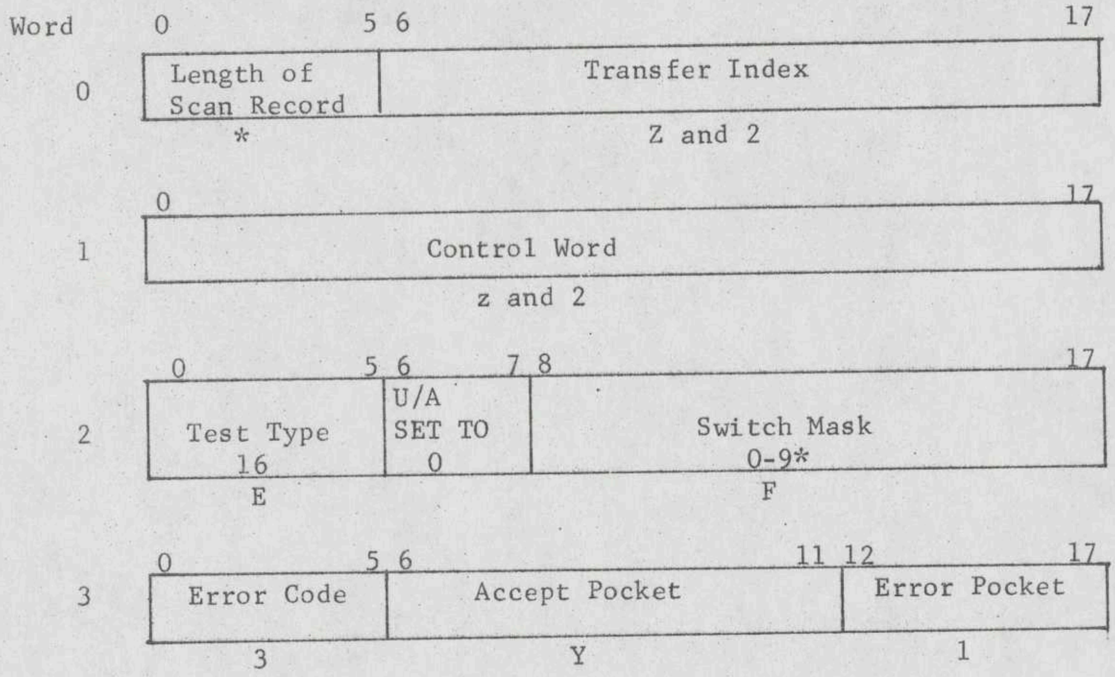
RESETSW

TEST TYPE 15

Word	0 5	6 7	8 17
0	Length of Scan Record		Transfer Index
	*		Z
	0 17		
1	Control Word		
	Z		
	0 5	6 7	8 17
	Test Type 15	U/A SET TO 0	Switch Mask 0-9*
	E		F

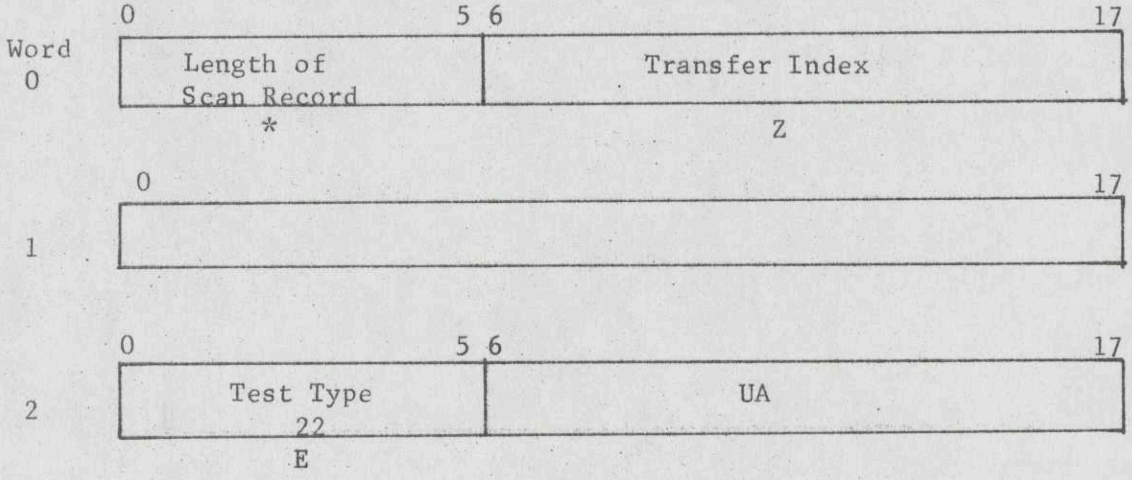
* The PSTG will convert the users switch settings 0-9 to 8-17. The 355 DHCP will use 8-17 as a mask to set the corresponding bit or bits in the switch word in the system Control area off (set to 0). There will be a switch word in the system control area for each handler 1-6.

TESTSW
TEST TYPE 16



* The PSTG will convert the users switch settings 0-9 to 8-17. The 355 DHCP will use 8-17 as a mask to test the corresponding bit or bits in the switch word in the system control area. There will be one switch word in the System Control area for each handler 1-6. The switch is considered to be on (Set to 1) when both corresponding bits are set to 1.

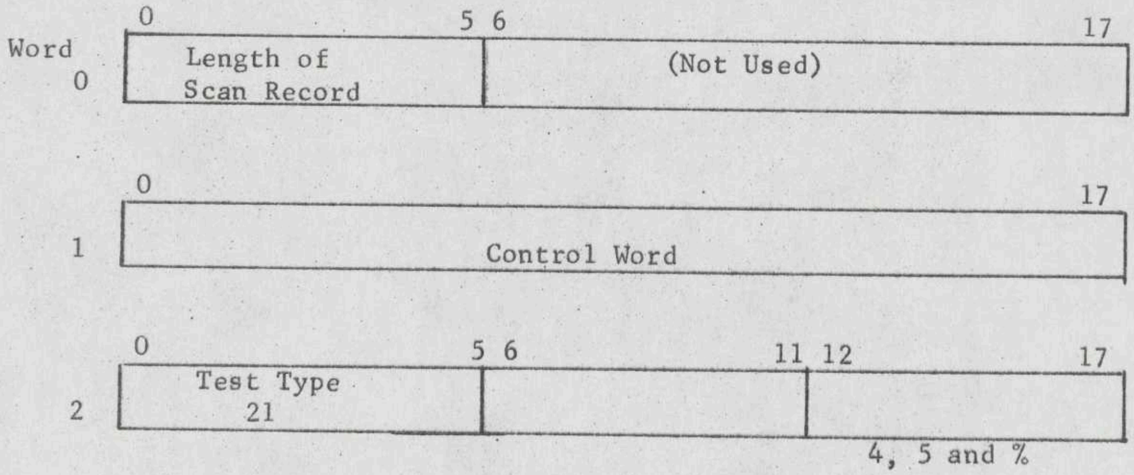
TRANSFER
TEST TYPE 22



This verb will cause the 355 DHCP to execute an unconditional transfer to the location given in word 0, bits 6-17.

Note: If a verb, (other than the Transfer verb) has both success and failure transfer specified then a Transfer verb will be generated for the success transfer by the PSTG.

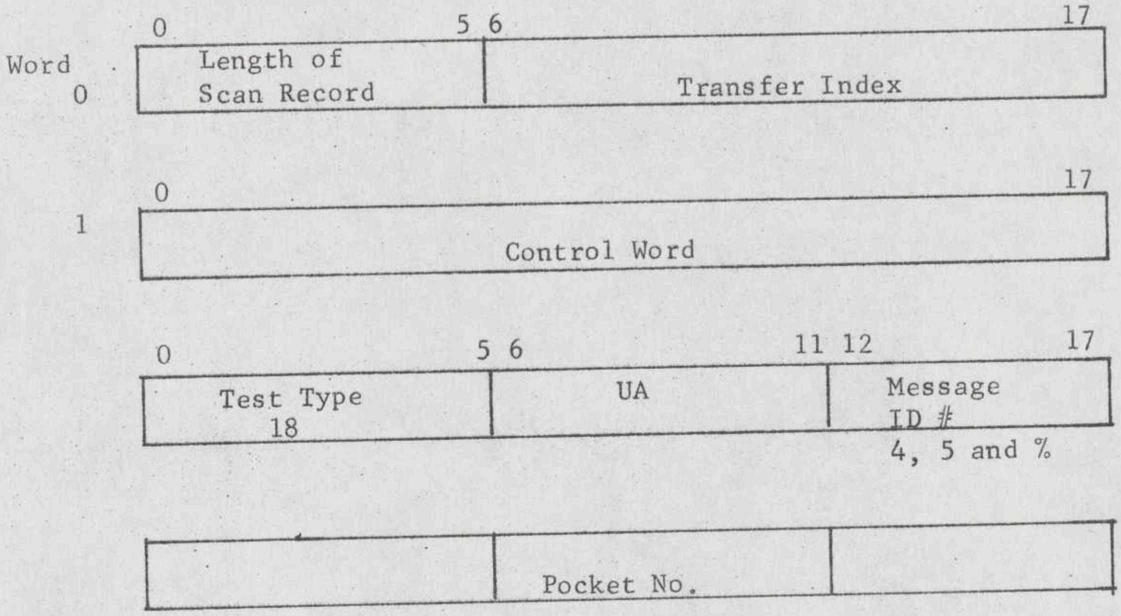
ENDRUN
TEST TYPE 21



When the ENDRUN verb is detected in the 355 DHCP, the test verb will be stored in the document status word (word 6-bits 0-5) and the document handler stopped. All following documents will be rejected (after the stop).

TYPEOUT

Test Type 18



On a "typeout" the 3tt DHCP will place the test type (18) in the comm status word (word 7 bits 0-5) and the message ID no. in the document status word no. 7, bits 0-5.

HALT DH

Test Type 17

	0 5	6 17
Word 0	Length of scan record	Transfer Index
	*	Z

	0 17
1	Control Word
	Z and Y

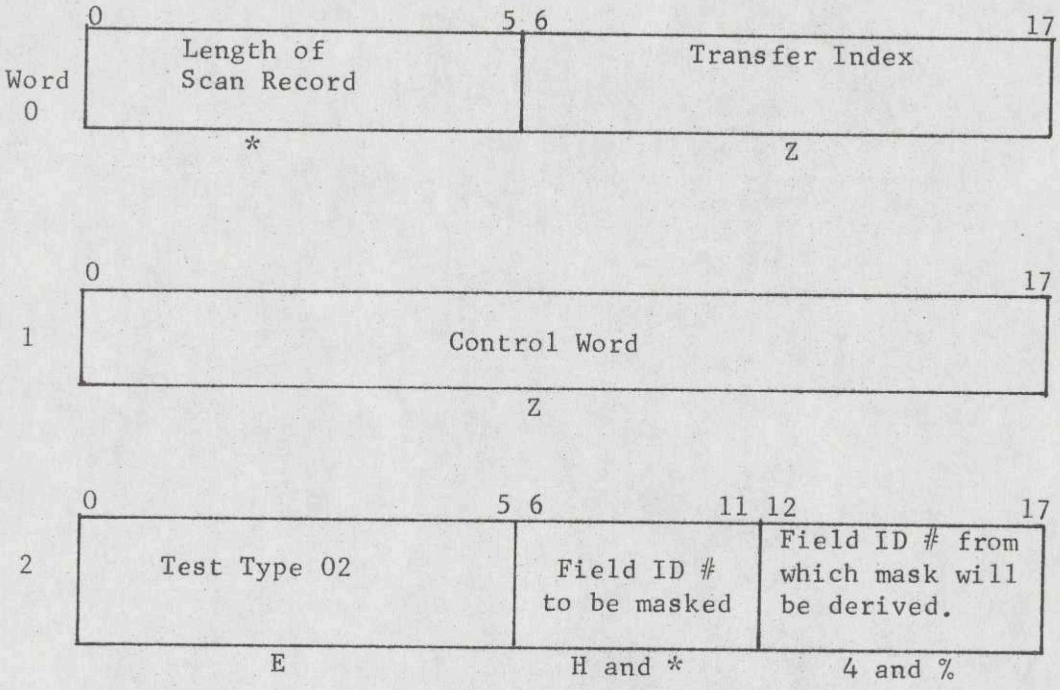
0 5	6 11	12 17
Test Type 17	UA	Message ID #
E		4 & 5 and %

	0 5	6 11	12 17
3		Accept Pocket	

On a HALTDH the 355 DHCP will issue a controlled stop to the document handler, put the document in the given pocket word (word 3 bits 6-11), and place the type (17) in the comm status word (word 7 bits 0-5) and the message ID# in word 7, bits 0-5, of the document status word. All documents which follow the HALTDH will be sent to the pocket no. given in word 3 bits 6-11 or the last pocket received for the document.

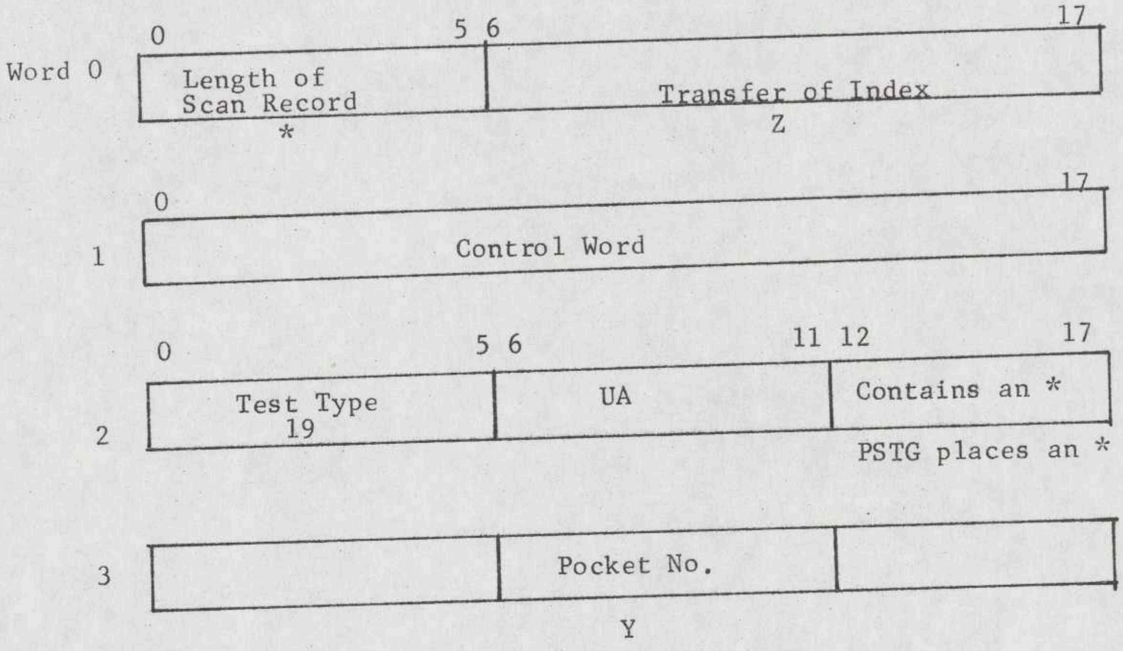
Legion
Letters and numbers below words = corresponding field from users scan input forms.
* = PSTG computed or tested.
% = General Table
C = Master Tables

MASK
Test Type 02



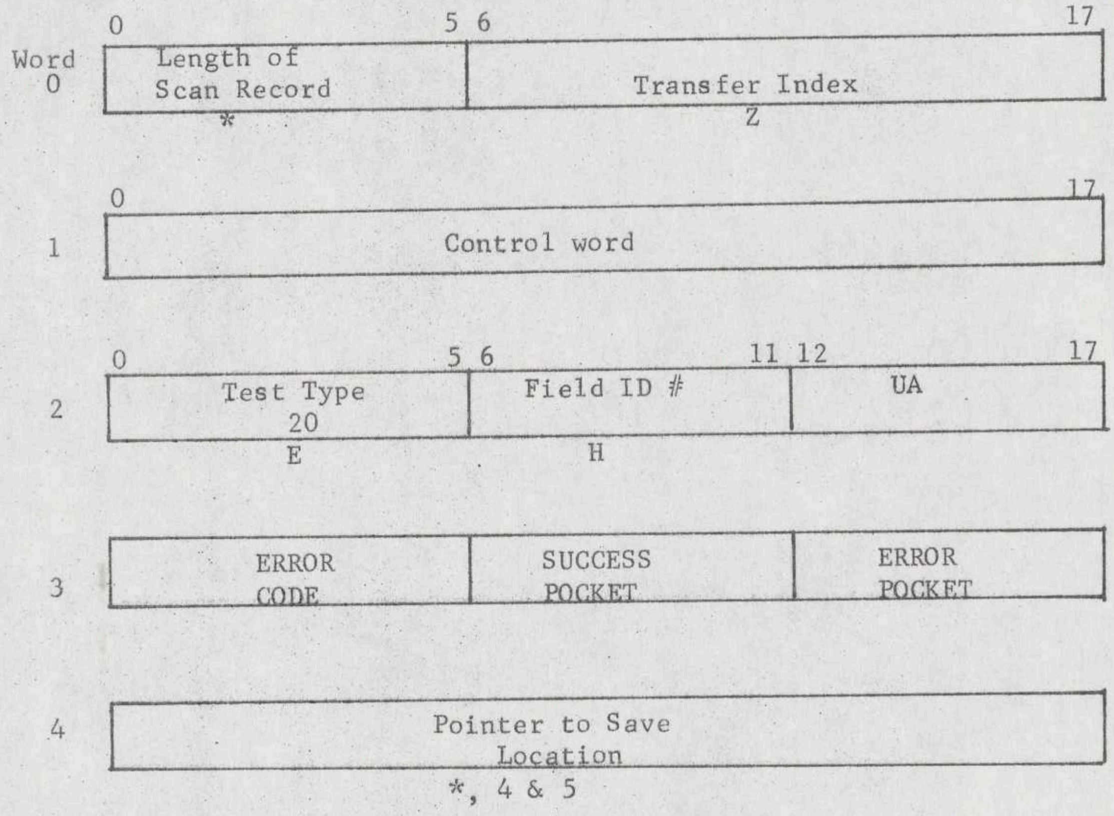
The mask verb is used to zero non-significant high order positions in a field. The field given in word 2 bits 6-11 is the field to be masked. The field given in word 2 bits 12-17 is the field from which the mask will be derived. The high order position of the field given in word 2 bits 12-17 will be searched until a non-zero digit is found. The no. leading zeroes found must be applied to the same high order positions of the field given in word 2 bits 6-11.

CKPOINT
Test Type 19



The 355 DHCP will place the verb type in the status word (word 6 bits 0-5).

SAVEFIELD
TEST TYPE 20



Honeywell/GE Engineering Product Specifications, Vol. 23
Nos. 43A177862-43A177886 (3 of 3)

DEVICE

The DEVICE verb does not require a scan record. It will be used to identify the device type (sorter type) 6000 PSTG that the sort job is programmed for. The PSTG will use this information for diagnostic purposes. At the present time there are two device types:

MRS200 - 12 pockets
DRD200 - 2 pockets
*H236 - up to 32 pockets

* Not included in present system

9.7 SETVALUE AND POCKET SELECT CONTROL TABLES

The SETVALUE verb will not require a scan record. It will be used by the user to set up pocket control information about control documents and pocket item counts. the PSTG will use this information to construct the pocket select control tables.

The SETVALUE verb will be used in conjunction with three pocket control codes: MI-M3 will be for defining multi pocket separators, I1-I3 will be used for a kill type separator to be pocketed depending on the I pockets item count, and C which will be used to control the count of items that all pockets will contain before a full pocket condition is reached.

The 355 DHCP will, before pocketing any document, determine if the document is an M, or I type document by testing the accept pocket value for a value of octal 50-55. An octal 50-52 will equal an I type document and an octal 53-55 will equal an M type document.

the SETVALUE used with a "C" control code sets the full pocket count in word 1 (by the PSTG) for each pocket. The 355 DHCP will keep a program count for each pocket, using word 11, and compare this count to the full pocket count. When the actual count exceeds the full pocket count the 355 DHCP will set bit 1 of word 0 on for that pocket, turn on the full pocket light and stop the handler. The reason code will be put in the documents status word to inform the 6000 DEP the reason for the stop.

When the operator on the device empties the full pocket, the full pocket light will turn off and depressing the operate button will restart the handler. The 355 DHCP will always test bit 1 whenever a full pocket is indicated by count and, if on, interrogate the full pocket status bit of the devices status word. If the status bit is no longer on bit 14 will be set to zero and the program count for the pocket, word 11, set to one and normal processing resumed.

```
                                  SET VALUE
        ACTION VERB
SCAN FORM ——→ 10 - 17             18 ————————

        SETVALUE                  C (00-11,300)
```

$$\underbrace{\text{SETVALUE} \qquad\qquad \text{C (00-11,300)}}$$

The PSTG will set the full pocket count at 300 for all pockets. The count of 300 will be put in word 1 of the pocket select control table for all pockets.

9.7 SETVALUE AND POCKET SELECT CONTROL TABLES (continued)

ACTION VERB

SCAN FORM → 10 - 17 18————

SETVALUE I1(02,200.220)(03,200,300)

The PSTG will put the minimum count given in word 1 and the maximum count in word 4 of pocket select control table for pockets two and three. Bit 12 if word 0 will be set to 1 for pockets two and three on the pocket select control table. The PSTG will make I1=50, I2=51, I3=52.

When an M type document is presented to the 355 DHCP for pocketing, the 355 DHCP will:

a. Determine if it is the first M type document received in the series.

b. If it is the first M type document in the series, set up to ensure that the following documents are the same M type until all pockets in use for the device have received one.

ACTION VERB

SCAN FORM → 10 - 17 18————

SETVALUE M1(00-11)

PSTG will set bit 12 and 17 of word 0 of the pocket select table to 1 for all pockets 0 - 11. In the event of a 32 pocket device and the M(00-31) is used, then all pockets 00-31 will be indicated as being in use by setting bit 12 & 17 on for each pocket. PSTG will make M1 = 53.

ACTION VERB

SCAN FORM → 10 - 17 18————

SETVALUE M2(00,6,7)

PSTG will set bit 12 and 17 of word 0 of the pocket select table to 1 for pockets 0, 6 and 7. PSTG will make M2z54.

9.7 SETVALUE AND POCKET SELECT CONTROL TABLES (continued)

PSTG will make M3 = 55 and will set bit 14 on word 0 of the pocket select table.

```
        ACTION VERB

SCAN FORM ——> 10  -  17        18——————

        SETVALUE               C(00-11,300)
```

The PSTG will set the full pocket count at 300 for all pockets. The count of 300 will be put in word 1 of the pocket select control table for all pockets.

```
        ACTION VERB

SCAN FORM ——> 10  -  17        18——————

        SETVALUE               I1(02,200.220)(03,200,300)
```

The PSTG will put the minimum count given in word 1 and the maximum count in word 4 of pocket select control table for pockets two and three. Bit 12 of word 0 will be set to 1 for pockets two and three in the pocket select control table. The PSTG will make I1=50, I2+51, I3=52.

When an M type document is presented to the 355 DHCP for pocketing, the 355 will:

a. Determine if it is the first M type document received in the series.

b. If it is the first M type document in the series, set up to ensure that the following documents are the same M type until all pockets in use for the device have received one.

c. If it is not the first in a series of M type documents place it in the next pocket in use which requires one, or if there are no more pockets requiring one reject it.

d. When the first non-M type document is received following a series of M type documents, ensure each pocket in use received an M type document. If each pocket in use for the device has received an M type document resume normal

9.7 SETVALUE AND POCKET SELECT CONTROL TABLES (continued)

processing until the next M type document is received repeat steps a--d over. If all pockets did not receive an M type document in the series issue a controlled stop, reject documents in transit--do not process these must be reread, and place the M type value in the document status communicator word (word 7 bits 0-55), octal 53, 54 or 55. This will inform the 6000 DEP a type out is required and what message to type.

When an I type document is presented to the DHCP for pocketing the 355 DHCP will:

a. Determine which pocket requires an I type separator by testing bits 12--14 of word zero of the pocket select table.

b. If the I type document is required for the pocket, test the min and max counts in the pocket table against the program count.

If the count is within the min and max range put the document in the pocket and reset the program count for the I separator to 0. Also set the appropriate bit for the I separator received in word 7 bits 12--17 of the document's status words.

c. In the event no pocket requires the I document, it will be rejected.

d. When an I type pocket exceeds the max. count allowed and has not received an I type document; the 355 DHCP will pocket the item document and stop the handler (controlled stop). An octal 50 will be put in the documents status word bits 0--5.

e. When an I pocket has more items than the maximum count allows step d will be repeated until an I document is received it will be pocketed, the count in word 4 reset to 0 and normal processing resumed.

Pocket Select Control Tables
format and usage

	0 5	6	7 11	12	13	14	15	16	17
0	Primary Pocket No.	X	Alternate Pocket No.	a	b	c	d	e	f

1 MIN count of items before receiving an I1 type separator

2 MIN count of items before receiving an I2 type separator

3 Min count of items before receiving an I3 type separator

4 Max. count of items before receiving an I1 type separator

5 Max. count if items before receiving an I2 type separator

6 Max. count of items before receiving an I3 type separator

Pocket Select Control Table
Format and Usage

7	Number of items counted by program since receiving an I1 separator
8	Number of items counted by program since receiving an I2 separator
9	Number of items counted by program since receiving an I3 separator
10	Number of items to be put in pocket before turning on full light PKT
11	Program count of number of items in pocket since turning off full light PKT

a=1=I1 separators in use for this pocket
b=1=I2 separators in use for this pocket
c=1=I3 separators in use for this pocket
d=1= pocket not in use
d=0= pocket in use
X=1= No alternate pocket in use
X=0= alternate pocket in use
f= To be used by the 355 DHCP SET to 1 when pocket full. It will be set back to zero when full pocket light turned off.
e= Pocket will require M type separator.

9.8 SCAN RECORD

Control Word Use

BITS	USE
0	Set to 1 = Format the field in the outlet location given in word 6 bits 5 - 11 of the GETFIELD verb set to 0 = no formatting.
1-3	100 = Numeric field 010 = Numeric field with dashes 001 = Numeric field which can contain plus, minus, or dash symbols. 110 reserved for 101 later use. These 111 will be used for alpha and alpha-numeric fields.
4	Bit 4 will indicate table type, BCD, or BIN, for the TEST and SEARCH Verbs. 0 = BCD 1 = BIN
5	Bit 5 will indicate if a table is segmented or non-segmented. It will be used in the TEST and SEARCH verbs. 0 = non-segmented 1 = segmented table
6	Unassigned.
7	Set to 1 = no queues delimit field. The field will be located by the starting location given in word 2, bits 12-17, of the GETFIELD verb. Set to 0 = ques delimit the field.
8	This bit is used in conjunction with the TEST verb. It will be set to 1 if the test to be made is between fields rather than a field to a constant or table. Set to 0 = The field is to be tested against a constant or table. Set to 1 = The field whose ID # is given in word 2, bits 6-11, of the TEST verb is to be compared against the field whose ID # appears in 12-17.

9.8 SCAN RECORD (continued)

Control Word Use

Bits	Use
9-11	These bits will be used to indicate the test condition wanted - i.e. - Equal, less than, greater than, or a range. 000 = Equal 100 = Less than 001 = Greater than 010 = Unequal 101 = Range condition Field 1st factor Field 2nd factor
12	This bit will be used to control success scan logic. Set to 0 = Stop scan logic on a successful test and pocket the document. Set to 1 = Continue testing if the test was successful.
13	This bit will be used in conjunction with bit 12. It will indicate whether to transfer on a successful test or continue sequentially to the next sequential scan record. Set to 0 = No transfer continue to next sequential scan record. Set to 1 = Transfer to the scan record whose location is given in word 0 bits 6=17 of the scan record.
14	This bit will be tried to control unsuccessful (failure) scan logic. Set to 0 = Stop and pocket the document if the test has failed. Set to 1 - Continue to another scan record if the test has failed.
15	This bit is used in conjunction with bit 14. It will indicate a transfer on failure or continue to the next sequential scan record. Set to 0 = Continue to the next sequential scan record. Set to 1 = Transfer to the scan record whose location is given in word 0, bits 6--17 of the scan record.
16-17	Unassigned.

9.9 SEGMENTED TABLES

The need to use segmented tables arises for two reasons:

1. To conserve space in the 355.
2. To conveniently handle generic numbers, specifically the FRD-ABA numbers.

Generic numbers are reference numbers, such as the FRD-ABA numbers. (Federal Reserve, District and American Banking Association number) which defines a bank within a city, state area within a Federal District. For example, the FRD-ABA number 1210-1001 defines a bank (1001) assigned to the San Francisco, Calif. area (10) within the twelfth Federal Reserve District (12).

Banks within the Federal Reserve system are sometimes interested in only the Federal Reserve District and/or the City State part of the FRD-ABA number. Still other cases, the entire number is required in their processing of documents. Therefore, by segmenting the table all of the possibilities are covered.

In order to conserve space in the 355 the FRD-ABA table will be carried in binary. The 355 DHCP can easily handle the binary conversion by setting up a BCD to binary conversion table in the following manner.

Units Position	-	no conversion needed
10 position	10 -	Table of Binary
	20 -	Equivalence
	30 -	
	40 -	
	50 -	
	60 -	
	70 -	
	80 -	
	90 -	
100 position	100 -	Table of Binary
	200 -	Equivalence
	↓	
	900 -	
1000 position	1000 -	Table of Binary
	↓	Equivalence
	9000 -	

9.10 FRD ABA TABLE (segmented)

Segment #1 FRD - DIST codes in binary

*Word 0 control word for segment

0 … 11	12	13 … 17
**No. of factors in this segment Max.=4095	0	Length of each factor in words

* each segment will begin with a control word
** factor is equal to words 1, 2, and 3

Word 1 FRD-DIST codes

0 … 17
FRD-DIST Code in Binary 2 digits - 99 always numeric

Word 2 - Pocket nos. DRD206 or 32 Pocket Sorter

0 … 5	6 … 11	12 … 17
Pocket # - Run 1	Pocket # - Run 2	Pocket # - Run 3

NOTE: words 1, 2, and 3 repeat for each unique or new FRD-DIST code. Table will be in ascending sequence.

NOTE: When DRD206 or 32 pocket device used, the pocket # fields will contain all bits to indicate the search is to continue.

Word 3

0 … 14	15 … 17
Pointer to next segment, unless no following segment for this segment then = 77777	MBZ ***

9.11

FRD - ABA Table (segmented) Segment #2 FRD-DIST and city state codes in binary

	0 11	12	13 17
*Word 0 control word for segment	**No. of factors in this segment Max.=4095	0	Length of each factor in words

* each segment will begin with a control word

** factor is equal to words 1, 2, and 3

	0 17
Word 1 DIST and City State Codes	DIST Code and city state code in binary Max. value is 9999 BCD 99 DIST + 99 city state

	0 5	6 11	12 17
Word 2-Pocket nos. for DRD206 or 32 Pocket Sorter	Pocket#- Run 1	Pocket # - Run 2	Pocket # - Run 3

NOTE: See note previous page.

	0 14	15 17
Word 3	Pointer to next segment, unless no following segment for this segment then = 77777	MBZ ***

9.12

FRD - ABA Table (segmented)
Segment #3 ABA codes within in city
(Final Segment) state and DIST

*Word 0
Control
Word for
segment

0 - 11	12	13 - 17
** No. of Factors in the segment Max.= 4095	1	Length of each factor in words

* each segment will begin with control word

** factor is equal to words 1 and 2

When bit 12 = 1 no pointer will be necessary

Set to 1 = Last Segment of table

Word 1
ABA Code

0	1 - 17
	ABA code in binary, max.value = 9999 BCD

on us bit when = 1 represents on us ABA no.
= 0 ABA no. of transit bank

Word 2 - If
DRD206 or
32 Pocket
Sorter

0 - 5	6 - 11	12 - 17
Pocket # Run 1	Pocket # Run 2	Pocket # Run 3

NOTE: see note previous page
*** must be zero

9.13 NON-SEGMENTED TABLES

Non-segmented tables will be used in all cases where generic numbers do not apply. These may or may not be in binary depending on the amount of space necessary to store them. However, the Pocket Select Table Generator will determine these cases and the appropriate bit will be set in the tables control word to indicate to the 355 DHCP that the table is in binary rather than BCD.

FORMAT FOR NON-SEGMENTED TABLES

	0–11	12	13–17
Word 0 - control word for table	No. of factors on this table	0	Length of each factor in words

	0–17
Word 1	* Argument in either binary or BCD

This may be 1, 2, 3, or more words longer.
This will be determined by field size in words.

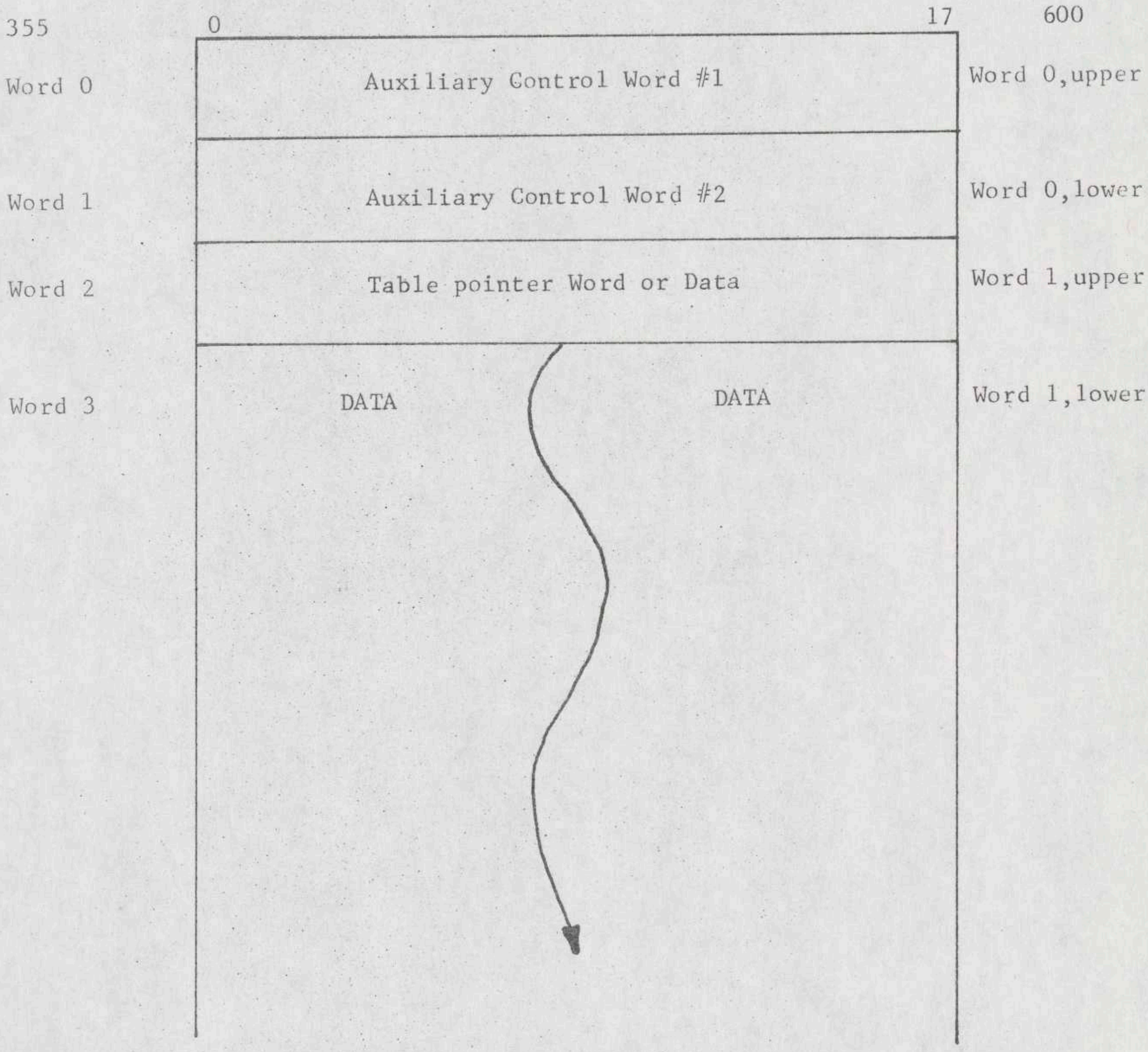
	0–5	6–11	12–17
*Word N-If DRD206 or 32 pockets	Pocket # Run 1	Pocket # Run 2	Pocket # Run 3

9.14 INITIALIZATION OF JOBS IN THE 355

The interface between the 6000 DEP and the 355 DHCP for setting up a job on a handler will be handled through two auxiliary control words, ACW#1 and ACW#2. These control words will inform the 355 DHCP the type of tables and pointers involved and where to place the data within the system control area and/or 355 memory relative to zero. The 355 DHCP will always update the pointers by adding the base address of the system control area.

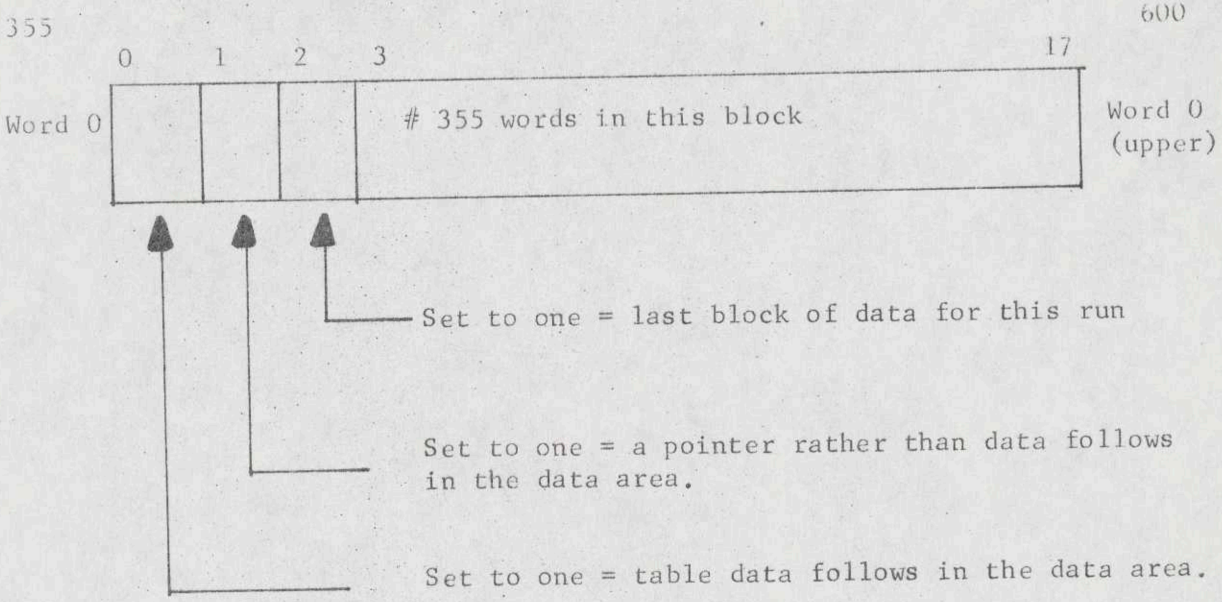
The messages to the 355 DHCP will be standard GERTS format; however, the 6000 DEP will ensure the first two words of the text as control words, ACW#1 and ACW#2 so that 355 DHCP can interpret the data.

INITIALIZATION TABLE FORMAT



355	0 — 17	600
Word 0	Auxiliary Control Word #1	Word 0,upper
Word 1	Auxiliary Control Word #2	Word 0,lower
Word 2	Table pointer Word or Data	Word 1,upper
Word 3	DATA DATA	Word 1,lower

AUXILIARY CONTROL WORD #1

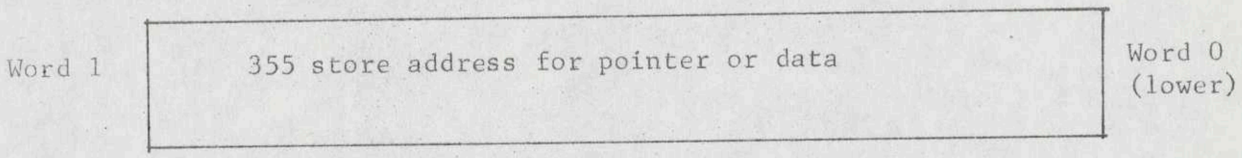


Set to one = last block of data for this run

Set to one = a pointer rather than data follows in the data area.

Set to one = table data follows in the data area.

AUXILIARY CONTROL WORD #2



9.15 DOCUMENT STATUS WORDS

Documents will be stored in a standard 355 GERTS block which consists of 32 18-bit words. The first nine words of each block will be made up of status information concerning the data and the status of the handler. The first four words of the eight are reserved for the GERTS 355 system while the last five will be used by the 355 DHCP.

355 Words	0 … 17	600 Words
1	Reserved for GERTS	* Upper
2	Reserved for GERTS	* Lower
3	RCW No. of Words in Record	1.Upper
4	L D (0) | RCW (1 – 9) | 10 | 355 ST. 14 | DH ST. 15 | 600 ST. 16 | P E 17	1.Lower
5	Handler Status Word Terminate Read | No. of Characters in Document	2.Upper
6	S.P. Index (0 – 11) | Pocket Doc. Put In (12 – 17)	2.Lower
7	355 Scan 0-5 Comm. Status | 6-11 User Error Code | 12 CP | 13 I1 | 14 I2 | 15 I3 | 16 DS | 17	3.Upper
8	User Message #1 0-5 User Message #2 6-11 Reserved 12 17	3.Lower
9	Data 24 18-bit words - 12 36-bit words	4.Upper

Note: all fields in binary except data.

* BLOCK ONLY. 1-4 in front of each record in the block. Upper half of word, in 6000, will contain number of words in the block. The number of words in the block does not include the block word count word.

9.15 DOCUMENT STATUS WORDS (continued)

Description of Document Status Words for 355/6000

Word 1.	Reserved for GERTS
Word 2.	Reserved for GERTS
Word 3.	No. of words in each record within block.
Word 4.	LD. - Last document indicator. Set to 1 = no more documents will be sent from 355 until handler is started again. 355 ST. - 355 stop indicator. Set to 1 = 355 stopped the 355 indicate a stop on no more buffers available in 355, etc. DH ST. - The document handler caused the stop as a result of 1.A manual stop, 2. a jam, 3. no pocket command, 4. Feeder/Pocket Alert, 5. Feed Alert. The handler status fields in word 5 and 6 0-9 will have to be tested to determine the exact reason for the stop. 600 ST. - The 600 stop indicator. Set to one indicates the 600 caused the stop. P.E. - Parity Error. The handler was not stopped but this document was read in error. However, it may have been read properly and a parity error was indicated. In any case, the 355 DHCP will attempt to process it.
Word 5.	bits 0-9. Handler status at terminate read time. 6000 DEP will test this word for approximate error control. See error conditions, page 9 of EPS.
Word 5.	bits 10-17. No. of characters read on document.
Word 6.	bits 0-11 = Send PI of float index.
Word 6.	bits 12-17. Pocket no document selected to by 355 DHCP.
Word 7.	355 Scan (355 DHCP) communication to 6000 DEP.

Cont. on Page 207

9.15 DOCUMENT STATUS WORDS (continued)

Codes	Description
17	The handler has been halted through the users sort control. DEP must look at word 8; bits 0-5, and 6-11 to see if a type out is required.
18	A user type out has been requested. The 6000 DEP must get the type out ID no. from word 8, bits 0-5 and 6-11.
21	The End of Sort Run has been given by the user. The 6000 DEP must look at word 8, bits 0-5 and 6-11 to see if a user type out is required. The DEP will type out a system message for end of run --------- "*ENDRUN XXX-yy" XXY = Job # yy = handler #
Word 7.	bits 6-11. User error code 1-63. The DEP will place this in the header of each document.
Word 7.	bit 12. Set to 1 = a check point document. The DEP will place an asterisk in the header record in document type field.
Bit 13-15	Set to 1 = a bill separator/or I1-13 type separator has been manually placed in the given pocket preceding the document. The DEP will set the document type field in the header to 1 if bit 13=1 to 2 if bit 11=1, to 3 if bit 15=1.
Word 7.	bit 16 = Document status Set to 1 = formatted document Set to 0 = non-formatted packed on one data
Word 8.	bit 0-5. Message index number to locate a message that the 6000 DEP must type out on the TTY for the handler.
	bits 6-11. Second message index to locate a message that the 6000 DEP must type out on the TTY for the handler.
	Note: Only two messages per document can be typed out at the users request. Bits 0-5, and 6-11 are used in conjunction with word 7, bits 0-5.

9.16 TRANSPOSITION CHECK DIGIT VERIFICATION

TCD will be performed by the software. The software must duplicate all TCD methods that were and are available in the hardware. In addition other TCD capabilities should be easily implemented in the software when required by the user. Below is a description of the TCD methods that are available within the hardware which the software must duplicate as a minimum TCD requirement.

The three basic methods of TCD verification have this in common: the TCD must be computed and added to the field. The TCD is normally added to the right end of the field and then becomes the low-order digit. However, the TCD can be assigned to any digit position desired, provided the correct weight is assigned. In each method of verification, the TCD feature uses the extra digit and, through computations made on the digits in the field, obtains a certain answer if the field is valid. If the answer is incorrect, the test fails and the document is sent to the reject pocket.

The three basic types of TCD verification are:

1. Mod 10 - Alternate 2's
2. Mod 11 - Natural progression
3. Mod 11 - Geometric progression

When the Mod 10, or alternate 2's method of TCD verification is installed, the TCD feature adds and tests the designated field using this formula:

$$\text{Sum} = X_{TCD} + 2X_1 + X_2 + 2X_3 + X_4 + 2X_5 + X_6 + 2X_7,$$

where $X_7\ X_6\ X_5\ X_4\ X_3\ X_2\ X_1\ X_{TCD}$ represents the tested field.

When a digit with a value within the range of 5 through 9 is used in a 2's multiplication term, each of the two product digits is individually added to the sum. If the TCD is valid, the least significant digit of the sum will be zero.

Given the account number 467-785-2, where 2 is the TCD, the verification would proceed as follows:

$$\begin{aligned} \text{Sum} &= 2 + 2(5) + 8 + 2(7) + 7 + 2(6) + 4 \\ &= 2 + 1 + 0 + 8 + 1 + 4 + 7 + 1 + 2 + 4 \\ &= 30 \end{aligned}$$

The account number is considered valid because the least significant digit of the sum is zero.

9.16 TRANSPOSITION CHECK DIGIT VERIFICATION (continued)

For Mod 10, the TCD is derived by using the formula above. For purposes of this calculation, the TCD is initially assumed to be zero. The derived sum is subtracted from the next highest number that ends with a zero. For example, if the sum is 34, then subtract 34 from 40 to give the TCD of 6.

Given the same account number as in the previous example--467-785-(0)--we can derive the TCD as follows:

$$\begin{aligned}\text{Sum} &= 0 + 2(5) + 8 + 2(7) + 7 + 2(6) + 4\\ &= 0 + 1 + 0 + 8 + 1 + 4 + 7 + 1 + 2 + 4\\ &= 28\\ \text{TCD} &= 30 - 28 = 2\end{aligned}$$

The account number would then become 467-785-2. Note that the dash is ignored. Should the account number be transposed by an encoding clerk, or the machine be unable to read a digit, or a digit be lost, then it is highly unlikely that the TCD logic would arrive at the answer of zero at the least significant digit.

When using the Mod 11 natural progression and geometric progression methods, two- and three- digit results of doubling or tripling the X value are used as decimal values. That is, 18 would be added as 18, not 1 + 8. The two Mod 11 methods are similar, except that a different system of progression is used for each. In both methods, the sum must be a multiple of 11 to be valid.

For the Mod 11 natural progression method of TCD verification, the TCD features adds and tests the designated field using this formula:

$X_n . . . X_4\ X_3\ X_2\ X_1\ X_{TCD}$ (represents tested field)

$$\text{Sum} = X_{TCD} + 2X_1 + 3X_2 + 4X_3 + 5X_4 + 6X_5 + . . . (n + 1)X_n$$

$$\text{Sum}/11 = \text{Quotient} + \text{Remainder}$$

For the TCD to be valid, the remainder must always be zero. As an example, assume an account number of 37286-2, where 2 is the TCD.

$$\begin{aligned}\text{Sum} &= 2 + 2(6) + 3(8) + 4(2) + 5(7) + 6(3)\\ &= 2 + 12 + 24 + 8 + 35 + 18\\ &= 99\end{aligned}$$

$$99/11 = 9$$

The TCD is valid, as the remainder is zero.

9.16 TRANSPOSITION CHECK DIGIT VERIFICATION (continued)

To derive the TCD for a given field the same formula is used, assuming zero for the TCD. Divide the sum by 11 to determine a quotient and remainder. If the remainder is 0, then the TCD will be zero. If not, then subtract the remainder from 11 to derive the TCD. That is,

TCD = 11 - remainder

By adding this TCD to the test sum, the test sum becomes a multiple of 11. When the field is correctly encoded and read by the reader/sorter, the TCD will always lead to a remainder of zero. In order to calculate the TCD for the number used above--37286(0)--the following procedure would be used:

Sum = (0) + 2(6) + 3(8) + 4(2) + 5(7) + 6(3)
  = 0 + 12 + 24 + 8 + 35 + 18
  = 97

97/11 = Quotient of 8 and remainder of 9

TCD = 11 - remainder = 11 - 9 = 2

The TCD for the Mod 11 natural progression and the Mod 11 geometric progression methods is restricted to one digit. If the derived TCD for a basic field is 10, then the basic field cannot be used.

Calculating a TCD value and testing a field with an included TCD for Mod 11 geometric progression is the same as for the natural progression method, except that the multiples for the X values progress geometrically, rather than arithmetically. For the number $X_n \; . \; . \; . X_4 \;\; X_3 \;\; X_2 \;\; X_1 \;\; X_{TCD}$ the TCD feature computes the sum as follows:

$$\text{Sum} = X_{TCD} + 2X_1 + 4X_2 + 8X_3 + 16X_4 + 32X_5 + 64X_6 \; .$$

Given a test field value of 14-16-10, the TCD would be derived as follows:

Sum = 2(0) + 4(1) + 8(6) + 16(1) + 32(4) + 64(1)
  = 0 + 4 + 48 + 16 + 128 + 64 = 260

260/11 = Quotient of 23 and remainder of 7
TCD = 11 - 7 = 4

Adding the TCD value of 4 to the previous sum of 260 gives 264, which is a multiple of 11. As mentioned above, the TCD can be any single digit 0-9.

9.17 DISPATCHING

The 355 Document Handler Control Program should be dispatched to on a priority basis. Naturally if only document handlers and console typewriters are configured on the system, this will be the case. However, looking ahead to running a restricted and limited communication system along with document handlers; the 355 operating system should run communications in a background mode. The system should only service communications requirements (wherever possible) when the 355 DHCP has relinquished its time because no documents are in memory to process.

9.18 STOP CONDITIONS

The following error conditions will require the handler on which they occur be stopped.

a. Jam
b. Full pocket alert and/or feeder alert
*c. Need a kill separator
*d. Need a block separator
*e. Need a batch separator

When a jam occurs the sorter will stop automatically, but those documents which preceded the jam should be processed and pocketed.

On a full pocket or feeder alert the sorter will stop (stop feed) and the documents which preceded the stop will be processed and selected.

10.0 SUPPORTING SOFTWARE

- 6000 GEMAP
- 355 MAP
- 355 Simulator
- 6000 COBOL
- 355 GERTS
- 6000 - 355 Bootload
- Test and Diagnostics (OPTS 6000)
- File System
- 355 Bootload (card reader to 355)
- GECOS III

11.0 GECOS CHANGES

- INCREASED NUMBER OF IOQ'S

  Presently standard GECOS provides for five IOQ's. Since the 6000 DES will have a need for at least 24 IOQ's a modification must be made to provide the required number of IOS's. This change has already been made for Time Sharing, and as mentioned there are personnel within OSE who understand the intricies of making this modification.

- CALLING THE 6000 DES FROM THE 6000 CONSOLE

  The means to spawn a job from the 6000 console already exists within GECOS. This will require adding the 6000 DES within the spawning mechanism of GECOS.

- ADDITIONAL SLAVE SERVICE AREA

  This will require no changes within GECOS. Use of the dollar limits card will result in GECOS assigning an extra SSA.

- MASTER MODE, QUE AND .LINKF

  The use of these features require only that the 6000 DES is a privileged slave running within GECOS control. The means to accomplish .Que and .LINKF are an integral part of GECOS.

- MASTER MODE, BUILD AND RELEASE PERM FILES ROUTINES

  This will be the most difficult requirement. The routing must be thoroughly tested to assure that it does not interfere with standard file procedures and that it does not affect the System and User Master Catalog System. This routine will handle the SPNXXY, DESXXX, TABXXY, MSTXXX files for the entry system.

12.0 6000 DOCUMENT ENTRY SYSTEM AND 355 DHCP SOFTWARE SYSTEM TEST REQUIREMENTS PHASE I AND II

General

The following information is to serve as a guide line for the requirements necessary to system test the 6000 Document Entry System on the MRS 200, DRD 200 and DRD 236 sorters.

- Software Test Requirements

Scope of System Test

---Insure the efficient operation of the 6000 Document Entry System (DES) under a real time environment with the 355 Document Handler Control Program (DHCP) under the control of GERTS 355 and GECOS III.

---Insure the efficient interaction of the 6000 Pocket Select Table Generator (PSTG) parameters with the 6000 Document Entry Program (DEP) and the 355 DHCP.

- Test Equipment

---Documents: MICR COC5 encoded documents.

---Full field encoder to alter and create existing or new fields on the MICR document.

---The pre-printed documents must be of high quality to assure consistent reading results.

- 6000 Document Entry Program Testing

---Insure by operator initiation that the 6000 DES program can be correctly spawned. Once the program is spawned, assure that program cannot be swapped out.

---Insure proper expansion of I/O queues depending on the number of document handlers on-line. One (1) I/O queue for each document handler.

---Insure that the 6000 DEP is correctly issuing a request for data within one GECOS Dispatch of 64 milliseconds.

---Insure the GECOS courtesy call feature for all real time I/O and the processing of the data will be done within the courtesy call.

---Insure the 6000 DEP minimization of writes to the mass media files.

- 6000 Document Entry Program Testing, Cont'd.

---Insure that the 6000 DEP will handle any mix of document handlers, e.g., MRS 200, DRD 200 and DRD 236.

---Insure that all control messages that are operator initiated through the TTYs are operational, e.g., starting, stopping and restarting the document handlers.

---Insure that access of data through the use of DEP to SLAVE communication (DESCMM) in the user coded module.

---Generate a user slave segment written in COBOL and GMAP to assure correct linkages to the 6000 DEP are being generated.

---Insure the basic 6000 DEP program is protected from user written segments.

---Insure that the 6000 DEP program is capable of processing all basic requirements (aside from user coded modules) in 30 milliseconds. This would be a worst case condition (6 data blocks to be processed from 6 different document handlers).

---Insure the slave capability to spawn the 355 boot load program and cause the 355 DHCM GERTS package to be loaded.

---Insure check points on the document handler are kept correct.

---Insure resources/status tables reflect correct status of 355 memory, active and inactive document handlers when document handler(s) are active.

---Insure that document count for each pocket and the total for all handlers on line is correct.

---Insure correctness of the following fields:

a. Handler status field
b. Pocket select field
c. Document status field

---Insure transferability to the alternate document handler when one handler fails during processing.

---Insure that roll back and restart capabilities will allow restarting in the event:

a. A handler or handlers fail
b. Failure of the 355 and/or 6000

- 6000 Document Entry Program Testing, Cont'd.

---Insure operator restart messages correspond to restart points. This should include the correct pocket select tables to the 355.

---Initiate restarts from one or more document handlers to verify correctness of data on mass media file.

---Verify that Permfile organization will allow 6000 DEP "All Permissions", i.e., read, write.

---Spawn correctly a reconciliation run upon request.

---Insure remote batch capabilities with another communication system, i.e., a second 355 must be used.

---Insure when using time sharing, no swap out occurs while the 6000 DEP is in core.

---Verify core release when the last batch is received from the 355 DHCP.

---Verify the capability of 6000 DEP to monitor correctly the number of times the document handler stops due to the following:

a. Buffers full in 6000 no data could be transferred to mass media (or tape).
b. Becuase of other users or I/O errors which caused the 355 to temporarily suspend operation of a handler(s).
c. Late pocket selects and missed read commands coming to the 355.

---Based on a pre-determined threshold limit, causes a serious degrading of the real time operation; perform an operator initiated kill of other jobs in core.

---Introduce a new threshold limit and repeat item listed directly above.

---Verify through operator initiation, the starting and stopping from any one of the six document handlers.

---Verify through operator initiation, the ability of killing a run coming from any one of the six document handlers.

---Verify the correctness of block and record size for a given run(s) according to user supplied data on the pocket select table.

- Pocket Select Table Generator (PSTG)

---Verify the PSTG ability to create and/or add a new table or tables from data on magnetic tape or punched card.

---Insure that a previously constructed pocket select table can be deleted from the file.

---Generate Transposition Check Digit (TCD) tables with MOD 10 and 11. In addition create various self-checking systems to verify the PSTG ability to generate correct TCD tables.

---Verify PSTG's ability to generate correctly from user supplied data QUE tables when one or more combinations of starting and ending ques are furnished.

---Verify PSTG's ability to generate correctly from user supplied data, one or more combinations of different field formats.

---Verify PSTG's ability to generate correctly from user supplied data, the control table. This table must correlate to the 6000 DES interval codes which are equivalent to the users code.

- Document Handler Control Program

---Verify DHCP is initiating on command from the 6000 DEP, start feeds and stop feeds.

---Verify DHCP is correctly selecting pockets based on user supplied data in the table. (Worst case 6 selects in 48 milliseconds.)

---Verify the document pocket is the actual document pocketed and transmitted to the 6000 DEP.

---Verify status pertinent to document handlers in every record transmitted to the 6000 DEP.

---Verify that on a stop feed, the DHCP allows for the documents in transit.

---Verify that the DHCP will select and pocket correctly documents in transit upon a stop feed issued from the 6000 DEP.

---Verify upon no request for data on a given document handler from the 6000 DEP, the buffers in 355 are full and the documents in transit are correctly pocketed.

- Document Handler Control Program, Cont'd.

---Verify after stop feed is issued to a given document handler that a normal operation can resume without operator intervention.

---Upon a hardware read error, verify that the document with the bad read is pocketed correctly and the status of the rejected document is in the record correctly.

---Verify no data was lost during transmission by comparing data in to data out.

---Verify that upon request from the 6000 DEP of transferring a job from a malfunctioning handler to another handler, the 355 DHCP correctly performs this operation while continuing production on the other handlers (requires a minimum of two document handlers).

---Perform the following data tests:

a. Dollar field
b. Blank field detection
c. Field limits to specified limits
d. Verify the capability of overriding error conditions with field under user control.
e. Perform invalid character tests on fields under user control.

13.0 DES6000 USERS MANUAL

DES6000 TABLE OF CONTENTS

13.1 VERBS

Action Verbs provide the user with complete logic flexibility for building sort runs. Validation and pocket selection of documents, including control documents, can be readily processed with the variety of action verbs offered. These verbs, along with the pocket tables and general tables, provide the user an easy to implement sort package which should cover nearly any kind of sort requirement. The action verbs by category are as follows:

13.1.1 Field/Subfield Indentification Verbs

- GETFIELD
- GETSUBF
- VALIDATE *

13.1.2 Test/Validation Verbs

- CKFORMAT
- CHECKTCD
- TEST
- SEARCH
- DIGITEST
- VALIDATE *

13.1.3 Logic Control Verbs

- ENDTEST
- SETSW
- RESETSW
- TESTSW
- TRANSFER

13.1.4 Document Handler Control Verbs

- STARTRUN
- ENDRUN
- TYPEOUT
- HALTDH

13.1.5 Special Purpose Verbs

- MASK
- CKPOINT
- SAVEFLD

*Combines GETFIELD and CKFORMAT verbs.

13.1 VERBS, Cont'd.

13.1.6 Device and Pocket Control Verbs

- DEVICE
- SETVALUE

13.1.7 File Control and Job Control Verbs

- FILECTL
- *SPAWNJOB

13.1.8 SLAVE Program to DEP Communication

*SPAWNJOB is provided through the use of the SPAWN control card see PSTG assembly control cards.

LEGION

Y = Must be filled in
O = Optional
Y = 1 of the fields must be filled in

GETFIELD - Establishes the existence of a defined field on a document. The GETFIELD or VALIDATE verbs must be used to locate and identify qued fields.

DOCUMENT ENTRY SYSTEM - SCAN RECORD INPUT

JOB RUN CONTROL			OPERATION			FIELD DEFINITION										SUBFIELD DEFINITION			TEST VALIDATION CONTROL										OPERAND TESTED SEARCHED	
	Job							Locators				Length							Format		TCD		Test Comparison						Table Name	
								Start		Stop														Success		Failure			Constant Name	
								Digit to Digit																					Literal Value	
Card Type	Run No.	Loop Time	Sequence	Action Verb	Switch Redefine	Type	Name I.D.	0-1 Qs	0-2 QUE	0-1 Qs	0-2 QUE	Maximum	Minimum	Process	Subfield Scale Ind.	Location Within Field	Length	Name I.D.	Format Control	Error Accept Count LRA	Mod	Type	Control <>	Pkt No.	Xfer Index	Pkt No.	Xfer Index	Error Code	Special Value	Value Type Ind.
1	2 3 4	5 6	7 8 9	10 11 12 13 14 15 16 17	18	19	20 21 22 23 24 25 26 27	28	29	30	31	32 33	34 35	36 37	38	39 40	41 42	43 44 45 46 47 48	49 50 51 52 53	54 55	56 57	58	59	60 61	62 63 64	65 66	67 68 69	70 71	72 73 74 75 76 77 78 79	80
Y	Y		Y	GETFIELD	Y		Y	Y	O			Y			O		O							O	O	O	O	O-USER CODE 01-63		
Ⓐ	Ⓑ	Ⓒ	Ⓓ	Ⓔ	Ⓕ	Ⓖ	Ⓗ	Ⓘ	Ⓙ	Ⓚ	Ⓛ	Ⓜ	Ⓝ	Ⓞ	Ⓟ	Ⓠ	Ⓡ	Ⓢ	Ⓣ	Ⓤ	Ⓥ	Ⓦ Ⓧ		Ⓨ	Z	1	②	③	④	⑤

Seq No.D to Transfer
Error Pocket
Seq. Number D to Transfer
Success Pocket
Length of scale
May be used to scale the field length L or R. **
If length of field is fixed M and N should be blank *
If M is filled in then N must have min. length.
May be used if field varies in length due to 0 suppression.
Que skip count 1-9, or 2nd Que which delimits field end.
Que which delimits end of the field, (BCD sybmols)
Que skip count or second Que which delimits field start.
Que that delimits start of the field (BCD symbol).
Unique name of the field.
N = Numeric field, D = dashes and numeric, X = dashes, plus/minus and numeric.
Re-defines, GETFIELD, may be re-defined up to 7 times 1-7 for different Ques.
GETFIELD -
Sequence No. In ascending sequence.
To be implemented later.
Job No. - 001-999.
Must contain the letter S.

* - USE PROCESS LENGTH WHEN FIELD LENGTH IS FIXED. FIELD LENGTH MAY BE 01-62.

** - L = LEFT, R = RIGHT.

(4) 72-79 - FIELD OUTPUT LOCATION 00000031-107.

GETSUBF - Establishes the existence of a defined field within a qued field on a document.

LEGION

Y - Must be filled in.
O = Optional
} Y = One of the fields must be filled in.

DOCUMENT ENTRY SYSTEM - SCAN RECORD INPUT

JOB RUN CONTROL			OPERATION			FIELD DEFINITION										SUBFIELD DEFINITION			TEST VALIDATION CONTROL										OPERAND TESTED SEARCHED	
	Job							Locators				Length							Format		TCD			Test Comparison						
								Start Digit to		Stop Digit														Success		Failure				
Card Type	Run No.	Loop Time	Sequence	Action Verb	Switch Redefine	Type	Name I.D.	Q1 Qs	Q2 QUE	Q1 Qs	Q2 QUE	Maximum	Minimum	Process	Subfield Scale Ind.	Location Within Field	Length	Name I.D.	Format Control	Error Accept Count LRA	Mod	Type	Control < >	Pkt No.	Xfer Index	Pkt No.	Xfer Index	Error Code	Table Name Constant Name Literal Value Special Value	Value Type Ind.
1	2 3 4	5 6	7 8 9	10 11 12 13 14 15 16 17	18	19	20 21 22 23 24 25 26 27	28	29	30	31	32 33	34 35	36 37	38	39 40	41 42	43 44 45 46 47 48	49 50 51 52 53	54 55	56 57	58	59	60 61	62 63 64	65 66	67 68 69	70 71	72 73 74 75 76 77 78 79	80
Y	Y		Y	Y		O	Y								Y	Y	Y	Y						O	O	O	O	O	-USER CODE 01-63	
Ⓐ	Ⓑ	Ⓒ	Ⓓ	Ⓔ	Ⓕ	Ⓖ	Ⓗ	Ⓘ	Ⓙ	Ⓚ	Ⓛ	Ⓜ	Ⓝ	Ⓞ	Ⓟ	Ⓠ	Ⓡ	Ⓢ	Ⓣ	Ⓤ	Ⓥ	Ⓦ	Ⓧ	Ⓨ	Z	1	②	③	④	⑤

Seq No.D to Transfer

N=NUMERIC

Error Pocket

D=DASHES & NUM.

Seq.No.D to Transfer

X=DASHES NUMERIC PLUS/ MINUS

Success Pocket

Name of Subfield

No. of Characters in subfield

Starting Location of Subfield within the Que Field L-R by character count from 1.

Must contain an "S"

Must contain the name of the Qued Field where the Subfield resides.

"GETSUBF" - This verb must be preceded by a "GETFIELD" verb *

Sequence No. in ascending sequence.

To be implemented later.

Job No. 001-999

Must contain an "S".

CE 326

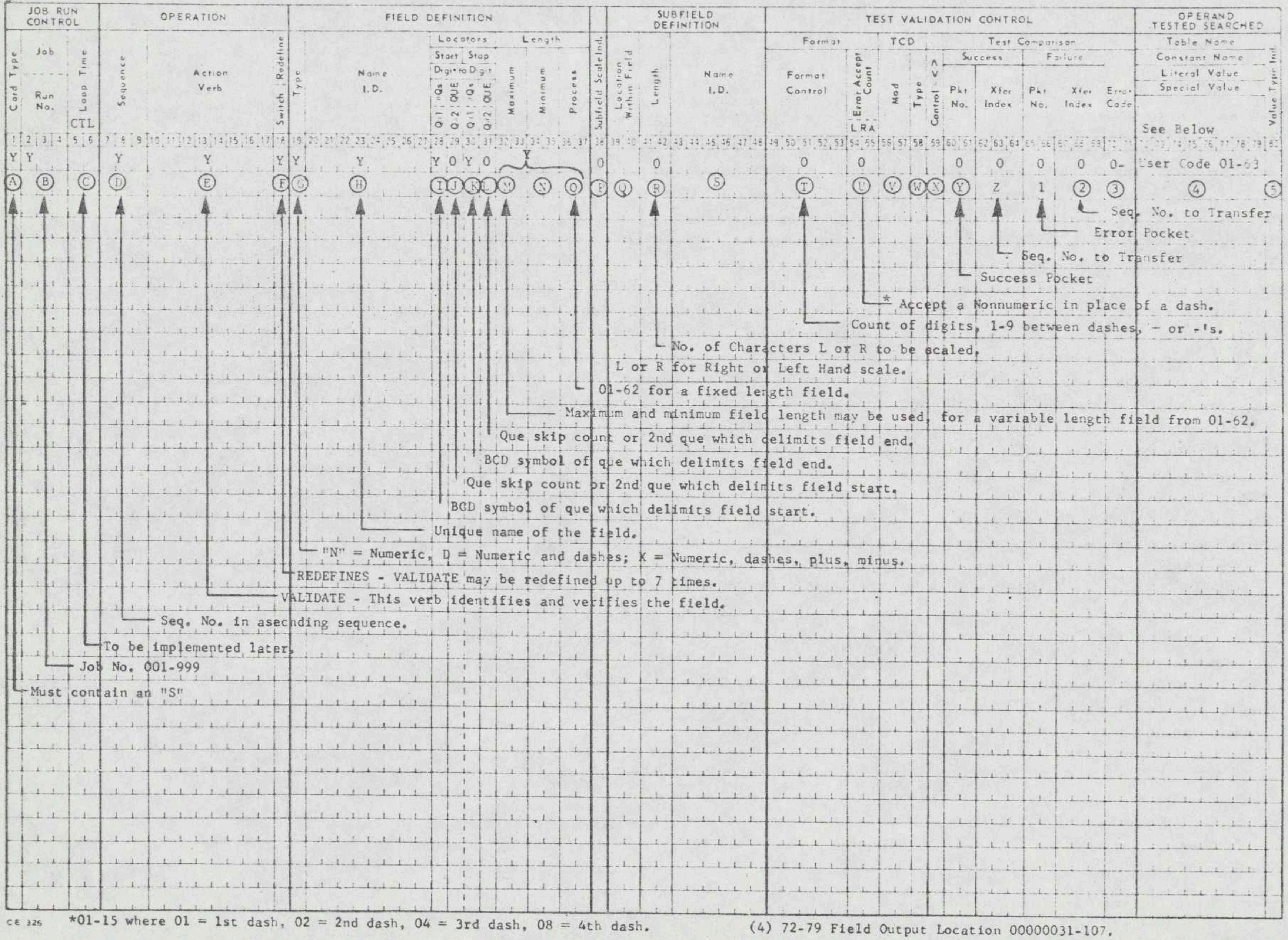
* May also be proceded by a "validate" verb.

Legion

Y = Must be filled in
0 = Optional
Y = One of the fields must be filled in

VALIDATE - Checks the given field for no character, proper formatting, and establishes the existence of the field.

DOCUMENT ENTRY SYSTEM - SCAN RECORD INPUT



Seq. No. to Transfer
Error Pocket
Seq. No. to Transfer
Success Pocket
* Accept a Nonnumeric in place of a dash.
Count of digits, 1-9 between dashes, - or -'s.
No. of Characters L or R to be scaled.
L or R for Right or Left Hand scale.
01-62 for a fixed length field.
Maximum and minimum field length may be used, for a variable length field from 01-62.
Que skip count or 2nd que which delimits field end.
BCD symbol of que which delimits field end.
Que skip count or 2nd que which delimits field start.
BCD symbol of que which delimits field start.
Unique name of the field.
"N" = Numeric, D = Numeric and dashes; X = Numeric, dashes, plus, minus.
REDEFINES - VALIDATE may be redefined up to 7 times.
VALIDATE - This verb identifies and verifies the field.
Seq. No. in asecnding sequence.
To be implemented later.
Job No. 001-999
Must contain an "S"

CE 326 *01-15 where 01 = 1st dash, 02 = 2nd dash, 04 = 3rd dash, 08 = 4th dash.

(4) 72-79 Field Output Location 00000031-107.

Legion

Y = Field must be filled in.
O = Optional field.
}Y = One of the fields must be filled in.

CKFORMAT - Allows the user to test a field for dashes, plus and minus signs in given positions within a field.

DØCUMENT ENTRY SYSTEM - SCAN RECORD INPUT

JOB RUN CONTROL	Card Type	1	Y	Ⓐ
	Job / Run No.	2 3 4	Y	Ⓑ
	Loop Time CTL	5 6	Y	Ⓒ
OPERATION	Sequence	7 8 9	Y	Ⓓ
	Action Verb	10 11 12 13 14 15 16 17	Y	Ⓔ
	Switch / Redefine	18	Y	Ⓕ
FIELD DEFINITION	Type	19	Y	Ⓖ
	Name I.D.	20 21 22 23 24 25 26 27	Y	Ⓗ
	Locators Start Digit to Stop Digit Q-1 Qs	28		Ⓘ
	Q-2 QUE	29		Ⓙ
	Q-1 Qs	30		Ⓚ
	Q-2 QUE	31		Ⓛ
	Length Maximum	32 33		Ⓜ
	Minimum	34 35		Ⓝ
	Process	36 37		Ⓞ
	Subfield Scale Ind.	38		Ⓟ
SUBFIELD DEFINITION	Location Within Field	39 40		Ⓠ
	Length	41 42		Ⓡ
	Name I.D.	43 44 45 46 47 48		Ⓢ
TEST VALIDATION CONTROL	Format / Format Control	49 50 51 52 53	Y	Ⓣ
	Error Accept Count / LRA	54 55	0	Ⓤ
	TCD Mod	56 57		Ⓥ
	Type	58		Ⓦ
	Control < >	59		Ⓧ
	Test Comparison Success Pkt No.	60 61	0	Ⓨ
	Xfer Index	62 63 64	0	Z
	Failure Pkt No.	65 66	0	1
	Xfer Index	67 68 69	0	②
	Error Code	70 71	0	③
OPERAND TESTED SEARCHED	Table Name / Constant Name / Literal Value / Special Value	72 73 74 75 76 77 78 79		④
	Value Type Ind.	80		⑤

User Code 01-63.

Seq. No. (D) to Transfer

Error Pocket

Seq. No. (D) to transfer.

Error Pocket

01-15 depending on number errors to be accepted. 01 = 1st dash, 02 = 2nd dash, 04 = 3rd and 08 = 4th dash.

Count of digits, 1-9, between dashes, plus or minus signs depending on type (G).

Unique field name which must have previously been identified by a GETFIELD or VALIDATE.

N, D, or X. D = Numeric and dashes, N = Numeric, X = Numeric +, -, or dashes.

1-7 depending on number of redefines.

* "CKFORMAT". CKFORMAT - must have been preceded by GETFIELD or VALIDATE.

Sequence No. in ascending sequence.

To be implemented later.

Job No. 001-999

Must contain an "S".

CE 326

Honeywell

Rev A

43A177886

Legion

Y = Field must be filled in.
O = Optional
}Y = One of the fields must be filled in.

CHECKTCD - Allows the user to perform transposition check digit test on the given field.

DOCUMENT ENTRY SYSTEM - SCAN RECORD INPUT

JOB RUN CONTROL			OPERATION			FIELD DEFINITION										SUBFIELD DEFINITION			TEST VALIDATION CONTROL										OPERAND TESTED SEARCHED	
								Locators				Length							Format		TCD			Test Comparison						
								Start Digit		to Stop Digit														Success		Failure				
Card Type	Job Run No.	Loop Time CTL	Sequence	Action Verb	Switch Redefine	Type	Name I.D.	Q-1 =Qs	Q-2 QUE	Q-1 =Qs	Q-2 QUE	Maximum	Minimum	Process	Subfield Scale Ind.	Location Within Field	Length	Name I.D.	Format Control	Error Accept Count LRA	Mod	Type	Control ><	Pkt No.	Xfer Index	Pkt No.	Xfer Index	Error Code	Table Name Constant Name Literal Value Special Value	Value Type Ind.
1	2 3 4	5 6	7 8 9	10 11 12 13 14 15 16 17	18	19	20 21 22 23 24 25 26 27	28	29	30	31	32 33	34 35	36 37	38	39 40	41 42	43 44 45 46 47 48	49 50 51 52 53	54 55	56 57	58	59	60 61	62 63 64	65 66	67 68 69	70 71	72 73 74 75 76 77 78 79	80
Y	Y																				Y	Y		0	0	0	0	0	0	0
Ⓐ	Ⓑ	Ⓒ	Ⓓ	Ⓔ	Ⓕ	Ⓖ	Ⓗ	Ⓘ	Ⓙ	Ⓚ	Ⓛ	Ⓜ	Ⓝ	Ⓞ	Ⓟ	Ⓠ	Ⓡ	Ⓢ	Ⓣ	Ⓤ	Ⓥ	Ⓦ	Ⓧ	Ⓨ	Z	1	②	③	④	⑤

⑤ T

④ Table Name for special TCD

③ User Code 01-63

② Seq. No. D to transfer.

1 Error Pocket

Z Seq. No. (D) to transfer.

Ⓨ Success Pocket No.

Ⓦ N = Natural G = Geometric
A = Alternate 2's. T = a special TCD.
N, G - 10,11 and A's are standard.

Ⓥ 10, 11 and alternate 2's are standard, others may be specified with T type.

Ⓗ Unique field name. Field name must have been identified previously by a GETFIELD or VALIDATE verb.

Ⓔ * "CHECKTCD", it must have been preceded by a VALIDATE or GETFIELD verb for the given field.

Ⓓ Sequence No. in ascending sequence.

Ⓒ To be implemented later.

Ⓑ Job No. 001-999.

Ⓐ Must contain the letter "S".

CE 326

TEST - Tests the given field against an operand, constant literal value or table name.

DØCUMENT ENTRY SYSTEM - SCAN RECORD INPUT

JOB RUN CONTROL			OPERATION			FIELD DEFINITION										SUBFIELD DEFINITION			TEST VALIDATION CONTROL											OPERAND TESTED SEARCHED	
								Locators				Length							Format		TCD			Test Comparison							
								Start Digit to		Stop Digit														Success		Failure				
Card Type	Job Run No.	Loop Time CTL	Sequence	Action Verb	Switch Redefine	Type	Name I.D.	Q=1 =0s	Q=2 QUE	Q=1 =0s	Q=2 QUE	Maximum	Minimum	Process	Subfield Scale Ind.	Location Within Field	Length	Name I.D.	Format Control	Error Accept Count LRA	Mod	Type	Control > <	Pkt No.	Xfer Index	Pkt No.	Xfer Index	Error Code	Table Name Constant Name Literal Value Special Value	Value Type Ind.
1	2 3 4	5 6	7 8 9	10 11 12 13 14 15 16 17	18	19	20 21 22 23 24 25 26 27	28	29	30	31	32 33	34 35	36 37	38	39 40	41 42	43 44 45 46 47 48	49 50 51 52 53	54 55	56 57	58	59	60 61	62 63 64	65 66	67 68 69	70 71	72 73 74 75 76 77 78 79	80
Y	Y		Y	Y			Y																Y	0	0	0	0	0	Y	Y
Ⓐ	Ⓑ	Ⓒ	Ⓓ	Ⓔ	Ⓕ	Ⓖ	Ⓗ	Ⓘ	Ⓙ	Ⓚ	Ⓛ	Ⓜ	Ⓝ	Ⓞ	Ⓟ	Ⓠ	Ⓡ	Ⓢ	Ⓣ	Ⓤ	Ⓥ	Ⓦ	Ⓧ	Ⓨ	Z	1	②	③	④	⑤

See above also may be a field name

K=constant name
T=table name
C=literal value
F=field*

User Code 01-63.

Seq. No. "D" to transfer

Error Pocket

Seq. No. "D" to transfer

Success Pocket

Compare condition =, ≠ must be listed in this field.

Unique name of the field to be tested. The field must have previously been identified with a GETFIELD or VALIDATE verb.

* TEST. This verb must have been preceded by GETFIELD or VALIDATE.

Sequence No. in ascending sequence.

To be implemented later.

Job No. 001-999

Must contain the letter "S".

---

*Use F if a field to field compare in the same document and no SAVEFLD will be used.

Legion

Y = Field must be filled in.
O = Optional
} Y = One of the fields must be filled in.

SEARCH - Allows the user to do a table look up with the given field against the table given in the operand and select a pocket number for the document.

DOCUMENT ENTRY SYSTEM - SCAN RECORD INPUT

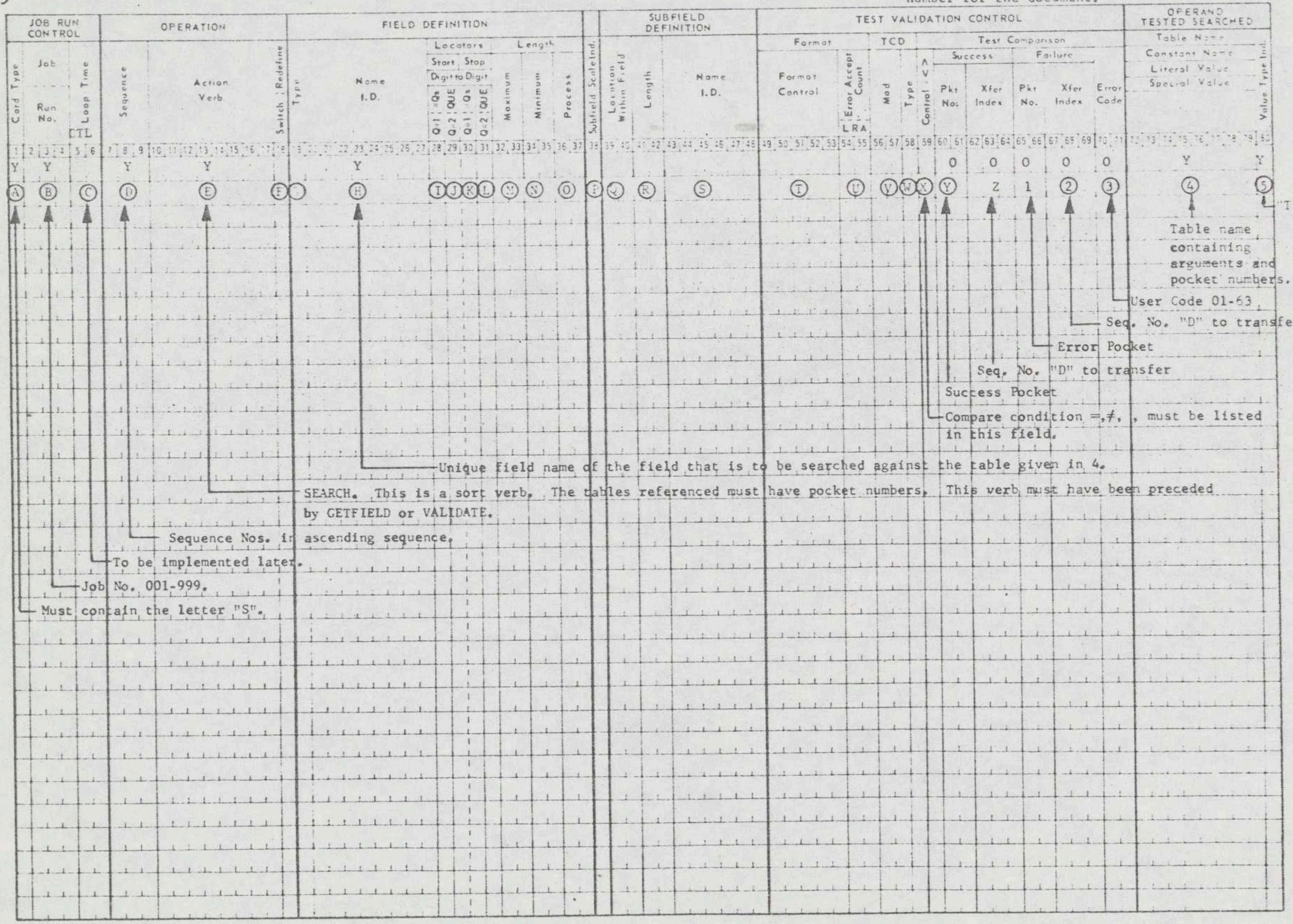


Table name containing arguments and pocket numbers.

User Code 01-63

Seq. No. "D" to transfer.

Error Pocket

Seq. No. "D" to transfer

Success Pocket

Compare condition =,≠, , must be listed in this field.

Unique field name of the field that is to be searched against the table given in 4.

SEARCH. This is a sort verb. The tables referenced must have pocket numbers. This verb must have been preceded by GETFIELD or VALIDATE.

Sequence Nos. in ascending sequence.

To be implemented later.

Job No. 001-999.

Must contain the letter "S".

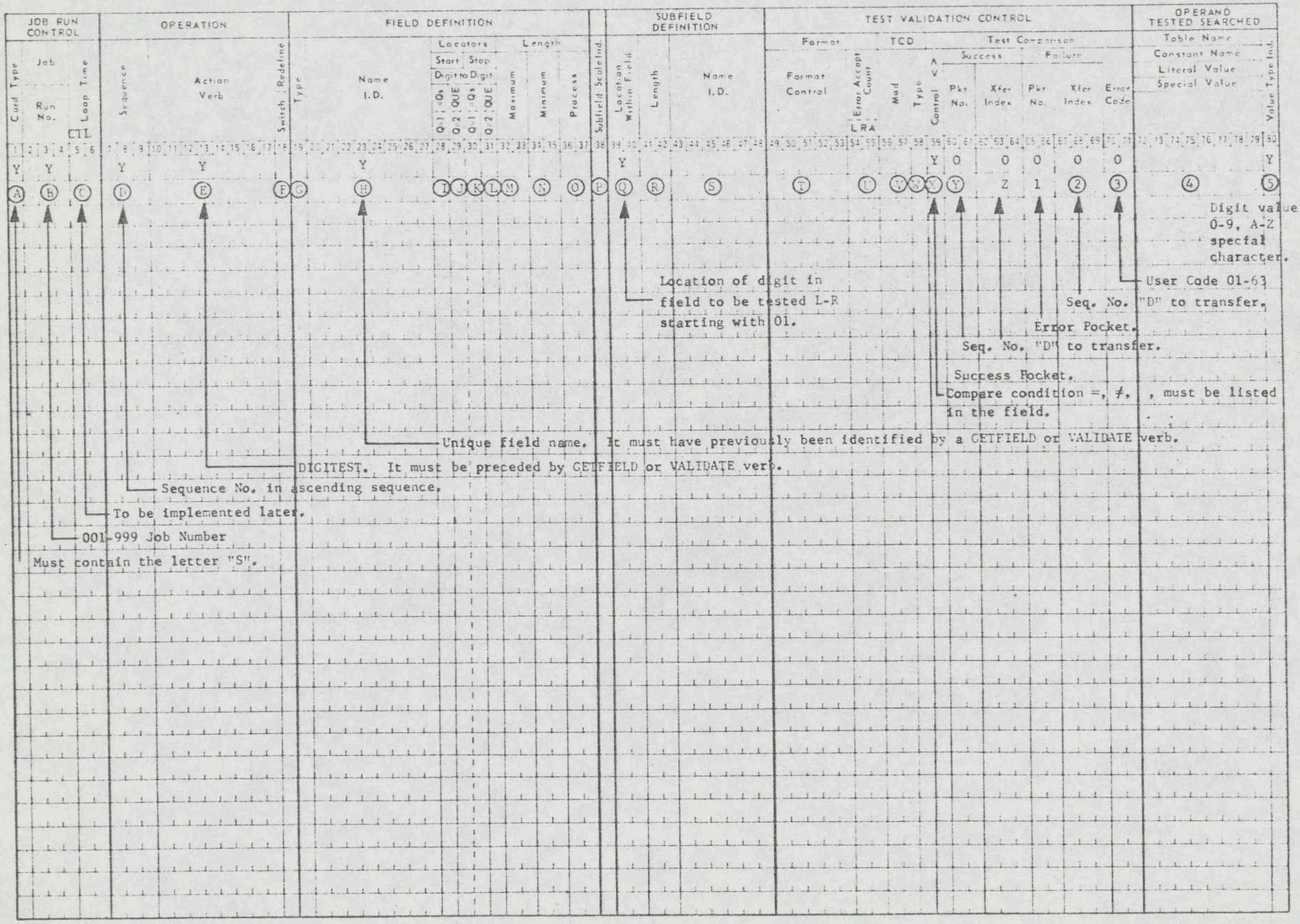
CE 326

Legion

Y = Field must be filled in.
O = Optional.
} Y = One of the fields must be filled in.

DIGITEST - Allows the user to test a given digit within the field against the character given in the value type indicator.

DOCUMENT ENTRY SYSTEM - SCAN RECORD INPUT



Digit value 0-9, A-Z special character.

User Code 01-63

Location of digit in field to be tested L-R starting with 01.

Seq. No. "D" to transfer.

Error Pocket.

Seq. No. "D" to transfer.

Success Pocket.

Compare condition =, ≠, , must be listed in the field.

Unique field name. It must have previously been identified by a GETFIELD or VALIDATE verb.

DIGITEST. It must be preceded by GETFIELD or VALIDATE verb.

Sequence No. in ascending sequence.

To be implemented later.

001-999 Job Number

Must contain the letter "S".

CE 326

Honeywell

Rev. A

43A177886

<u>Legion</u>

Y = Must be filled in.
O = Optional.
} Y = One of the fields must be filled in.

<u>ENDTEST</u> - Notifies the program that the testing is completed on this document and it is to be pocketed in the last received pocket whether success or failure.

DOCUMENT ENTRY SYSTEM - SCAN RECORD INPUT

JOB RUN CONTROL			OPERATION			FIELD DEFINITION											SUBFIELD DEFINITION			TEST VALIDATION CONTROL										OPERAND TESTED SEARCHED	
								Locators				Length								Format		TCD			Test Comparison						
								Start Digit to		Stop Digit															Success		Failure				
Card Type	Job Run No.	Loop Time CTL	Sequence	Action Verb	Switch Redefine	Type	Name I.D.	Q=1 Qs	Q=2 QUE	Q=1 Qs	Q=2 QUE	Maximum	Minimum	Process	Subfield Scale Ind.	Location Within Field	Length	Name I.D.	Format Control	Error Accept Count LRA	Mod	Type	Control < >	Pkt No.	Xfer Index	Pkt No.	Xfer Index	Error Code	Table Name Constant Name Literal Value Special Value	Value Type Ind.
1	2 3 4	5 6	7 8 9	10 11 12 13 14 15 16 17	18	19	20 21 22 23 24 25 26 27	28	29	30	31	32 33	34 35	36 37	38	39 40	41 42	43 44 45 46 47 48	49 50 51 52 53	54 55	56 57	58	59	60 61	62 63 64	65 66	67 68 69	70 71	72 73 74 75 76 77 78 79	80
Y																														
Ⓐ	Ⓑ	Ⓒ	Ⓓ	Ⓔ	Ⓕ	Ⓖ	Ⓗ	Ⓘ	Ⓙ	Ⓚ	Ⓛ	Ⓜ	Ⓝ	Ⓞ	Ⓟ	Ⓠ	Ⓡ	Ⓢ	Ⓣ	Ⓤ	Ⓥ	Ⓦ	Ⓧ	Ⓨ	Z	1	②	③	④	⑤

Ⓔ ENDTEST

Ⓓ Sequence No. in ascending sequence.

Ⓒ To be implemented later.

Ⓑ Job No. 001-999.

Ⓐ Must contain the letter "S".

CE 326

Legion

Y = Field must be filled in.
O = Optional.
} Y = One of the fields must be filled in.

SETSW Allows the user to set one of the device switches (0-9) to an ON position.

DOCUMENT ENTRY SYSTEM - SCAN RECORD INPUT

JOB RUN CONTROL			OPERATION			FIELD DEFINITION										SUBFIELD DEFINITION			TEST VALIDATION CONTROL										OPERAND TESTED SEARCHED	
								Locators				Length							Format		TCD		Test Comparison							
								Start Digit to Digit		Stop														Success		Failure				
Card Type	Job Run No.	Loop Time CTL	Sequence	Action Verb	Switch Redefine	Type	Name I.D.	Q.1 #Qs	Q.2 QUE	Q.1 Qs	Q.2 QUE	Maximum	Minimum	Process	Subfield Scale Ind.	Location Within Field	Length	Name I.D.	Format Control	Error Accept Count LRA	Mod	Type	Control < >	Pkt No.	Xfer Index	Pkt No.	Xfer Index	Error Code	Table Name Constant Name Literal Value Special Value	Value Type Ind.
1	2 3 4	5 6	7 8 9	10 11 12 13 14 15 16 17	18	19	20 21 22 23 24 25 26 27	28	29	30	31	32 33	34 35	36 37	38	39 40	41 42	43 44 45 46 47 48	49 50 51 52 53	54 55	56 57	58	59	60 61	62 63 64	65 66	67 68 69	70 71	72 73 74 75 76 77 78 79	80
Y	Y		Y	Y	Y																				0					
Ⓐ	Ⓑ	Ⓒ	Ⓓ	Ⓔ	Ⓕ	Ⓖ	Ⓗ	Ⓘ	Ⓙ	Ⓚ	Ⓛ	Ⓜ	Ⓝ	Ⓞ	Ⓟ	Ⓠ	Ⓡ	Ⓢ	Ⓣ	Ⓤ	Ⓥ	Ⓦ	Ⓧ	Ⓨ	Z	1	②	③	④	⑤

Z — Transfer on success to Seq. No. "D" may be indicated.

Ⓕ — Switch No. 0-9, 10 switches are provided 0-9.

Ⓔ — SETSW

Ⓓ — Sequence No. in ascending sequence.

Ⓒ — To be implemented later.

Ⓑ — Job No. 001-999.

Ⓐ — Must contain an "S"

CE 326

Honeywell

Rev A

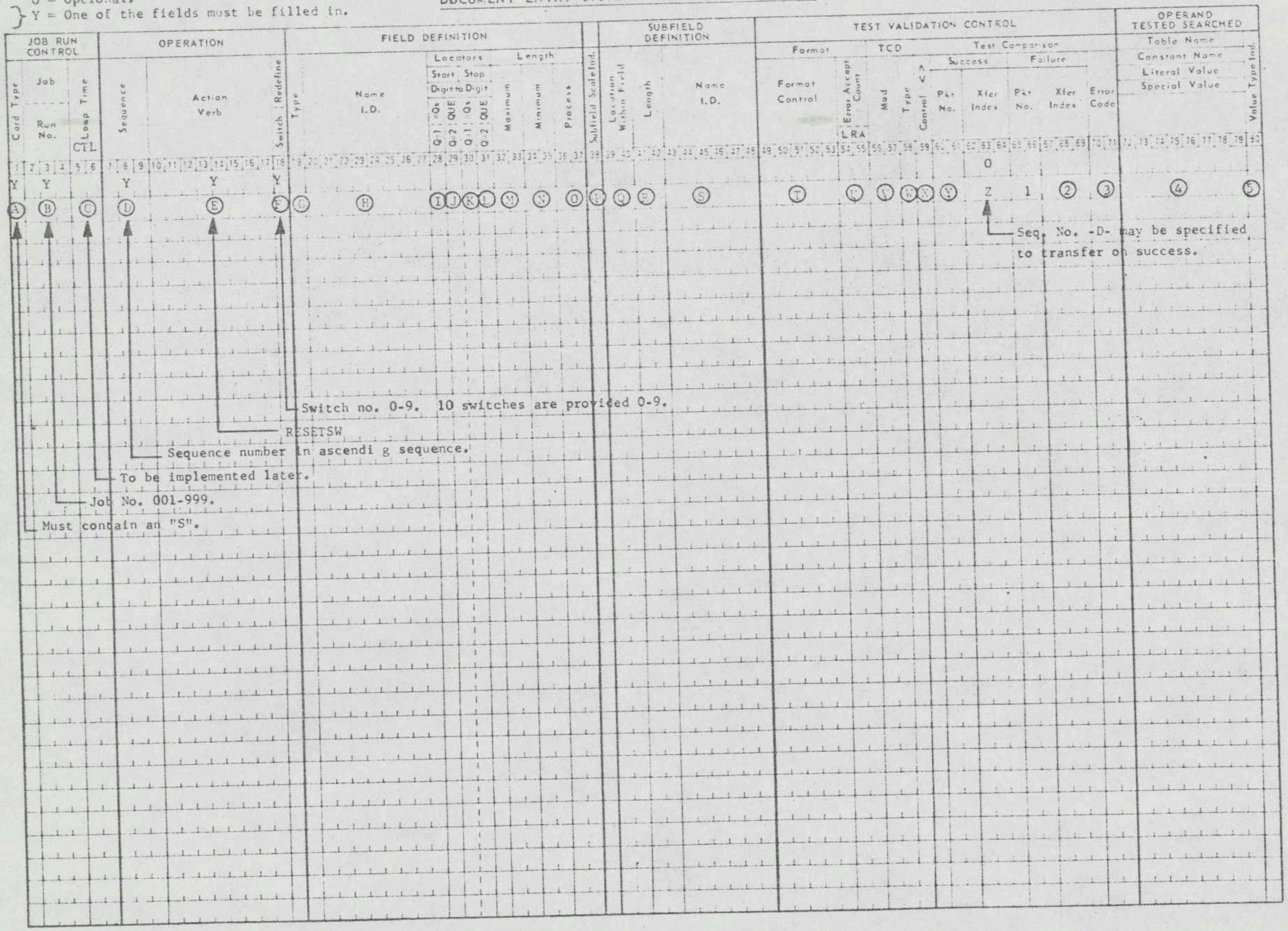
43A177886

Legion

Y = Field must be filled in.
O = Optional.
} Y = One of the fields must be filled in.

RESETSW - Allows the user to set one of the device switches (0-9) to an off condition (0).

DØCUMENT ENTRY SYSTEM - SCAN RECORD INPUT



Seq. No. -D- may be specified to transfer on success.

Switch no. 0-9. 10 switches are provided 0-9.

RESETSW

Sequence number in ascendi g sequence.

To be implemented later.

Job No. 001-999.

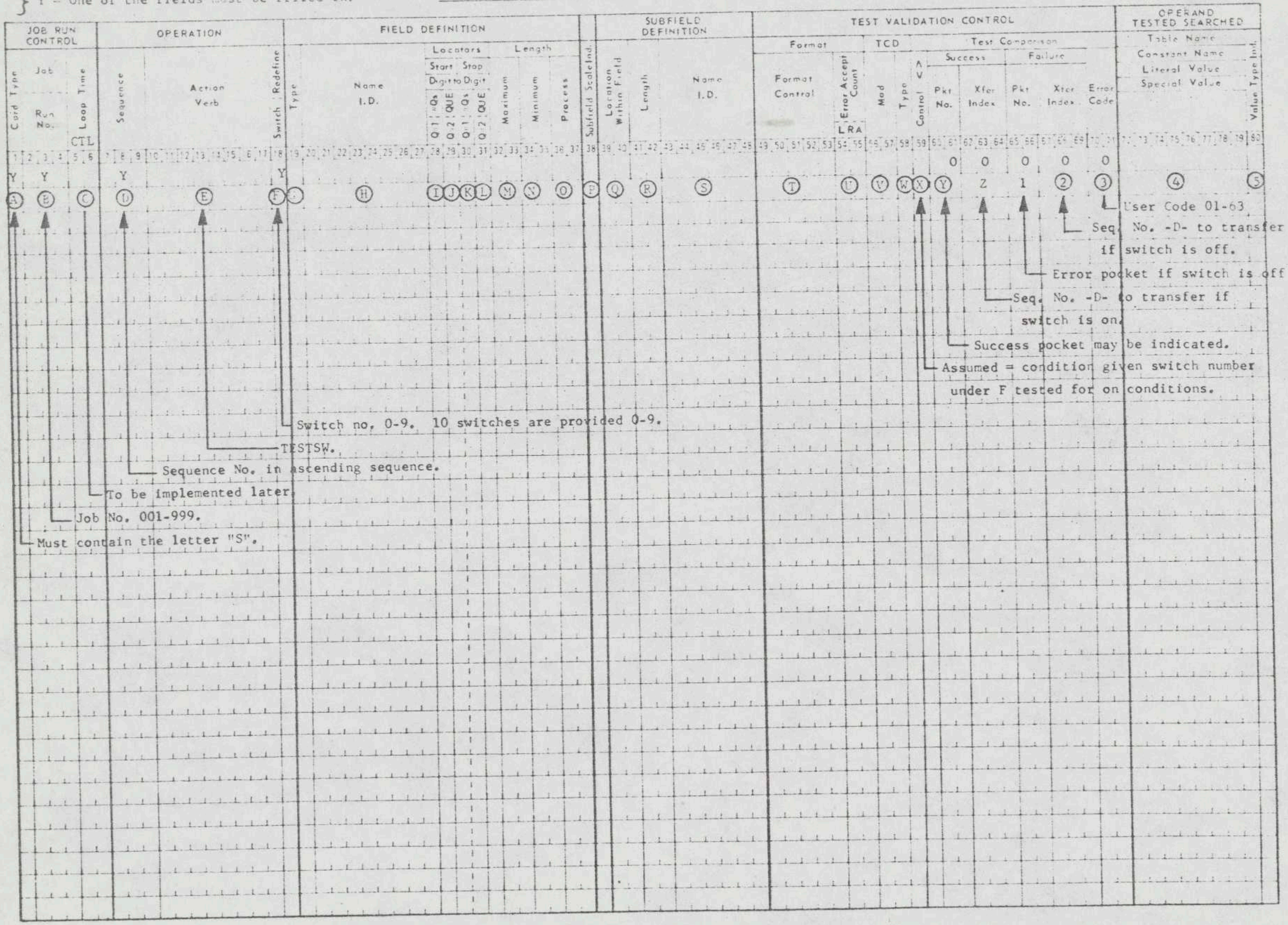
Must contain an "S".

Legion

Y = Field must be filled in.
0 = Optional.
} Y = One of the fields must be filled in.

TESTSW - Allows the user to test one of the device switches (0-9) for an on condition. If the switch is on the test is considered successful. If the switch is off the test fails.

DOCUMENT ENTRY SYSTEM - SCAN RECORD INPUT



JOB RUN CONTROL			OPERATION			FIELD DEFINITION										SUBFIELD DEFINITION			TEST VALIDATION CONTROL										OPERAND TESTED SEARCHED	
								Locators				Length							Format		TCD		Test Comparison							
								Start Digit to		Stop Digit														Success		Failure				
Card Type	Job Run No.	Loop Time CTL	Sequence	Action Verb	Switch Redefine	Type	Name I.D.	Q.1 0s	Q.2 QUE	Q.1 0s	Q.2 QUE	Maximum	Minimum	Process	Subfield Scale Ind.	Location Within Field	Length	Name I.D.	Format Control	Error Accept Count LRA	Mod	Type	< > Control	Pkt No.	Xfer Index	Pkt No.	Xfer Index	Error Code	Table Name Constant Name Literal Value Special Value	Value Type Ind.
1	2 3 4	5 6	7 8 9	10 11 12 13 14 15 16 17	18	19	20 21 22 23 24 25 26 27	28	29	30	31	32 33	34 35	36 37	38	39 40	41 42	43 44 45 46 47 48	49 50 51 52 53	54 55	56 57	58	59	60 61	62 63 64	65 66	67 68 69	70 71	72 73 74 75 76 77 78 79	80
Y	Y		Y		Y																			0	0	0	0	0		
Ⓐ	Ⓑ	Ⓒ	Ⓓ	Ⓔ	Ⓕ	Ⓖ	Ⓗ	Ⓘ	Ⓙ	Ⓚ	Ⓛ	Ⓜ	Ⓝ	Ⓞ	Ⓟ	Ⓠ	Ⓡ	Ⓢ	Ⓣ	Ⓤ	Ⓥ	Ⓦ	Ⓧ	Ⓨ	Z	1	②	③	④	⑤

Legion

Y = Field must be filled in.
O = Optional.
} Y = One of the fields must be filled in.

DOCUMENT ENTRY SYSTEM - SCAN RECORD INPUT

TRANSFER - Allows the user to make an unconditional transfer to another scan record defined in the transfer index. The transfer index must contain the sequence number "D" to XFER.

JOB RUN CONTROL			OPERATION			FIELD DEFINITION										SUBFIELD DEFINITION				TEST VALIDATION CONTROL										OPERAND TESTED SEARCHED	
								Locators				Length								Format		TCD			Test Comparison						
								Start Digit	to	Stop Digit															Success		Failure				
Card Type	Job Run No.	Loop Time CTL	Sequence	Action Verb	Switch Redefine	Type	Name I.D.	Q-1 =Q+	Q-2 QUE	Q-1 Q+	Q-2 QUE	Maximum	Minimum	Process	Subfield Scale Ind.	Location Within Field	Length	Name I.D.	Format Control	Error Accept Count LRA	Mod	Type	Control < >	Pkt No.	Xfer Index	Pkt No.	Xfer Index	Error Code	Table Name Constant Name Literal Value Special Value	Value Type Ind.
1	2 3 4	5 6	7 8 9	10 11 12 13 14 15 16 17	18	19	20 21 22 23 24 25 26 27	28	29	30	31	32 33	34 35	36 37	38	39 40	41 42	43 44 45 46 47 48	49 50 51 52 53	54 55	56 57	58	59	60 61	62 63 64	65 66	67 68 69	70 71	72 73 74 75 76 77 78 79	80
Y	Y		Y	Y																					Y					
Ⓐ	Ⓑ	Ⓒ	Ⓓ	Ⓔ	Ⓕ	Ⓖ	Ⓗ	Ⓘ	Ⓙ	Ⓚ	Ⓛ	Ⓜ	Ⓝ	Ⓞ	Ⓟ	Ⓠ	Ⓡ	Ⓢ	Ⓣ	Ⓤ	Ⓥ	Ⓦ	Ⓧ	Ⓨ	Z	1	②	③	④	⑤

Sequence No. -D- to transfer.

TRANSFER

Sequence No. in ascending sequence.

To be implemented later.

Job No. 001-999.

Must contain an "S".

Legion

Y = Field must be filled in.
O = Optional.

STARTRUN - Pseudo operation to notify PSTG that Scan Sort Logic is commencing.

DOCUMENT ENTRY SYSTEM - SCAN RECORD INPUT

JOB RUN CONTROL			OPERATION			FIELD DEFINITION										SUBFIELD DEFINITION			TEST VALIDATION CONTROL										OPERAND TESTED SEARCHED	
								Locators				Length							Format		TCD		Test Comparison						Table Name Constant Name Literal Value Special Value	
								Start Digit to		Stop Digit														Success		Failure				
Card Type	Job Run No.	Loop Time CTL	Sequence	Action Verb	Switch Redefine	Type	Name I.D.	Q-1 Qs	Q-2 QUE	Q-1 Qs	Q-2 QUE	Maximum	Minimum	Process	Subfield Scale Ind	Location Within Field	Length	Name I.D.	Format Control	Error Accept Count LRA	Mod	Type	Control < >	Pkt No.	Xfer Index	Pkt No.	Xfer Index	Error Code		Value Type Ind.
1	2 3 4	5 6	7 8 9	10 11 12 13 14 15 16 17	18	19	20 21 22 23 24 25 26 27	28	29	30	31	32 33	34 35	36 37	38	39 40	41 42	43 44 45 46 47 48	49 50 51 52 53	54 55	56 57	58	59	60 61	62 63 64	65 66	67 68 69	70 71	72 73 74 75 76 77 78 79	80
Y	Y		Y	Y																										
Ⓐ	Ⓑ	Ⓒ	Ⓓ	Ⓔ	Ⓕ	Ⓖ	Ⓗ	Ⓘ	Ⓙ	Ⓚ	Ⓛ	Ⓜ	Ⓝ	Ⓞ	Ⓟ	Ⓠ	Ⓡ	Ⓢ	Ⓣ	Ⓤ	Ⓥ	Ⓦ	Ⓧ	Ⓨ	Z	1	②	③	④	⑤

STARTRUN

Seq. No. in ascending sequence.

Must contain an "S".

CE 326

Legion

Y = Field must be filled in.
O = Optional.
} Y = One of the fields must be filled in.

ENDRUN - Notifies the system software that the sort run is complete, so that files may be closed housekeeping performed, etc.

DOCUMENT ENTRY SYSTEM - SCAN RECORD INPUT

JOB RUN CONTROL			OPERATION			FIELD DEFINITION									
								Locators				Length			
								Start Stop Digit to Digit							
Card Type	Job Run No.	Loop Time CTL	Sequence	Action Verb	Switch / Redefine	Type	Name I.D.	Q-1 Qs	Q-2 QUE	Q-1 Qs	Q-2 QUE	Maximum	Minimum	Process	Subfield Scale Ind.
1	2 3 4	5 6	7 8 9	10 11 12 13 14 15 16 17	18	19	20 21 22 23 24 25 26 27	28	29	30	31	32 33	34 35	36 37	38
Y	Y		Y	Y											
Ⓐ	Ⓑ	Ⓒ	Ⓓ	Ⓔ	Ⓕ	Ⓖ	Ⓗ	Ⓘ	Ⓙ	Ⓚ	Ⓛ	Ⓜ	Ⓝ	Ⓞ	Ⓟ

SUBFIELD DEFINITION			TEST VALIDATION CONTROL											OPERAND TESTED SEARCHED	
			Format		TCD			Test Comparison						Table Name	
								Success		Failure				Constant Name Literal Value Special Value	
Location Within Field	Length	Name I.D.	Format Control	Error Accept Count LRA	Mod	Type	Control < >	Pkt No.	Xfer Index	Pkt No.	Xfer Index	Error Code			Value Type Ind.
39 40	41 42	43 44 45 46 47 48	49 50 51 52 53	54 55	56	57	58 59	60 61	62 63 64	65 66	67 68 69	70 71		72 73 74 75 76 77 78 79	80
Ⓠ	Ⓡ	Ⓢ	Ⓣ	Ⓤ	Ⓥ	Ⓦ	Ⓧ	Ⓨ	Z	1	②	③		④	⑤

Ⓔ — ENDRUN

Ⓓ — Sequence No. is ascending sequence.

Ⓒ — To be implemented later.

Ⓑ — Job No. 001-999.

Ⓐ — Must contain the letter "S".

Legion

Y = Field must be filled in.
O = Optional.
} Y = One of the fields must be filled in.

TYPEOUT - Allows the user to type messages to the console typewriter or TTY servicing the handler, the document handler may or may not be in halt mode.

DOCUMENT ENTRY SYSTEM - SCAN RECORD INPUT

JOB RUN CONTROL			OPERATION			FIELD DEFINITION									SUBFIELD DEFINITION			TEST VALIDATION CONTROL										OPERAND TESTED SEARCHED	
								Locators				Length							Format		TCD		Test Comparison						
								Start Digit to		Stop Digit														Success		Failure			
Card Type	Job Run No.	Loop Time CTL	Sequence	Action Verb	Switch / Redefine	Type	Name I.D.	Q=1 Qs	Q=2 QUE	Q=1 Qs	Q=2 QUE	Maximum	Minimum	Process	Subfield Scale Ind.	Location Within Field	Length	Name I.D.	Format Control	Error Accept Count LRA	Mod	Type	Control < >	Pkt No.	Xfer Index	Pkt No.	Xfer Index	Error Code	Table Name Constant Name Literal Value Special Value	Value Type Ind.
1	2 3 4	5 6	7 8 9	10 11 12 13 14 15 16 17	18	19	20 21 22 23 24 25 26 27	28	29	30	31	32 33	34 35	36 37	38	39 40	41 42	43 44 45 46 47 48	49 50 51 52 53	54 55	56 57	58	59	60 61	62 63 64	65 66	67 68 69	70 71	72 73 74 75 76 77 78 79	80
Y	Y		Y	Y																				O	O				Y	Y
Ⓐ	Ⓑ	Ⓒ	Ⓓ	Ⓔ	Ⓕ	Ⓖ	Ⓗ	Ⓘ	Ⓙ	Ⓚ	Ⓛ	Ⓜ	Ⓝ	Ⓞ	Ⓟ	Ⓠ	Ⓡ	Ⓢ	Ⓣ	Ⓤ	Ⓥ	Ⓦ	Ⓧ	Ⓨ	Z	1	②	③	④	⑤

④ "C" or [illegible]

④ Literal value constant name to be typed out.

Z Seq. No. -D- to transfer to may be indicated.

Ⓨ Pocket No. for document after typeout executed.

Ⓔ TYPEOUT

Ⓓ Sequence no. in ascending order.

Ⓒ To be implemented later.

Ⓑ Job No. 001-999.

Ⓐ Must contain an "S".

Legion

Y = Field must be filled in.
O = Optional.
} Y = One of the fields must be filled in.

HALTDH - Allows the user to put the document handler in the Halt Mode. The document which caused the halt and subsequent documents in transit will either go to the last pocket received or the pocket indicated in the HALTDH verb.

DOCUMENT ENTRY SYSTEM - SCAN RECORD INPUT

JOB RUN CONTROL	Card Type	1	Y	Ⓐ
	Job Run No.	2 3 4	Y	Ⓑ
	Loop Time CTL	5 6		Ⓒ
OPERATION	Sequence	7 8 9	Y	Ⓓ
	Action Verb	10 11 12 13 14 15 16 17	Y	Ⓔ
	Switch Redefine	18		Ⓕ
FIELD DEFINITION	Type	19		Ⓖ
	Name I.D.	20 21 22 23 24 25 26 27		Ⓗ
	Locators Start Digit to Q=1 Qs	28		Ⓘ
	Locators Start Digit to Q=2 QUE	29		Ⓙ
	Locators Stop Digit Q=1 Qs	30		Ⓚ
	Locators Stop Digit Q=2 QUE	31		Ⓛ
	Length Maximum	32 33		Ⓜ
	Length Minimum	34 35		Ⓝ
	Process	36 37		Ⓞ
	Subfield Scale Ind.	38		Ⓟ
SUBFIELD DEFINITION	Location Within Field	39 40		Ⓠ
	Length	41 42		Ⓡ
	Name I.D.	43 44 45 46 47 48		Ⓢ
TEST VALIDATION CONTROL	Format: Format Control	49 50 51 52 53		Ⓣ
	Format: Error Accept Count LRA	54 55		Ⓤ
	TCD: Mod	56 57		Ⓥ
	TCD: Type	58		Ⓦ
	Control - < >	59		Ⓧ
	Test Comparison Success Pkt No.	60 61	O	Ⓨ
	Test Comparison Success Xfer Index	62 63 64	O	Z
	Test Comparison Failure Pkt No.	65 66		1
	Test Comparison Failure Xfer Index	67 68 69		②
	Error Code	70 71		③
OPERAND TESTED SEARCHED	Table Name Constant Name Literal Value Special Value	72 73 74 75 76 77 78 79	Y	④
	Value Type Ind	80	Y	⑤

Ⓐ - Must contain an "S".

Ⓑ - Job No. 001-999.

Ⓒ - To be implemented later.

Ⓓ - Sequence No. in ascending sequence.

Ⓔ - HALTDH

Ⓨ - Pocket the document and the in transit documents are to go to following the HALTDH.

Z - Seq. No. -D- to transfer to after handler is started.

④ - Constant name. Literal value to be typed out.

⑤ - K or C

CE 326

Legion

Y = Field must be filled in.
O = Optional.
} Y = One of the fields must be filled in.

MASK - Allows the user to set a mask to strip leading zero positions of all subsequent items for that field so that error characters in unused leading positions will not cause document rejects.

DOCUMENT ENTRY SYSTEM - SCAN RECORD INPUT

Group	Field	Card columns	Entry	Ref.
JOB RUN CONTROL	Card Type	1	Y	Ⓐ
	Job Run No.	2–4	Y	Ⓑ
	Loop Time	5–6 CTL		Ⓒ
OPERATION	Sequence	7–9	Y	Ⓓ
	Action Verb	10–17	Y	Ⓔ
	Switch ; Redefine	18		Ⓕ
FIELD DEFINITION	Type	19		Ⓖ
	Name I.D.	20–27	Y	Ⓗ
	Locators Start Digit Q1–Q4	28		Ⓘ
	Locators Start QUE	29		Ⓙ
	Locators Stop Digit Q1–Q4	30		Ⓚ
	Locators Stop QUE	31		Ⓛ
	Length Maximum	32–33		Ⓜ
	Length Minimum	34–35		Ⓝ
	Length Process	36–37		Ⓞ
	Subfield Scale Ind.	38		Ⓟ
SUBFIELD DEFINITION	Location Within Field	39–40		Ⓠ
	Length	41–42		Ⓡ
	Name I.D.	43–48		Ⓢ
TEST VALIDATION CONTROL	Format Control	49–53		Ⓣ
	Error Accept Count LRA	54–55		Ⓤ
	TCD Mod	56		Ⓥ
	TCD Type	57–58		Ⓦ
	Control < >	59		Ⓧ
	Test Comparison Success Pkt. No.	60–61		Ⓨ
	Test Comparison Success Xfer Index	62–64	O	Z
	Test Comparison Failure Pkt. No.	65–66		1
	Test Comparison Failure Xfer Index	67–69		②
	Error Code	70–71		③
OPERAND TESTED SEARCHED	Table Name / Constant Name / Literal Value / Special Value	72–79	Y	④
	Value Type Ind.	80	Y	⑤ C

Ⓐ Must contain an "S".

Ⓑ Job No. 001-999.

Ⓒ To be implemented later.

Ⓓ Sequence No. in ascending sequence.

Ⓔ MASK

Ⓗ Name of field to be masked.

Z Seq. No. -D- to transfer to after mask is executed.

④ Constant Name where field from which mask is to be derived is located. This field must have been saved with a SAVEFLD.

Honeywell

Rev. A

43A177886

Legion

Y = Field must be filled in.
O = Optional.
}Y = One of the fields must be filled in.

CKPOINT - Allows the user to ask for a restart checkpoint from a key document such as block header, batch header or checkpoint document.

DOCUMENT ENTRY SYSTEM - SCAN RECORD INPUT

JOB RUN CONTROL			OPERATION			FIELD DEFINITION										SUBFIELD DEFINITION			TEST VALIDATION CONTROL										OPERAND TESTED SEARCHED	
								Locators				Length							Format		TCD		Test Comparison						Table Name Constant Name Literal Value Special Value	
								Start Digit to		Stop Digit														Success		Failure				
Card Type	Job Run No.	Loop Time CTL	Sequence	Action Verb	Switch ¦ Redefine	Type	Name I.D.	Q=1 ¦ =Qs	Q=2 ¦ QUE	Q=1 ¦ =Qs	Q=2 ¦ QUE	Maximum	Minimum	Process	Subfield Scale Ind.	Location Within Field	Length	Name I.D.	Format Control	Error Accept Count LRA	Mod	Type	Control = < >	Pkt No.	Xfer Index	Pkt No.	Xfer Index	Error Code		Value Type Ind.
1	2 3 4	5 6	7 8 9	10 11 12 13 14 15 16 17	18	19	20 21 22 23 24 25 26 27	28	29	30	31	32 33	34 35	36 37	38	39 40	41 42	43 44 45 46 47 48	49 50 51 52 53	54 55	56 57	58	59	60 61	62 63 64	65 66	67 68 69	70 71	72 73 74 75 76 77 78 79	80
Y	Y		Y	Y																				O	O					
Ⓐ	Ⓑ	Ⓒ	Ⓓ	Ⓔ	Ⓕ	Ⓖ	Ⓗ	Ⓘ	Ⓙ	Ⓚ	Ⓛ	Ⓜ	Ⓝ	Ⓞ	Ⓟ	Ⓠ	Ⓡ	Ⓢ	Ⓣ	Ⓤ	Ⓥ	Ⓦ	Ⓧ	Ⓨ	Z	1	②	③	④	⑤

Seq. No. -D- to transfer after CKPOINT is executed.

Pocket to put document after checkpoint verb is executed.

CKPOINT

Sequence No. in ascending sequence.

To be implemented later.

Job No. 001-999.

Must contain an "S".

CE 326

Honeywell

Rev. A

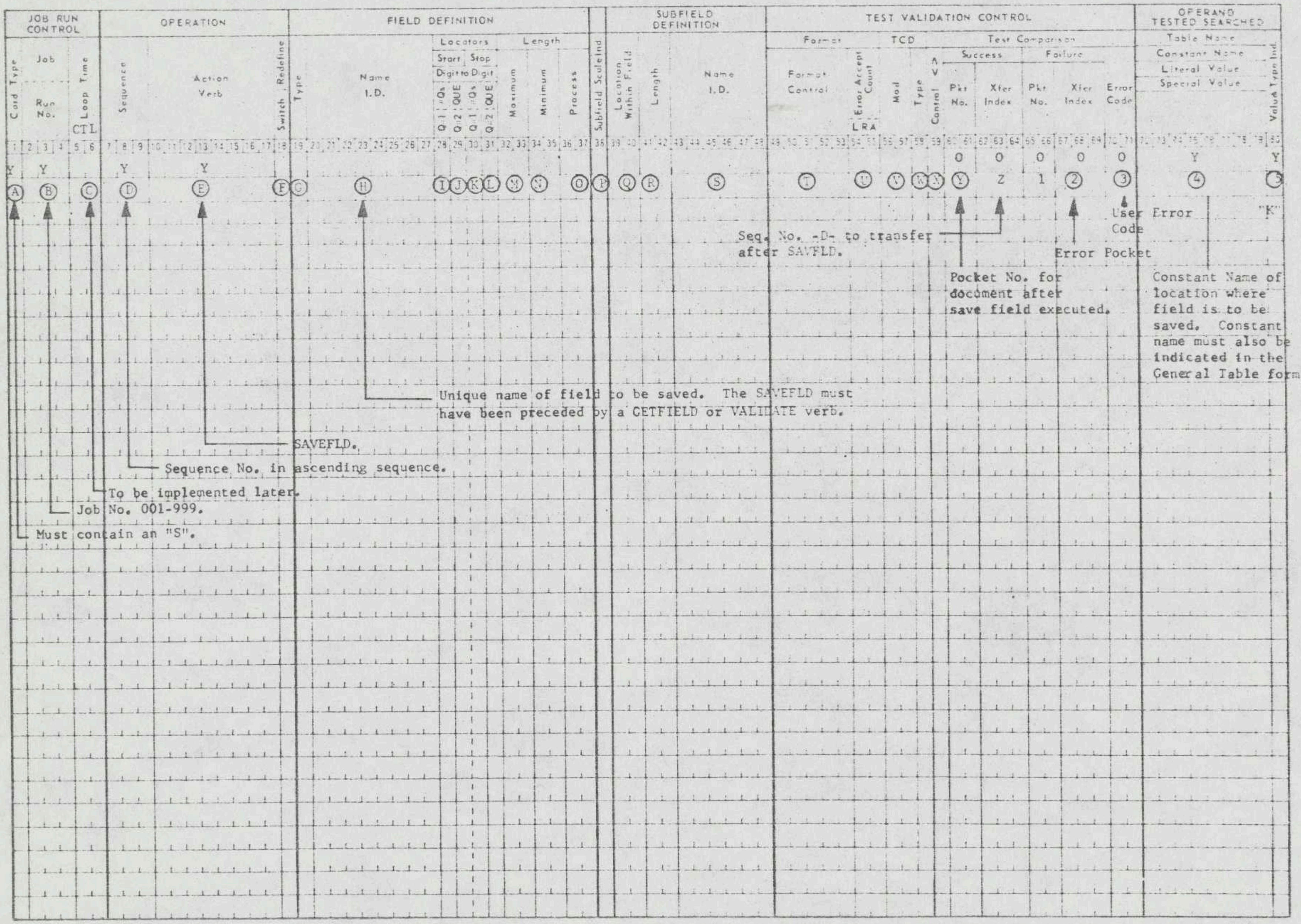
43A177886

Legion

Y = Field must be filled in.
O = Optional.
} Y = One of the fields must be filled in.

SAVEFLD - Allows the user to save fields from the documents, such as, batch and block headers for subsequent tests against item document fields.

DOCUMENT ENTRY SYSTEM – SCAN RECORD INPUT



JOB RUN CONTROL			OPERATION			FIELD DEFINITION										SUBFIELD DEFINITION			TEST VALIDATION CONTROL										OPERAND TESTED SEARCHED	
								Locators				Length							Format		TCD			Test Comparison						
								Start Digit		to Stop Digit														Success		Failure				
Card Type	Job Run No.	Loop Time CTL	Sequence	Action Verb	Switch, Redefine	Type	Name I.D.	Q 1 #Qs	Q 2 QUE	Q 1 #Qs	Q 2 QUE	Maximum	Minimum	Process	Subfield Scaleind	Location Within Field	Length	Name I.D.	Format Control	Error Accept Count LRA	Mod	Type	Control < >	Pkt No.	Xfer Index	Pkt No.	Xfer Index	Error Code	Table Name Constant Name Literal Value Special Value	Value Type Ind.
1	2 3 4	5 6	7 8 9	10 11 12 13 14 15 16 17	18	19	20 21 22 23 24 25 26 27	28	29	30	31	32 33	34 35	36 37	38	39 40	41 42	43 44 45 46 47 48	49 50 51 52 53	54 55	56 57	58	59	60 61	62 63 64	65 66	67 68 69	70 71	72 73 74 75 76 77 78 79	80
Y	Y		Y	Y																				O	O	O	O	O	Y	Y
Ⓐ	Ⓑ	Ⓒ	Ⓓ	Ⓔ	Ⓕ	Ⓖ	Ⓗ	Ⓘ	Ⓙ	Ⓚ	Ⓛ	Ⓜ	Ⓝ	Ⓞ	Ⓟ	Ⓠ	Ⓡ	Ⓢ	Ⓣ	Ⓤ	Ⓥ	Ⓦ	Ⓧ	Ⓨ	Z	1	②	③	④	⑤
																												User Code	Error	"K"

Seq. No. -D- to transfer after SAVFLD.

Error Pocket

Pocket No. for document after save field executed.

Constant Name of location where field is to be saved. Constant name must also be indicated in the General Table form.

Unique name of field to be saved. The SAVEFLD must have been preceded by a GETFIELD or VALIDATE verb.

SAVEFLD.

Sequence No. in ascending sequence.

To be implemented later.

Job No. 001-999.

Must contain an "S".

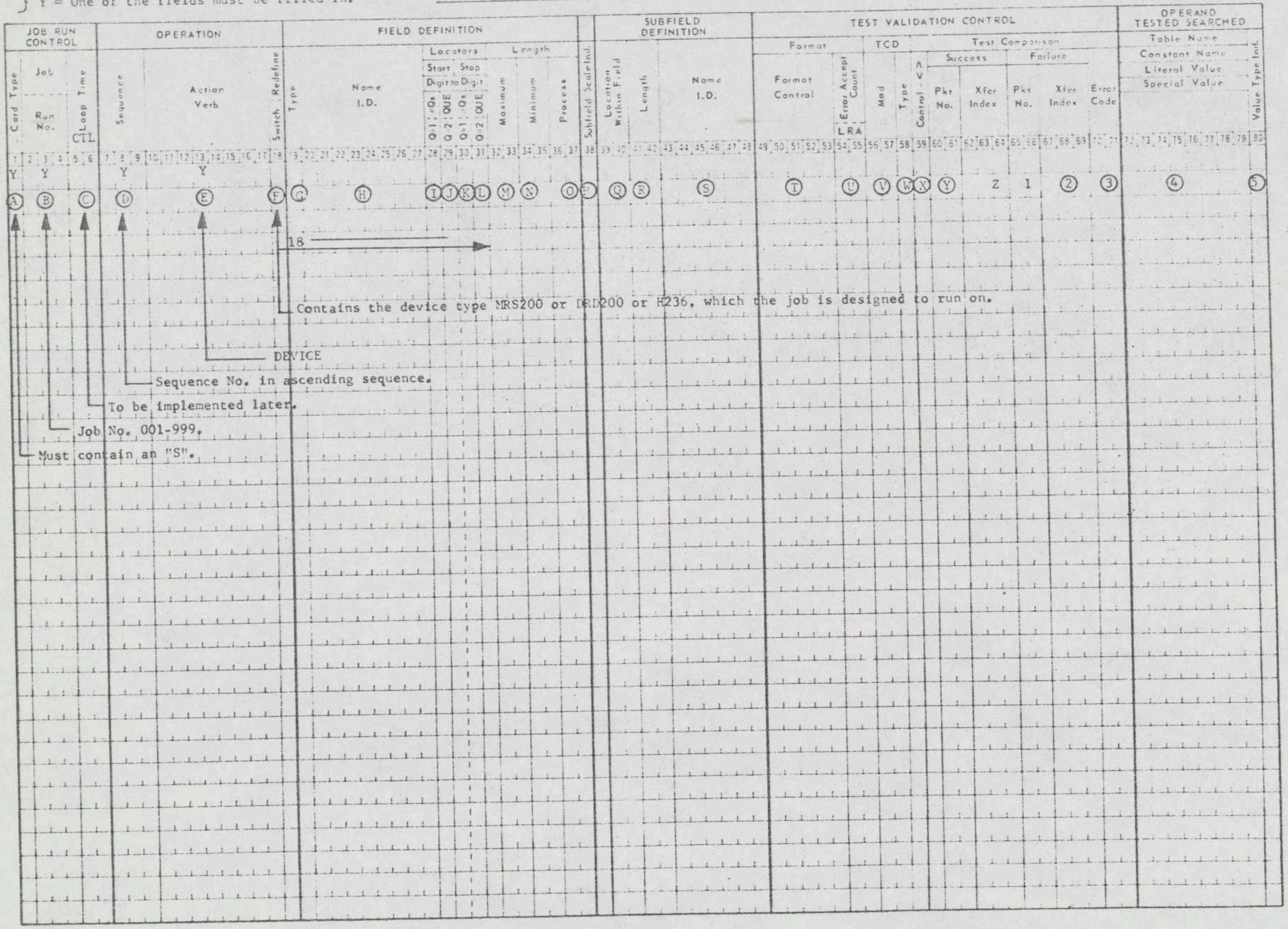
CE 326

DEVICE - Notifies the Pocket Select Table Generator the type of device that the program is to be generated for.

Legion

Y = Field must be filled in.
O = Optional.
} Y = One of the fields must be filled in.

DOCUMENT ENTRY SYSTEM - SCAN RECORD INPUT



JOB RUN CONTROL			OPERATION			FIELD DEFINITION									
								Locators				Length			
								Start Digit to		Stop Digit					
Card Type	Job Run No.	Loop Time CTL	Sequence	Action Verb	Switch Redefine	Type	Name I.D.	Q=1 =Qs	Q=2 QUE	Q=1 =Qs	Q=2 QUE	Maximum	Minimum	Process	Subfield Scale Ind.
1	2 3 4	5 6	7 8 9	10 11 12 13 14 15 16 17	18	19 20	21 22 23 24 25 26 27	28	29	30	31	32 33	34 35	36 37	38
Y	Y		Y	Y											
Ⓐ	Ⓑ	Ⓒ	Ⓓ	Ⓔ	Ⓕ	Ⓖ	Ⓗ	Ⓘ	Ⓙ	Ⓚ	Ⓛ	Ⓜ	Ⓝ	Ⓞ	Ⓟ

SUBFIELD DEFINITION			TEST VALIDATION CONTROL										OPERAND TESTED SEARCHED	
			Format		TCD		Test Comparison							
								Success		Failure				
Location Within Field	Length	Name I.D.	Format Control	Error Accept Count LRA	Mod	Type	Control < >	Pkt No.	Xfer Index	Pkt No.	Xfer Index	Error Code	Table Name Constant Name Literal Value Special Value	Value Type Ind.
39 40	41 42	43 44 45 46 47 48	49 50 51 52 53	54 55	56 57	58	59	60 61	62 63 64	65 66	67 68 69	70 71	72 73 74 75 76 77 78 79	80
Ⓠ	Ⓡ	Ⓢ	Ⓣ	Ⓤ	Ⓥ	Ⓦ	Ⓧ	Ⓨ	Z	1	②	③	④	⑤

Contains the device type MRS200 or DRD200 or H236, which the job is designed to run on.

DEVICE

Sequence No. in ascending sequence.

To be implemented later.

Job No. 001-999.

Must contain an "S".

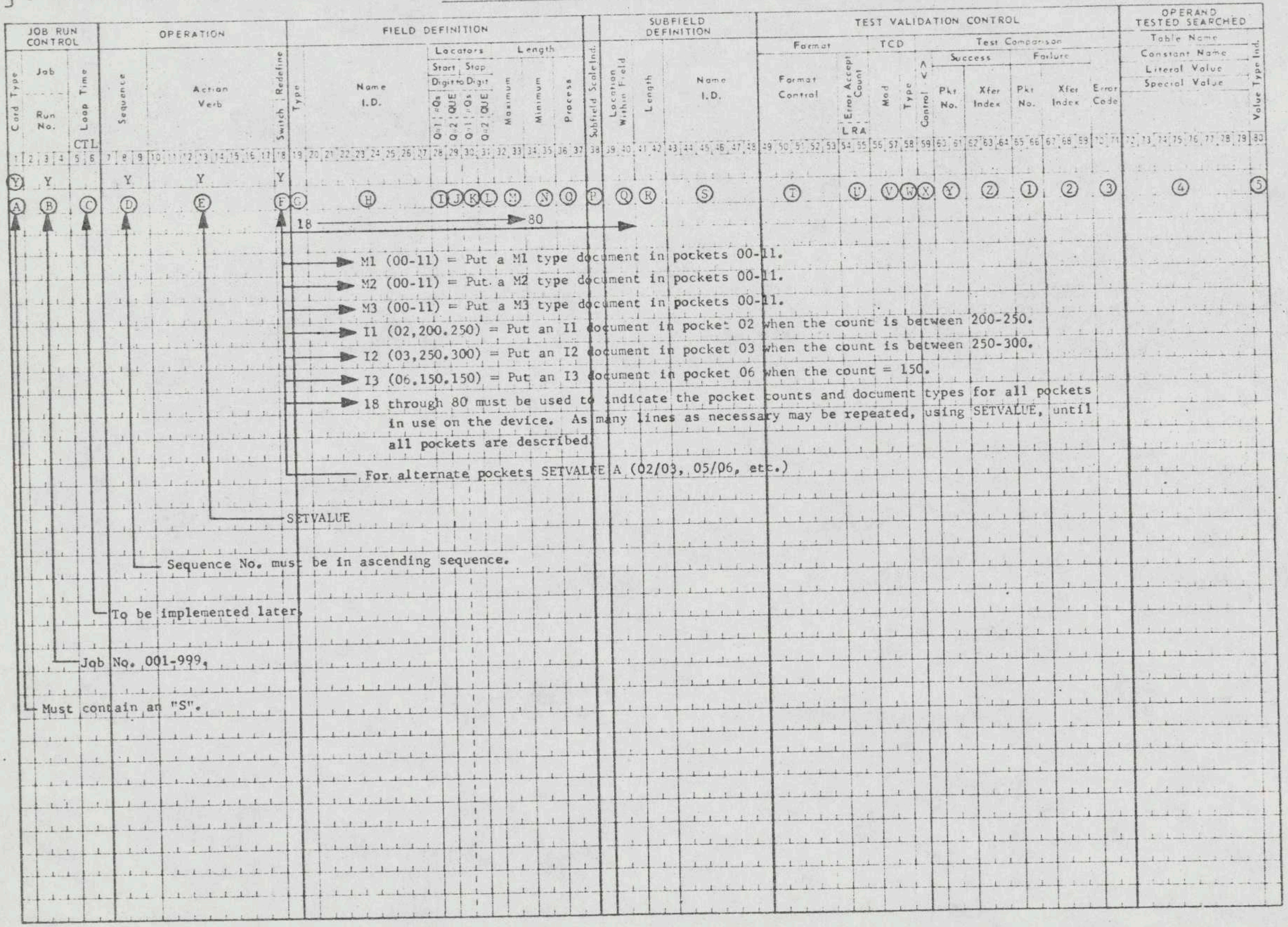
CE 326

Legion

Y = Field must be filled in.
O = Optional.
} Y = One of the fields must be filled in.

SETVALUE - Allows the user to specify to the system software the type of control documents in use by type, by pocket and count -M1-M3 and I1-I3.

DOCUMENT ENTRY SYSTEM - SCAN RECORD INPUT



JOB RUN CONTROL			OPERATION			FIELD DEFINITION										SUBFIELD DEFINITION			TEST VALIDATION CONTROL										OPERAND TESTED SEARCHED	
								Locators				Length							Format		TCD			Test Comparison						
								Start Stop Digit to Digit																Success		Failure				
Card Type	Job Run No.	Loop Time CTL	Sequence	Action Verb	Switch Redefine	Type	Name I.D.	Q=1 #Qs	Q:2 QUE	Q=1 #Qs	Q=2 QUE	Maximum	Minimum	Process	Subfield Scale Ind.	Location Within Field	Length	Name I.D.	Format Control	Error Accept Count LRA	Mod	Type	Control <>	Pkt No.	Xfer Index	Pkt No.	Xfer Index	Error Code	Table Name Constant Name Literal Value Special Value	Value Type Ind.
1	2 3 4	5 6	7 8 9	10 11 12 13 14 15 16 17	18	19	20 21 22 23 24 25 26 27	28	29	30	31	32 33	34 35	36 37	38	39 40	41 42	43 44 45 46 47 48	49 50 51 52 53	54 55	56 57	58	59	60 61	62 63 64	65 66	67 68 69	70 71	72 73 74 75 76 77 78 79	80
Ⓨ	Y		Y	Y	Y																									
Ⓐ	Ⓑ	Ⓒ	Ⓓ	Ⓔ	Ⓕ	Ⓖ	Ⓗ	Ⓘ	Ⓙ	Ⓚ	Ⓛ	Ⓜ	Ⓝ	Ⓞ	Ⓟ	Ⓠ	Ⓡ	Ⓢ	Ⓣ	Ⓤ	Ⓥ	Ⓦ	Ⓧ	Ⓨ	Ⓩ	①	②	③	④	⑤

18 → 80

M1 (00-11) = Put a M1 type document in pockets 00-11.

M2 (00-11) = Put a M2 type document in pockets 00-11.

M3 (00-11) = Put a M3 type document in pockets 00-11.

I1 (02,200.250) = Put an I1 document in pocket 02 when the count is between 200-250.

I2 (03,250.300) = Put an I2 document in pocket 03 when the count is between 250-300.

I3 (06.150.150) = Put an I3 document in pocket 06 when the count = 150.

18 through 80 must be used to indicate the pocket counts and document types for all pockets in use on the device. As many lines as necessary may be repeated, using SETVALUE, until all pockets are described.

For alternate pockets SETVALUE A (02/03, 05/06, etc.)

SETVALUE

Sequence No. must be in ascending sequence.

To be implemented later.

Job No. 001-999.

Must contain an "S".

CE 326

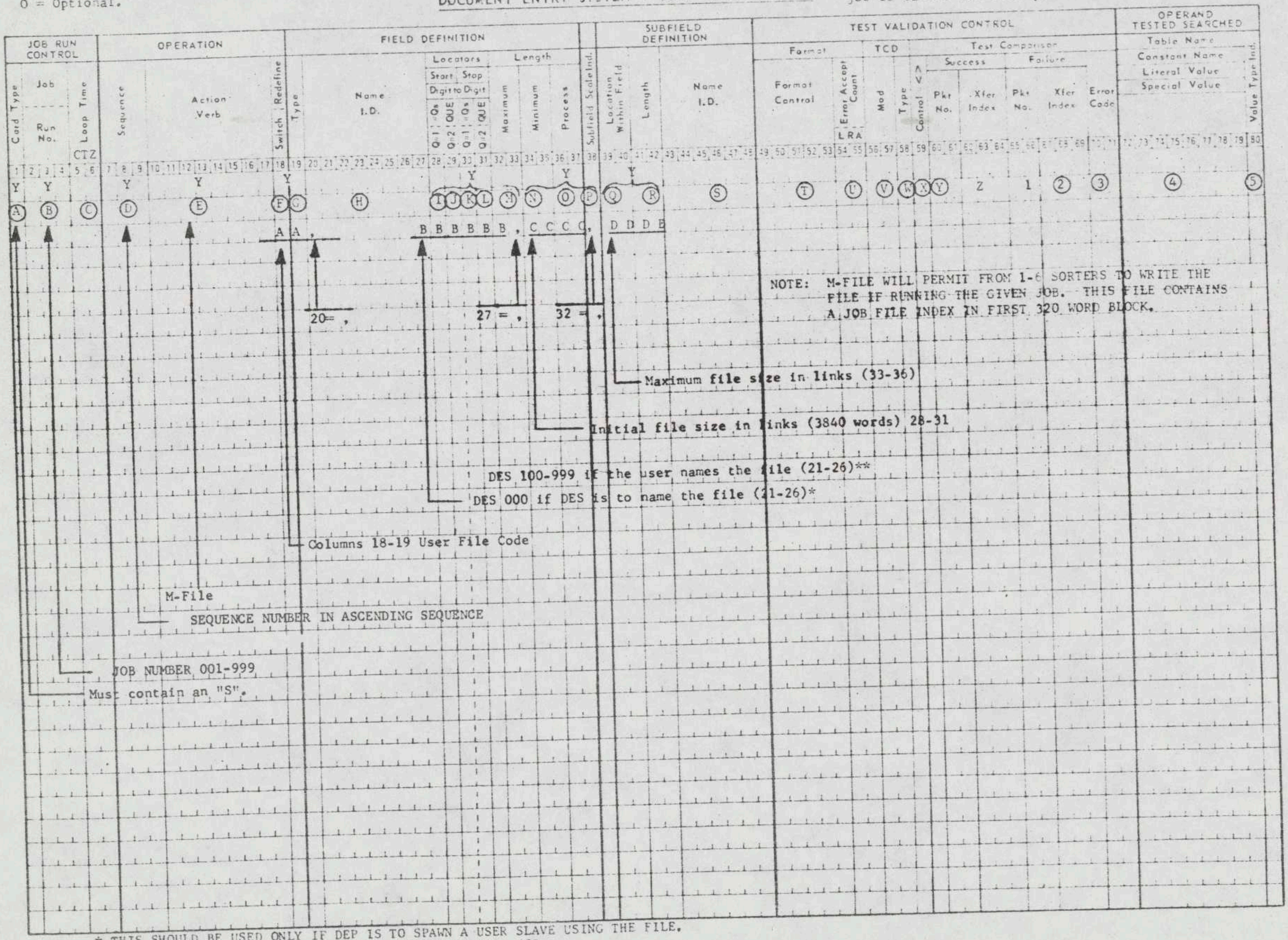
LEGION

Y = Field must be filled in.
O = Optional.

FILE CONTROL VERBS

"M-FILE" - DEP will write a Random Linked File for the given sort run. Only this given Sort Job will be filed to the file. File will be created when job is called for first time.

DOCUMENT ENTRY SYSTEM - SCAN RECORD INPUT



NOTE: M-FILE WILL PERMIT FROM 1-6 SORTERS TO WRITE THE FILE IF RUNNING THE GIVEN JOB. THIS FILE CONTAINS A JOB FILE INDEX IN FIRST 320 WORD BLOCK.

Maximum file size in links (33-36)

Initial file size in links (3840 words) 28-31

DES 100-999 if the user names the file (21-26)**

DES 000 if DES is to name the file (21-26)*

Columns 18-19 User File Code

M-File

SEQUENCE NUMBER IN ASCENDING SEQUENCE

JOB NUMBER 001-999

Must contain an "S".

CE 326

* THIS SHOULD BE USED ONLY IF DEP IS TO SPAWN A USER SLAVE USING THE FILE.
** JOB MAY COME THROUGH THE JOB STREAM OR DEP SPAWN A JOB.

Rev A

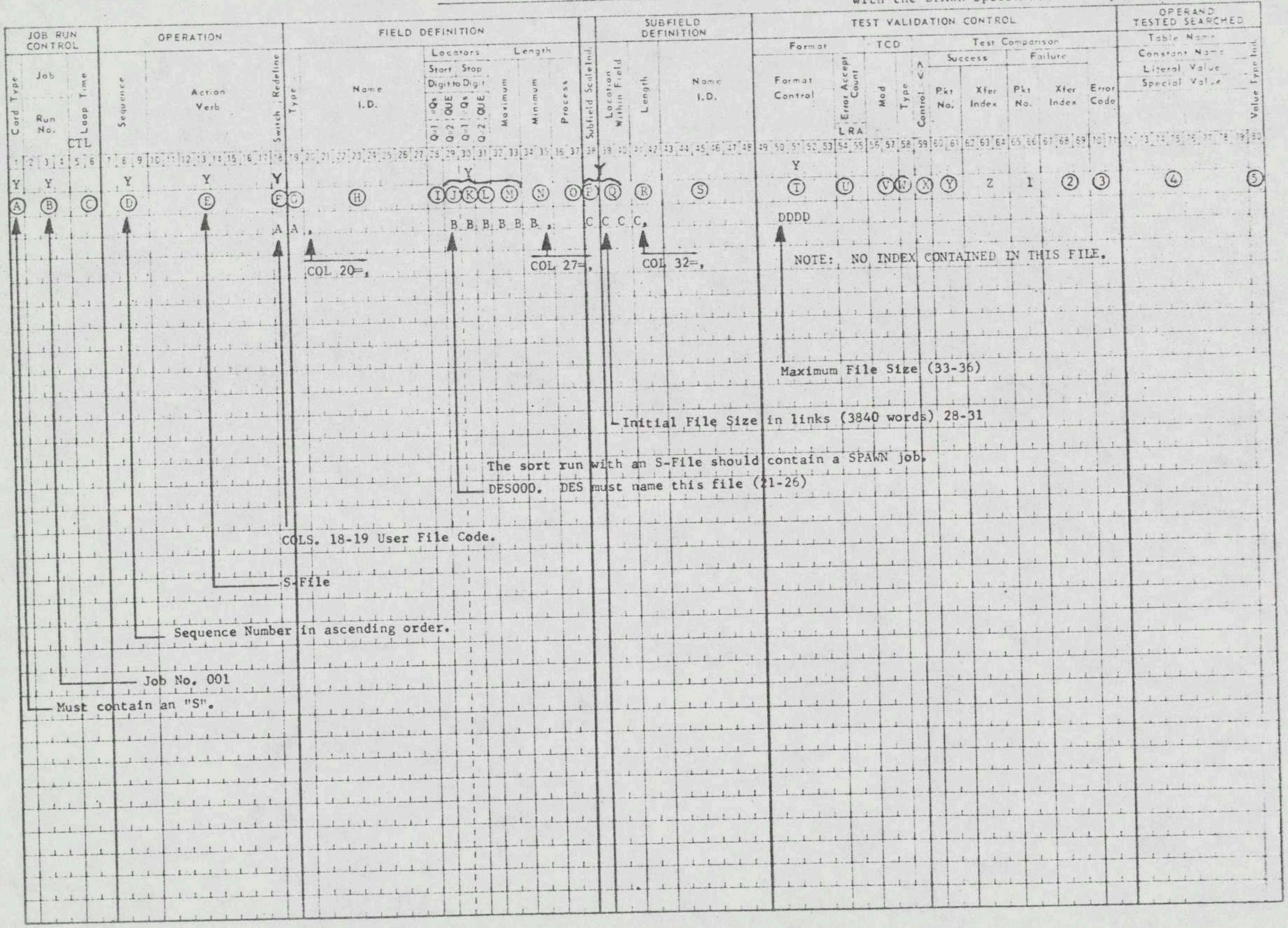
FILE CONTROL VERBS

Legion

Y = Field must be filled in.
O = Optional.

S-FILE - DEP will write a sequential file for this job and sorter. Each time the job is called for a new file will be created and written. This option should only be used with the SPAWN option with multiple copies.

DOCUMENT ENTRY SYSTEM - SCAN RECORD INPUT



COL 20=,

COL 27=,

COL 32=,

NOTE: NO INDEX CONTAINED IN THIS FILE.

Maximum File Size (33-36)

Initial File Size in links (3840 words) 28-31

The sort run with an S-File should contain a SPAWN job.

DES000. DES must name this file (21-26)

COLS. 18-19 User File Code.

S-File

Sequence Number in ascending order.

Job No. 001

Must contain an "S".

CE 326

Honeywell

Rev. A

43A177886

Page 243

LEGION

Y = Field must be filled in.
O = Optional.

FILE CONTROL VERBS

"N-FILE" - DEP will write a random link file for the given sort run. However, other sort jobs may also specify the given file and DEP will write to it. File will be created when first sort job using the file is called.

DOCUMENT ENTRY SYSTEM - SCAN RECORD INPUT

JOB RUN CONTROL			OPERATION			FIELD DEFINITION											SUBFIELD DEFINITION			TEST VALIDATION CONTROL											OPERAND TESTED SEARCHED	
	Job							Locators				Length								Format		TCD			Test Comparison						Table Name	
								Start Digit to		Stop Digit															Success		Failure				Constant Name	
Card Type	Run No.	Loop Time	Sequence	Action Verb	Switch, Redefine	Type	Name I.D.	Q1 Qs	Q2 QUE	Q1 Qs	Q2 QUE	Maximum	Minimum	Process	Subfield Scale Ind.	Location Within Field	Length	Name I.D.	Format Control	Error Accept Count LRA	Mod	Type	Control < >	Pkt No.	Xfer Index	Pkt No.	Xfer Index	Error Code	Literal Value Special Value	Value Type Ind.
1	2 3 4	5 6 CTZ	7 8 9	10 11 12 13 14 15 16 17	18	19	20 21 22 23 24 25 26 27	28	29	30	31	32 33	34 35	36 37	38	39 40	41 42	43 44 45 46 47 48	49 50 51 52 53	54 55	56 57	58	59	60 61	62 63 64	65 66	67 68 69	70 71	72 73 74 75 76 77 78 79	80
Y	Y		Y	Y	Y			Y									Y													
Ⓐ	Ⓑ	Ⓒ	Ⓓ	Ⓔ	Ⓕ	Ⓖ	Ⓗ	Ⓘ	Ⓙ	Ⓚ	Ⓛ	Ⓜ	Ⓝ	Ⓞ	Ⓟ	Ⓠ	Ⓡ	Ⓢ	Ⓣ	Ⓤ	Ⓥ	Ⓦ	Ⓧ	Ⓨ	Z	1	②	③	④	⑤
				A A		,		B	B	B	B	B ,		C C	C	C ,	DD	DD												
						20=,						27=,				32=,														

NOTE: THIS FILE CONTAINS A JOB FILE INDEX IN FIRST 320 WORD BLOCK.

Maximum file size in links (33-36)

Initial file size in links (3840 words) 28-31

User must name N-File see below.

User named file DES100-DES999 (21-36)*

Colums 18-19 User File Code.

N-File

Sequence number in ascending sequence

Job Number 001-999.

Must contain an "S".

*USER MAY ASK DEP TO SPAWN A SLAVE OR USE THE SPAWN CONTROL CARD TO INDICATE A SLAVE JOB WILL COME THROUGH THE JOB STREAM.

CE 326

13.3 SCAN RECORD INPUT DESCRIPTIONS

13.3.1 Job/Run Control

13.3.1.1 CARD TYPE always "S" for SCAN RECORDS, or an * for comments card.

13.3.1.2 JOB/RUN NO. specifies the particular RUN that this SCAN RECORD applies to, 001 - 999.

13.3.1.3 LOOP TIME CONTROL. This will be used by the PSTG to compute the amount of 355 time in MILSECS and Microsecs, required to execute the verbs given within the program loops. This will be implemented at a later date.

13.3.2 Operation

13.3.2.1 SEQUENCE NO. is assigned by the user and must be in order in the source input deck within any given RUN definition. Indicates to the System's Software in what order operations are to be performed.

13.3.2.2 ACTION VERB denotes the particular operation the user wishes to perform on a defined field. There are 17 verbs which may be employed logically by the user to process documents through verifications and sorting (pocketing the documents). ~~For a description of the verbs see Section~~ __________, ~~also for an example of their use see sample program~~.

13.3.2.3 SWITCH NO. There are ten switches available for each job (0-9) that the user may specify for run control purposes. These switches may be set on by the "SETSW" verb or set off by the "RESETSW". The switches are interogated by the "TESTSW" verb. If the switch is on the "TESTSW" - successful. If the switch is off the test - fails. The switch number to be tested with the TESTSW verb is put in field F, "switch no." Initially all switches are set off.

13.3 SCAN RECORD INPUT DESCRIPTIONS, Cont'd.

13.3.3 Operation, Cont'd.

13.3.2.3 EXAMPLE:

Sequence No. 7-9	Action Verb 10-17	Switch No. 18	Success XFER Index 62-64	Failure XFER Index 67-69
010	SETSW	5		
020	TESTSW	5	028	
028	GETFIELD			

In the above, logic switch 5 has been set on at sequence number 010. Since switch 5 is on, and the test is considered successful, a transfer will be made to sequence number 28.

13.3.2.4 REDEFINE allows the user to state a number of conditions; which, if any are met, make the field acceptable. This feature is permissable with the following verbs:

- GETFIELD
- CKFORMAT
- VALIDATE

EXAMPLE:

Assume the dollar field is the sought field on the document. It is ten digits in length (numeric) and normally defined by a beginning dollar queue($) and an ending dollar queue ($). However, it is desired to accept the field under the following deviations from normal.

(a) The beginning dollar queue ($) is present followed by ten numerics but an ending account queue (A).

(b) The beginning dollar queue is present followed by ten numerics but an ending error signal which is pound sign #.

(c) The beginning dollar queue ($) is missing follwed by ten numeric and an ending queue ($).

First, using the GETFIELD and the CKFORMAT verbs, the coding is as follows:

13.3 SCAN RECORD INPUT DESCRIPTIONS, Cont'd.

13.3.3 Operation, Cont'd.

Seq. No. 7-9	Action Verb 10-17	Redefine 18	Type 19	Name ID 20-27	Start Q1 Q2 28-29	Stop Q1 Q2 30-21	Length Max. 32-33	Length Proc. 36-37
005	GETFIELD	1	N	Dollar	$	$		10
005	GETFIELD	2	N	Dollar	$	A		10
005	GETFIELD	3	N	Dollar	$	#		10
005	GETFIELD	4	N	Dollar	0	$		10
*006	CKFORMAT	1	N	Dollar				10

In the above example the queue arrangement of the field called "Dollar" is given in four acceptable forms. If any one of the four are found, the field is considered valid by the queue arrangement. Next the CKFORMAT will determine if the 10 digits found between the queues are all numeric.

Use of the "VALIDATE" verb combines the GETFIELD and the CKFORMAT verbs. Thus, it is only necessary to do the following.

*The queue symbols need to be given only for the GETFIELD and the VALIDATE verbs.

Seq. No. 7-9	Action Verb 10-17	Redefine 18	Type 19	Name ID 20-27	Start Q1 Q2 28-29	Stop Q1 Q2 30-31	Length Max. 32-33	Length Proc. 36-37
005	VALIDATE	1	N	Dollar	$	$	10	
005	VALIDATE	2	N	Dollar	$	A	10	
005	VALIDATE	3	N	Dollar	$	#	10	
005	VALIDATE	4	N	Dollar	0	$	10	

13.3.4 Field Definition

13.3.4.1 TYPE is normally numeric (N) or numeric with dashes (D). However, for the DRD200 where a plus or minus may be in a field as acceptable data, then an (X) is permissable.

N = numeric field
D = numeric field with dashes
X = numeric field with plus or minus sign and dashes

Field type must always be indicated when using the "GETFIELD", "VALIDATE", "GETSUBF", and "CKFORMAT" verbs.

13.3 SCAN RECORD INPUT DESCRIPTIONS, Cont'd.

13.3.4 Field Definition, Cont'd.

13.3.4.2 NAME I.D. User supplied 8 character alpha numeric name for this field. All names must be unique.

13.3.4.3 LOCATORS are the queues that define the start and the end of the given field. When one queue defines the start of a field and a second queue defines the end of a queue (which is the normal case); the BCD symbols that represent the octal value of the queues must be placed in fields I and I. When there are two queues, back to back, which define the start and/or the end of the field; the second defining queue must be given in field J and/or L. Assume a field which is eight digits in length. It is defined by a beginning and ending queue which has an octal value of 21. The coding will be as follows:

		Locators				
Action	Name	Start	Stop	Field Length		
Verb	ID	Q1 Q2	Q1 Q2	Max.	Min.	Proc.
10-17	20-27	28-29	30-31	32-33	34-35	36-37
GETFIELD	FIELD	A	A			08

Since an A is the BCD equivalent of octal 21; it is placed in the start and stop queue field (fields I and K). There is only one starting queue and one ending queue so J and L are blank (29 and 31). Only one length is acceptable, eight, so minimum and maximum length do not apply, only process lengths.

Assume the field is defined, start and end, by two queues back to back and the queue values, in octal, are 21 and 22. The coding 2ill be as follows:

		Locators				
Action	Name	Start	Stop	Field Length		
Verb	ID	Q1 Q2	Q1 Q2	Max.	Min.	Proc.
10-17	20-27	28-29	30-31	32-33	34-35	36-37
GETFIELD	FIELD	A B	A B			08

Since B is the BCD equivalent of octal 22, it is placed in second start and stop queue field position (J and L).

13.3 SCAN RECORD INPUT DESCRIPTIONS, Cont'd.

13.3.4 Field Definition, Cont'd.

13.3.4.3 Continued

In the event that there are multiple fields on a document that are defined by identical queues; fields J and K may be used to give the queue count which identifies the sought field. Assume that the document has several fields on it which are defined by dashes as follows:

--Field 1--Field 2--Field 3--Field 4--Field 5--

The sought field is field 3 which lies between the third and fourth dash.

		Locators				
Action	Name	Start	Stop		Field Length	
Verb	ID	Q1 Q2	Q1 Q2	Max.	Min.	Proc.
10-17	20-27	28-29	30-31	32-33	34-35	36-37
GETFIELD	FIELD 3	__ 3	__ 4			08

In the above case fields J and L of the scan record identify the sought field by giving the queue count. A maximum queue count of ten is permitted, 1-9 and 0 which is equal to 10.

13.3.4.4 MAX. LENGTH AND MIN. LENGTH are used to seek a given field which may vary in length due to zero suppression or because it is not properly queued. In the case of improper defining queues; generally, the queue, either starting or ending, may not be present or has been mis-read. The user may provide for the conditions through the use of redefines, already explained, and giving an acceptable minimum and maximum field length. Whenever maximum and minimum field length is given; the maximum lenght right justified will be used for all subsequent processing. Therefore, if the field should be less than the tiven maximum length scaling, either left or right, must be indicated in fields P and R.

Scaling is accomplished by an "L" for Left scaling or/and "R" for right scaling field P and giving the digit count to be retained, left or right, in field R.

13.3 SCAN RECORD INPUT DESCRIPTIONS, Cont'd.

13.3.4 Field Definition, Cont'd.

13.3.4.4 Continued

Assume two cases. The first case; the sought field may vary from five to eight digits in length. Proper field size is considered eight digits and no scaling is necessary. In case two, the sought field may vary from five to ten digits in length but proper lengths is the five left hand digits in all cases.

				Locators		Length		Scale		
Action Verb 10-17	Redefine 18	Type 19	Name ID 20-27	Start Q1 Q2 28-29	Stop Q1 Q2 30-31	Max. 32-33	Min. 34-35	Proc. 36-37	Ind. 38	Length 41-42
Case 1										
GETFIELD	0	N	FIELD	A	A	08	05			
Case 2										
GETFIELD	0	N	FIELD	D	C	10	05		L	05

13.3.4.5 PROCESS LENGTH should be indicated for all fields which are fixed in lenght and no variables are anticipated such as zero suppression, etc. When process length is used the field must contain the stated digit count, exactly, or it will be considered in error. Process Length cannot be used if the maximum and minimum count fields are used.

13.3.4.6 SCALE INDICATOR is used in combination with fields M and N, maximum and minimum field length. This permits the user to scale off the number of desired digits when the field length may be between the minimum and the maximum count.

The two indicators permitted for scaling are: L for left hand scaling and R for right hand scaling. The number of left or right hand digits to be retained must be given in field R.

13.3 SCAN RECORD INPUT DESCRIPTIONS, Cont'd.

13.3.4 Field Description, Cont'd.

13.3.4.6 Continued

Assume a field which has a minimum acceptable digit count of 05 and a maximum acceptable digit count of 10. The desired field length consists of the right hand fine digits. The coding would be as follows:

Length		Scale	
Max.	Min.	Ind.	Length
32-33	34-35	38	41-42
10	05	R	05

Also see examples under M and N.

SUBFIELD is used to define a second or third **field** that may be embedded within a field defined by queues. This field will contain an "S" to indicate a subfield exists within the queue field.

Assume a major field, field defined by starting and ending queue, which consists of 12 digits. A subfield is embedded within the field consisting of digits 5-9. The coding would be as follows:

Action Verb 10-17	Name ID (Major Field) 20-27	Sub Field IND 38	Location of SUBF (Within Major Field) 39-40	Length Subfile 41-42	Name ID (Subfield) 43-48
*GETSUBF	BIGFIELD	S	05	05	SMALL F

*The queues do not have to be given a "GETFIELD" for the BIGFIELD must have been given previous to the GETSUBF.

13.3 SCAN RECORD INPUT DESCRIPTIONS, Cont'd.

13.3.4 Field Description, Cont'd.

13.3.4.7 LOCATION WITHIN FIELD. User indicates digit position in the field where the subfield begins. The subfield location must be given within the field from left to right.

Assume a subfield exists within the account field. The account field is twelve digits in length and the subfield is the high order from digits, (i.e. beginning at location 01 from left to right.)

Example: Queue 1234--6789786Queue

In the above example the subfield consists of 1234 and would be defined as starting in location 01 and beginning 04 digits long. For more information see example under

13.3.4.8 LENGTH - character length of subfield.

13.3.4.9 NAME I.D. - User may specify up to a 6 alpha numeric character subfield name. All names must be unique within a given document type.

13.3.5 Test/Validation Control

13.3.5.1 Format

o FORMAT CONTROL allows the user to specify dashes within a field and their acceptable locations within the field based on digit counts before and after dashes. In addition the user may indicate through the case of the "error accept counts" accept an error condition in place of a dash, or dashes. The user may indicate up to and including four dashes and nine digits between dashes.

Assume a field of twelve digits in length which contains three dashes as follows:

999-999-99-9

13.3 SCAN RECORD INPUT DESCRIPTIONS, Cont'd.

13.3.5 Test/Validation Control, Cont'd.

13.3.5.1 Format, Cont'd.

- The coding to test for the dashes would be as follows:

Action Verb 10-17	None ID 20-27	Format Control 49-53
CKFORMAT	Big Field	33 21

In the above example the user has indicated to the program to look for three digits a dash, three more digits and a dash followed by two digits a dash and one digit.

Assume the same field in the above example except for zero suppression. Then the field could possibly take the following form:

999-999-99-9

9 - 9 -9-9

Therefore, any combination between 999-999-99-9 and 9-9-9-9 is considered acceptable. The coding would be as follows to define this condition.

Action Verb 10-17	Name ID 20-27	Redefine 18	Format Control 49-53
CKFORMAT	Field	0	33 21
CKFORMAT	Field	1	11 11

In the above example, any combination of digits before and after dashes that fall between 3 and 1, 3 and 1, 2 and 1 and 1 and 1 will be accepted.

13.3 SCAN RECORD INPUT DESCRIPTIONS, Cont'd.

13.3.5 Test/Validation Control, Cont'd.

13.3.5.1 Format, Cont'd.

- ERROR ACCEPT COUNT permits the user to accept errors in place of dashes. This is useful when it is known that a dash should occur in a given position and, assuming the digit count was correct, it is desired to accept any non-numeric, normally an error signal, character in its place. One, two, three, or four error conditions may be specified in place of a dash or dashes. The following values must be used to specify that an error signal or condition is to be accepted in place of a dash.

  Accept error signal in place of 1st dash = 01

  Accept error signal in place of 2nd dash = 02

  Accept error signal in place of 3rd dash = 04

  Accept error signal in place of 3rd dash = 08

  Whenever two or more dashes are to be accepted in place of an error signal; the given values must be added together to represent the desired combination.

  Assume the first and third dash positions may contain error signals; then 01 and 04 or 05 would be coded in this field.

  Assume the field 999-999-99-9 and the first and third dash positions may contain error signals. The coding would be as follows:

13.3 SCAN RECORD INPUT DESCRIPTIONS, Cont'd.

13.3.5 Test/Validation Control, Cont'd.

13.3.5.1 Format, Cont'd.

Action Verb 10-17	Format Control 49-53	Error Accept Count 54-55
*CKFORMAT	33 21	05

*The "VALIDATE" verb may also be used with the format control logic.

13.3.5.2 TCD

- MOD - there are 62 TCD modes (02-63) which the system will handle automatically. When the user wishes to use a Mod 10 or Mod 11 TCD this field must contain the desired MODE, I.E., 10 or 11, or any value 02-63. The user may also specify a special TCD type, see below under
- TYPE is used in conjunction with Mod to indicate the desired TCD type. There are three TCD types that the system will automatically perform. There are Natural, Geometric and Alternate 2's. Along with the Mod, 02-63, the user must specify TCD type. The TCD type codes are as follows:

  Natural = N

  Geometric = G

  Alternates 2's = A

  *Other = T

  *When the user must use a TCD other than those given above, he must prepare a TCD table giving the Table Name, the divisor, remainder and multipliers. This is outlined under the General Tables User Input Form. The Type in this case, must contain a "T" and the MOD field will be blank.

13.3 SCAN RECORD INPUT DESCRIPTIONS, Cont'd.

13.3.5 Test/Validation Control, Cont'd.

13.3.5.2 TCD, Cont'd.

- For the three types that the system will perform, the coding will be as follows:

TCD

MOD 56-57	TYPE 58

Div. 02-63

N = Natural

G = Geometric

A = Alternate 2's

Example:

TCD

MOD 56-57	TYPE 58	Table Name 72-80
	T	TCD TABLE*

*In the above example, the T indicates a non-standard TCD which is located in TCD table. See general tables for TCD formats.

All standard TCD types assume a field with a format which the right most character contains the TCD digit. Dashes and non-numeric within the TCD field will be assumed as zeroes.

13.3.5.3 Test Comparison

- CONTROL indicates which type of comparison is to be made, an =, <, > or ≠ comparing the contents of the indicated field to the constant indicated in 4, the operand field. When the test is equal to the given control, =, <, >, ≠, it is considered not equal the given control, =, ≠, <, >, it is considered a failure.

13.3 SCAN RECORD INPUT DESCRIPTIONS, Cont'd.

13.3.5 Test/Validation Control, Cont'd.

13.3.5.3 Test Comparison, Cont'd.

Examples:

Seq. No. 7-9	Action Verb 10-17	Name ID 20-27	Control 5-9	Success XFER 62-64	Failure XFER 67-69	Operand 72-79 HI Dollar	Type 80 K
010	TEST	Dollar			050	HI Dollar	K

HI Dollar = 0000500000

050

In the above example the dollar field is being compared to $5000.00. The comparison is for a greater than condition. If a dollar field is greater than $5000.00, the test is considered successful, as given under control, and the next sequential verb will be executed. If the dollar field is less than $5000.00, the transfer on failure will be mode to verb #50.

13.3.5.4 Success

- PKT. NO. - Pocket number for the document if the test is successful. The pocket number may or may not be specified. Pocket numbers from 00-31 depending on the sorter may be specified and three special control documents M, N, I. An M or N type document may be specified for seperators such as block or batch types where a seperator is required in all pockets in use on the sorter. An I type document may be specified for a seperator, such as a bill seperator, which will be pocketed depending on a count of items.

  If both a pocket number and a transfer index is indicated, the XFER index will take precedence, but if no pocket tests or searches, the item will go to the pocket no. indicated in this scan record.

13.3 SCAN RECORD INPUT DESCRIPTIONS, Cont'd.

13.3.5 Test/Validation Control, Cont'd.

13.3.5.4 Success, Cont'd.

Examples:

			Success	Failure			Operand	
Action Verb 10-17	Name ID 20-27	Control 59	PKT No. 60-61	Xfer 62-64	Pkt. No. 65-66	XFER 67-69	Name 72-79	Type 80
TEST	FIELD 1	-	05	026	07		CONSTANT	k

In the above example Field 1 is being tested to "constant" for an equal condition. If an equal results from the comparison a transfer will be made to verb number 025 with pocket 05 being the select pocket unless a subsequent verb changes the pocket number. If the comparison is not equal, the next sequential verb will be executed with pocket 07 being the select pocket unless a subsequent verb changes the pocket number.

			Success				Operand	
Action Verb 10-17	Name ID 20-27	Control 59	PKT. No. 60-61	XFER 62-64	PKT. NO. 65-66	XFER 67-69	Name 72-79	Type 80
TEST	FIELD 2	=			07		Constant	K

In the above example no XFER is set for success or failure. Therefore, if the tests fails testing will be terminated and the document put in the failure pocket 07. If the test results in an equal condition, the next sequential verb will be executed.

- XFER INDEX (Success) - This field will contain the sequence number (verb) to which a transfer is to be made if the test succeeds (see first example under 13.3.5.4). This field may be left blank - if so - the next sequential verb will be executed.

13.3.5.5 Failure

- PKT NO. - Test failure pocket number. This may be any value from 00-31 depending on the sorter type as well as the special document types: M - N or I (See 13.3.5.4).

13.3 SCAN RECORD INPUT DESCRIPTIONS, Cont'd.

13.3.5 Test/Validation Control, Cont'd.

13.3.5.5 Failure

- XFER INDEX - This field will contain the sequence number (verb) to which a transfer is to be made if the test fails. This field may be left blank - if so - testing will terminate and the document pocketed in the error or reject pocket; unless, the XFER on success is set to a transfer. In this case, the next sequential instruction will be executed (see examples under 13.3.5.5)
- ERROR CODE - User supplied error code when failure PKT. is specified - 01-62.

13.3.6 Operand Tested/Searched

13.3.6.1 Table Name - Name of Table to be searched with "SEARCH" or TEST verb.

13.3.6.2 Constant Name - Name of constant to compare contents of field to.

13.3.6.3 Literal Constant - Literal value to be compared to.

13.3.6.4 Value Type Indicator - Indicates the type of value entered in the Operand:

T - Table Name

V - Literal Constant

K - Constant Name

F - Field to Field Compare

13.4 GENERAL TABLES USER INPUT FORM

13.4.1 Ident

The ident will be preprinted to "GT", which indicates the input is a general table.

13.4.2 Run Number

The run number or job number for which the table is used must be given in this field.

13.4.3 Sequence Number

Sequence numbers may start with 0001 and extend to 9999. Each entry should contain a sequence number and must be in ascending order.

13.4.4 Table Name

The table name must appear on the first entry. Subsequent entries in the same table need not contain the name.

13.4.5 Sort Indicator

An "S" must be placed in this field if the table is a sort table, i.e., will be used to select packets for documents. If the table or data is not for sorting purposes, this field will be left blank.

If an S is placed in this field, field N must contain the packet number.

13.4.6 Table Type

There are six table types that the generator will recognize and one of the names listed below must be placed in this field.

TCDT: Transposition Check Digit Table

If the user specified a TCD other than G10, G11, N10, N11 or alternate two's, this field must contain TCDT.

TRLT: Transaction Code Transliteration Table

This field must contain TRLT if a transliteration table is to be generated.

13.4 GENERAL TABLES USER INPUT FORM, Cont'd.

13.4.6 Table Type, Cont'd.

TIME: Time Table

The Time Table is used along with the Master Transit Table Send Points--Collection Time Codes. The time table may consist of 24 entries 01-24. The codes must be given in 26-27 and the time of day in 28-35.

PKTT: Pocket Table

The user may construct his own pocket table given item counts beginning column 28 and pocket number and type in fileds M and N.

MISC: Miscellaneous Table

This applies to all tables, either sort or non-sort, which contsist of an argument only or an argument and a pocket number.

CNST: Constant

This indicates the table is a constant only.

13.4.7 TCD ID

This field is used in conjunction with field F. If field F contains TCDT, then this field must contain TCD identification number 0 - 9. This identification number must be referred to, for special TCD, in field V of the scan record input formats by a "T" followed by the ID number 0 - 9.

13.4.8 Length of Argument in Characters

When a single argument in given, the length of it must be specified in characters in this field. If this field contains a comma in column 24, the field length will be determined by the number of digits between commas in the data entered in columns 28 - 76.

13.4.9 TCD Division or Time Codes

Depending on field F (TCDT or TIME) this field will contain the TCD divisor 01-99 or the time of day codes of the times appearing in fields

13.4 GENERAL TABLES USER INPUT FORM, Cont'd.

13.4.10-11 Collection Times

These fields are used in conjunction with fields F and I. If F contains TIME, and I a code 01-24, then the times of day 0000 - 2400 must be entered in these fields.

13.4.12 Table Data

Other than J and K above, table data will begin in column 28 and may continue through 76. A single argument may be given with the character count in field H., or multiple arguments may be given, separated by dashes through column 76.

13.4.13 Pocket Type

This field used in conjunction with Field F. If field F contains PKTT, then pocket type may contain the following codes:

K = Kill pocket

R = Rerun pocket

O = On us pocket

13.4.14 Pocket Number

This field contains pocket number for the sort table or pocket table. It is used in conjunction with fields E and F.

13.4.15 Alternate Pocket

An Alternate pocket, 1-9, for the prime pocket in field N may be given here. If no alternate pocket, this field may be left blank.

REV. A

GENERAL TABLES USER INPUT FORM - DES 6000

USER INPUT

										TABLE DATA 28 →					
										COLLECTION TIMES TIME OF DAY					
	IDENT	RUN NO	SEQ. NUMBER	TABLE NAME	SORT IND.	TABLE TYPE	TCD ID NO	LENGTH OF ARG. IN CHARS.	TCD DIV OR TIME CODES 01-24	FROM 00 2400	TO 00 2400		PKT TYPE	POCKET NO.	ALT PKT
COLS	1-2	3-5	6-9	10-17	18	19-22	23	24-25	26-27	28-31	32-35	36 76	77	78-79	80
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O

4DA17/500

Cont. on Page 264 Page 2

13.5 MASTER TRANSIT TABLE USER INPUT FORM

13.5.1 Master Ident

Identifies the table as a master table. The first digit must be "M". The next two digits may be any number or letter.

13.5.2 Sequence Number

Sequence numbers may start with 6001 and extend to 9999. Each entry should contain a sequence number and must be in ascending sequence.

13.5.3,4,5 FRB - ABA

The Federal Reserve District, City State and American Banking Association codes must be provided by the users in this field. The user may give the FRB only or the FRB and City State or the FRB, City State, and the ABA number.

13.5.6 Multiple ABA's

When the user wishes to tie several ABA's to the same send point and/or pocket number, an asterisk must be placed in this field. Succeeding ABA's can then be placed in columns 17-80 and separated by dashes. However, if this is utilized, a previous line must have defined all other required control input, such as run number, send point, pocket number, etc.

13.5.7 Prime Run Number

This field must contain the run number of the first run (first sort pass or prime run) which requires the table for sorting.

13.5.8 Prime Pocket Number

The pocket number which is to be used on the first run (prime run) is entered here. This corresponds to the run number entered in field G. Depending on the sort in use, up to 32 pockets, 1 - 32, may be indicated.

13.5.9 Alternate Pocket Number

An alternate pocket, 1 - 9, may be entered here for the prime run. This field is used in conjunction with field G and H. If no alternate pocket, this field will be left blank.

13.5 MASTER TRANSIT TABLE USER INPUT FORM

13.5.10 Send Point Code

The send point code may be any value from 000 - 999, or if no send point code, this field may be left blank. However, if sorting is to take place by send point, this field must contain a send point code.

13.5.11 Float

The float code, or number of days float, for each send point may be entered here. If no float, this field may be left blank.

13.5.12 Secondary Run Number

This field may contain the run number of the second run (sort pass) which requires the table for sorting.

13.5.13 Pocket Number

The pocket number which is to be used on the second run is entered here. Depending on the sorter in use, up to 32 pockets 1-32 may be indicated.

13.5.14 Alternate Pocket Number

An alternate pocket, 1-9, may be entered here for the second run. If no alternate pocket, this field will be left blank.

13.5.15,16 and 17 Same as 13.5.12, 13, and 14 for the third run, which requires the table.

13.5.18,19 and 20 Same as 13.5.12, 13, and 14 for the fourth run, which requires the table.

13.5.21,22 and 23 Same as 13.5.12, 13, and 14 for the fifth run, which requires the table.

13.5.24 Big Items - Amount

The amount in hundreds, thousands, and ten thousands which is to be considered a big item amount for the ABA's or send point may be entered here. If no big item amount, this field will be left blank.

13.5 MASTER TRANSIT TABLE USER INPUT FORM

13.5.25 Big Item Pocket Number

The pocket number to which big items are to be selected may be entered here. Items which are equal to or greater than the amount given in field Y will be selected in this pocket.

13.5.26 Send Points - Collection Time Codes

There are twenty-four time periods which the user may designate as valid for items going to the various send points. The user may indicate in this field which time periods are valid for the send point given in field J. If at the time of processing the time of day falls within the designated times, the document will be pocketed in the normal pocket. If the time of day is outside of the range of times given in the time table, the document will be tested for a big dollar item. If the document is a big dollar item, it will be selected to the big item pocket. When no big item value or pocket is given, documents will be selected to their normal pockets.

The user must provide the corresponding collection time table if he chooses to use this feature. See the Miscellaneous Tables Input writeup fields, F, I, J and K.

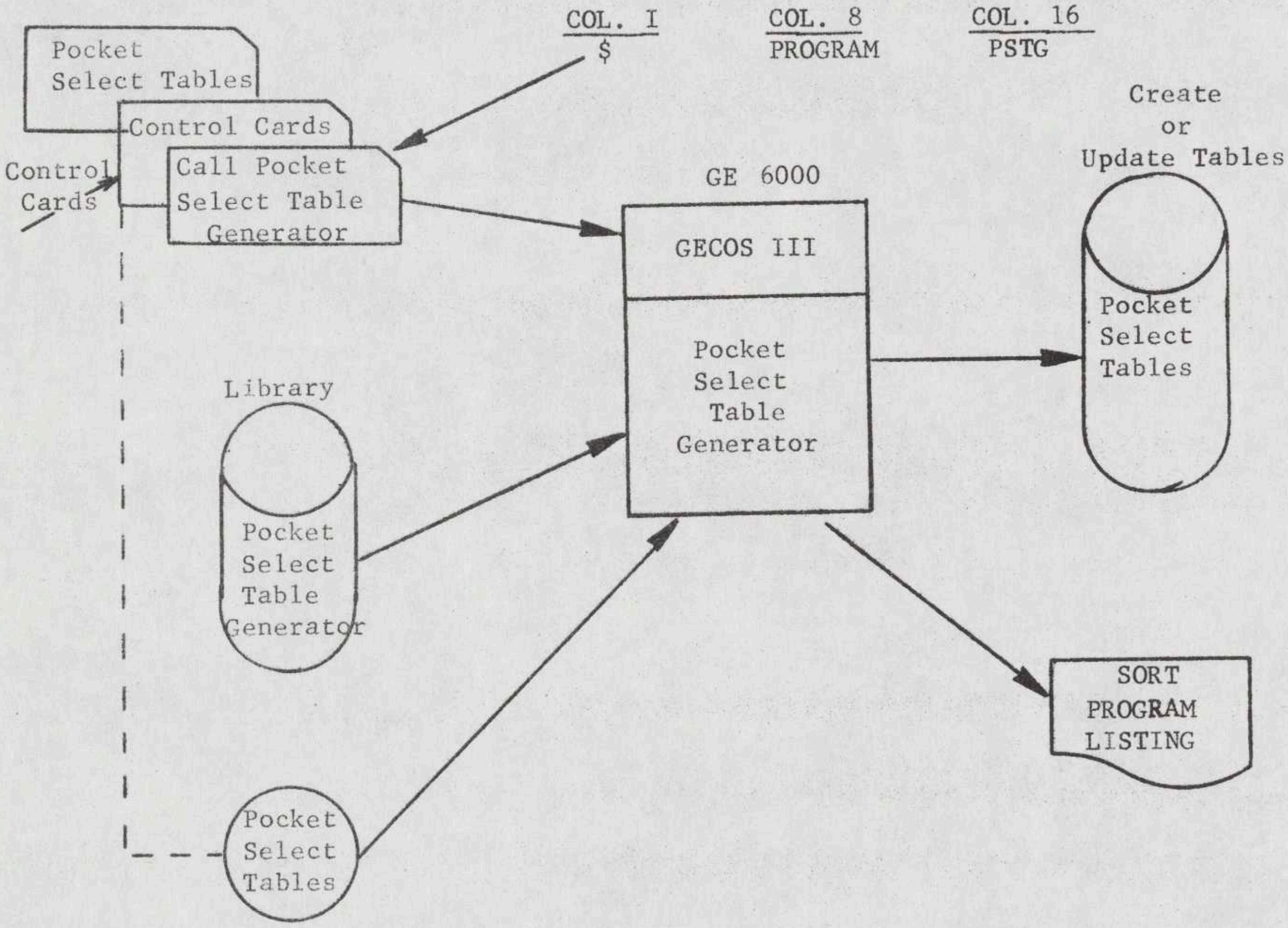
REV.A

Cont. on Page 268

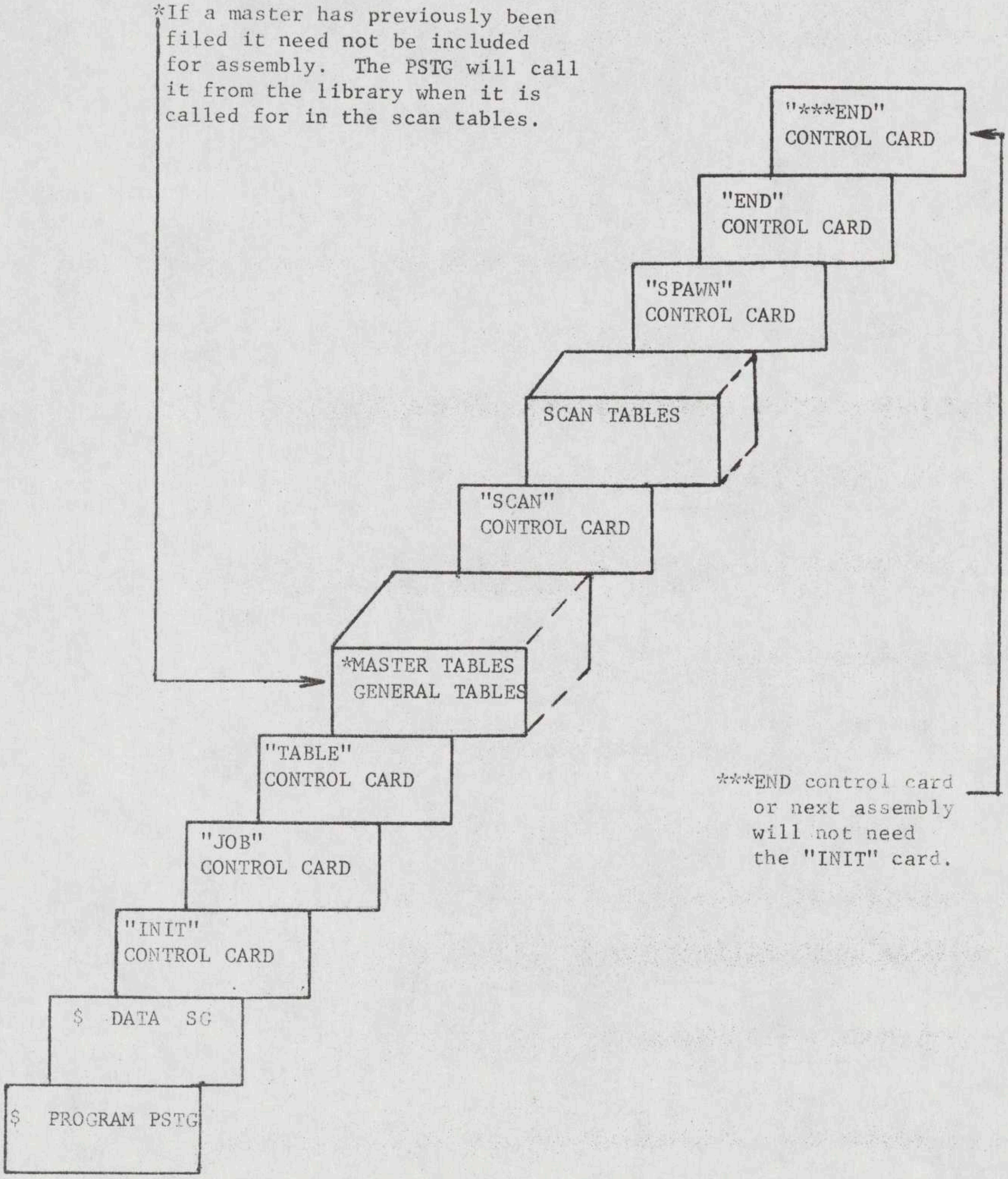
USER INPUT 6000 MASTER TRANSIT TABLE USER INPUT FORM DES 6000

			CARD COLS	
		MASTER IDENT.	1-3	A
		SEQ. NUMBER	4-7	B
FRB-ABA	FRB	DIST	8-9	C
		CITY STATE	10-11	D
	ABA		12-15	E
		MULTIPLE ABA'S	16	F
		PRIME RUN NUMBER	17-19	G
		PRIME POCKET NO.	20-21	H
			22	I
		SEND PT. CODE	23-25	J
		FLOAT	26-27	K
SECONDARY RUN NO'S. AND POCKETS	1	RUN NO.	28-30	L
		PRT NO.	31-32	M
			33	N
	2	RUN NO.	34-36	O
		PKT NO.	37-38	P
			39	Q
	3	RUN NO.	40-42	R
		PKT NO	43-44	S
			45	T
	4	RUN NO.	46-48	U
		PKT NO.	49-50	V
			51	W
BIG ITEMS	AMT.	10,000	52	
		1,000	53	X
		100	54	Y
		PKT NO.	55-56	
SEND POINTS - COLLECTION TIME CODES		T 1	57	
		T 2	58	
		T 3	59	
		T 4	60	
		T 5	61	
		T 6	62	
		T 7	63	
		T 8	64	
		T 9	65	
		T 10	66	
		T 11	67	Z
		T 12	68	
		T 13	69	
		T 14	70	
		T 15	71	
		T 16	72	
		T 17	73	
		T 18	74	
		T 19	75	
		T 20	76	
		T 21	77	
		T 22	78	
		T 23	79	
		T 24	80	

LOADING AND RUNNING THE POCKET SELECT GENERATOR PROGRAM



13.7 PSTG ASSEMBLY DECK SETUP



*If a master has previously been filed it need not be included for assembly. The PSTG will call it from the library when it is called for in the scan tables.

***END control card or next assembly will not need the "INIT" card.

13.8 PSTG CONTROL CARDS

There will be eight input control cards to the PSTG. These control cards control the assembly of scan records, tables, and assembly updates. Their description and use are as follows:

- INIT

1 - 6	42-47
"INIT"	"DES6000"

Signals the PSTG to initialize for a 6000 assembly.

- JOB

1 - 6	8 - 10	13-24
"JOB"	Job No. 001-999, or alfa or alfa-numeric job numbers	"DATE"

Provides the generator with the library identification number for filing the assembly. The TTY operator will also use this number to call the Job into execution.

- TABLE

1 - 6	13-24
"TABLE"	"ASSEMBLE" or "UPDATE"

Signals the generator that a Table, either general type or master type, is to be assembled or updated. The ADD and DEL control cards are used with the "UPDATE" control word.

13.8 PSTG CONTROL CARDS, Cont'd.

- SCAN

1 - 6	13-24
"SCAN"	"ASSEMBLE" or "UPDATE"

Signals the generator that SCAN tables follow and that either an "assembly" or assembly "update" is to be processed. The ADD and DEL cards are used with the "update" control word/

- SPAWN

1 - 6	12 - 18	20	24	26	30-37
SPAWN	IDENT	SYSTEM CONTROL	COPIES	SPAWN CODE	CONDITION CODE

Signals the generator that a user 6000 slave is to be placed in the library for the DEP to spawn, or the slave job whose IDENT is given will come through the job stream and use the sorted file or files created by the sort job. The object deck, for spawing must be set up for running under "Spawn" as outlined in GECOS manual CPB 1518 B.

IDENT: Unique number of (alfa or alfa numeric) which identifes the 6000 slave job. This identifier will be put in the DEP slave control stack -- see slave job management block.

SYSTEM CONTROL: used to indicate respawning jobs after system disaster; 1 = respawn
2 = no respawn

COPIES: Single or multiple copies of the job may or may not be spawned;

S = Single only
M = Multiple
*N = Null - no spawn identifies a slave job which will come through the job stream.

13.8 PSTG CONTROL CARDS, Cont'd.

SPAWN CODE: If multiple copies are to be spawned, "M" in Col. 24, give condition for spawning multiple copies:

1 = one copy for each sorter running the sort job.
2 = one copy each time the job is called from a sorter, i.e., each time the job is started.

CONDITION CODE: The condition under which this job is to be spawned.

- "CHECKPT" - spawn as soon as there is a sort checkpoint.
- "IMMED" - spawn as soon as the sort job starts.
- "ENDJOB" - spawn at the end of the sort job.
- Spawn this job on a document count. 00000001-99999999.

*N -- The job is not to be spawned, i.e., no object deck follows the SPAWN card. The SPAWN card in this case identifies a slave program that will access DEP created files, but will come through the normal job stream.

Note: There may be more than one "spawn" card for NULL, "N" "spawn" jobs.

13.8 PSTG CONTROL CARDS, Cont'd.

- DEL

1 - 6	8 - 13	15 - 20
DEL	Starting Seq. No. to be deleted	Ending Seq. No. to be deleted

The "DEL" control card signals the generator to delete the given sequence numbers. The "DEL" card may be followed by sequence cards replacing those numbers deleted and cards to be merged in (added) to the source program. The DEL control card must follow either a TABLE or SCAN Control card. The cards to be deleted, added, or changed must be punched in standard scan or table form - see input forms.

- ADD

1 - 6	8 - 13
	Starting Sequence No. where following cards are to be added.

The "ADD" control card signals the generator to add the following cards to the source deck after the number given in 8-13 in the ADD card. The ADD control card must follow a "TABLE" or "SCAN" control card. Cards to be added must be punched in standard scan or table form - see input forms.

- END

1 - 6
"END"

Signals the generator the final source card for the given assembly (job) has been read.

13.8 PSTG CONTROL CARDS, Cont'd.

- ***END

1 - 6

***END

Signals the generator that all source jobs for assembly in the stack have been read in.

1 -	8 - 14	16
$	PROGRAM	PSTG

First card of deck which calls PSTG into core.

1	8 - 14	16
$	DATA	SG

Second card of deck setup preceding PSTG control cards and source program.

13.9 DES FILE SYSTEM

The DES file system will be PERM file system named "DES", i.e., the User Master Catalog name. The DES will contain six major sub catalogs. Such Catalogs will be named as follows:

13.9.1* DESCMM - DEP to slave communications.

13.9.2** DESCMN - Slave to DEP communications.

13.9.3 SPNXXX - These files will contain all user slaves that the DEP will spawn. XXX will be numeric 001-999. The PSTG will create and write these files. The DEP will read these files with MME GENEWS.

13.9.4 TABXXX - These files will contain the assembled sort programs, with the exception of master tables. Master Tables will be filed on the MSTXXX files. The PSTG will create and write the assembled scan records and general tables on these files. XXX will be the user's sort job number, i.e., TAB001-TAB999. The DEP will read these files whenever a sort job is called for from the TTY's.

13.9.5 MSTXXX - These files will contain the sort jobs master tables. The PSTG will create and write the assembled master tables on these files. XXX will be the user master table name, i.e., 01-99. The DEP will read these files when cross referenced from the TABXXX files, and include the master table in the requested sort job.

13.9.6 DESXXX - The DEP will create and write sorted data to these files. The size and name of these files will be provided on the TABXXX files as part of the user's sort job parameters. The DEP will also release these files when requested by the user.

Other than DESCMM and DESCMN the subfiles will be created and released dynamically through the use of a DES master mode routine. The file building routine must do the following:

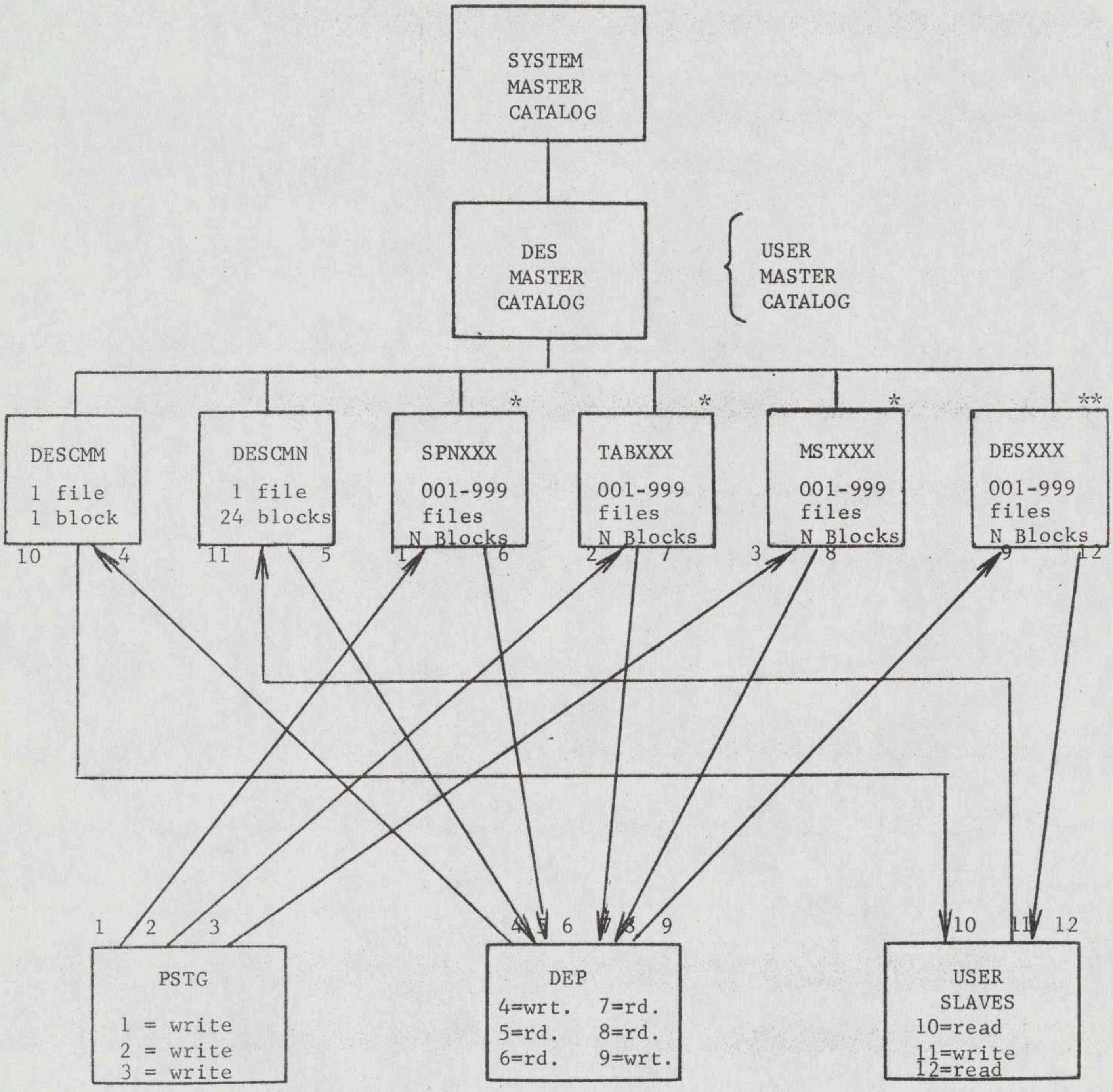
a) Build PAT pointers for the files in the SSA.

b) Build PAT bodies.

c) Enter file names and spare requirements in the Catalog System.

*DESCMM will not be created through master mode routine. It will be a normal perm file consisting of one block (320 words).

**DESCMN will not be created through the master mode routine. It will be a normal perm file consisting of 24 blocks (24x320 words).



* Created by the PSTG using the master mode file create routine

** Created by the DEP using the master mode file create routine

Note: XXX = file name with any subname

13.10 DEP to SLAVE COMMUNICATION (DESCMM)*

The DEP will write a management communication file to mass storage which will provide the user sufficient information to properly read DEP created sort files, concurrent with DEP writing these files. The communication file will be a perm file named DESCMM which allows user slaves read permissions. This file will contain one (1) 320 word block of file control information necessary for user file access to DEP created files.

The 320 DESCMM block will contain the following management blocks and control words.

- Word No. 1 will contain "DES6000". If the DEP is not running, this will contain zeroes. This word should always contain DES6000 for spawned jobs.
- Word No. 2: Date MMDDYR
- Word No. 3: Time OOHRMN
- Word No. 4: Message sequence number. Each time DEP writes a new message, this counter will be incremented by one. The first message will start with 1. The user may test this to determine if a later message has been filed.
- Word No. 7 (0-17): Number of DEP sort files in the stack 0-15. If this word is zero, there are no sort files open with data.
- Words 8-187: Fifteen file management blocks. Each block contains 12 words. The user must access these to determine the number of blocks that can be processed and checkpoint sequence number for each sorter.
- Word No. 188, bits 0-17: Number of slave jobs running or holding sort files, 0-12. If this counter is zero, no slave jobs holding or using files.
- Words 189-212: Twelve slave job management blocks. Each block contains two words. The user must test the first word of each block to find his slave programs identifying name or number which was placed on the "SPAWN" control card. The second word bits 0-14, for files sequence numbers 1-15, identify the slaves file management blocks within the stack area, words 8-187.

  Example - bit 0 is on, the file management block starts at word 8 as it is the first file in the stack area.

13.10 DEP to SLAVE COMMUNICATION (DESCMM)* Cont'd.

- Example - bit 1 is on, the file is at location 8 plus 12 as it is the second file in the stack.

  Example - bit 2 is on, the file is at location 8 plus 24 as it is the third file in the stack.

- Word No. 213, bits 0-17: Number of sorters in execution 0-6. If zero, no sorters are running.

- Words 214-273: Six sorter management blocks. Normally the user will not need to reference these block. The DEP uses the information in case of restart. Each block contains 10 words.

- Word No. 274, bits 0-17: Number of sort jobs in abort hold status 0-8. If counter equals zero, no sort jobs are in abort hold. Normally the user will not need to reference this data.

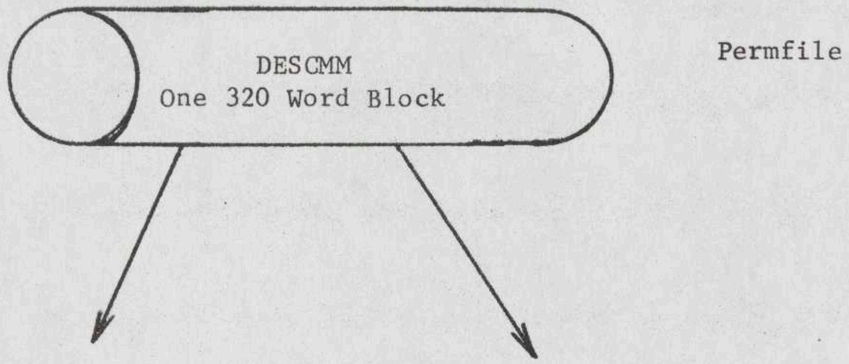
- Words 275-306: Contains eight abort hold management blocks. Each block contains four words. Normally the user will not need to reference these blocks.

The user must do the following to properly access the DESCMM file.

- Include a "PERMFIL" card in his object deck set up giving the name DESCMM with read permissions.

- Each time the file is accessed in the slave program, the sequence should be Open, Read, Close. If using GMAP, a 320 word block must be specified, the file read with a MME GEINOS and rewound. These steps are necessary as the DESCMM file is only one block in size and DEP overlaps the previous block on each write.

13.10 DEP to SLAVE COMMUNICATION (DESCMM)* Cont'd.



1. User Reads	1. DEP Writes
2. Rewinds	2. Rewinds
3. Reads Again	3. Writes Again

DESCMM COMMUNICATION AND RESTART MGMT. BLOCK

	0	35
0	Reserved for I/O	
1	Zero or "DES6000"	
2	Date	
3	Time	
4	Message Seq. No.	
5	Zero	
6	Zero	
7	0 No Des Files 17	18 Zero 35
	Fifteen file mgmt. blocks. Each entry = twelve words	
187	0 17	18 35
188	No. Slave Jobs Running	Zero
	12 slave mgmt. blocks Each block = 2 words	
212		
213	0 No. Sorters Running 17	Next available SNUMB
	Six sorter mgmt. blocks Each block = 10 words	
273		
274	0 No. Jobs Aborted 17	Zero
	Eight abort mgmt. blocks Four words each block	
306		
307	Zero	
319		

13.11 SLAVE JOB MANAGEMENT BLOCK

After the user has read the DESCMM file, for the first time he should determine that the DEP is in and running by testing word one (1) of the DESCMM block for the word "DES6000". If DES6000 is contained in word one the DES6000 is running.

The next step is to find the slave program's IDENT (the six digit name placed in the "Spawn" control card at sort generation time) in the Slave job management block. When the IDENT is found word 2, bits 0-23, will contain the file locators. See Slave Job Mgmt. Block. After calculating all the file location, test word 4, bits 12-35, of each file mgmt. block to determine if the file contains any data that may be processed. If word four, bits 12-35, of the file mgmt. block is greater than zero, the file may be opened and read.

SLAVE JOB MGMT. BLOCK

	0 23	24 29	30 35
1	SLAVE JOB IDENT from Spawn Control Card		
2	* File Locators 1-24 bit 0=File 1, bit 1=File 2 etc.	SPAWN CTL	User Comm File #

User placed "END JOB" in word 1 when slave goes to END at job. This will allow respawning on subsequent sort passes where only a single copy at the slave is required.

*Bit 0 = File mgmt. block #1 located (starting) at word 8.

Bit 1 = File mgmt. block #2 located (starting) at word 20.

Bit 2 = File mgmt. block #3 located (starting) at word 32.

Bit 14 = File mgmt. block #15 located (starting) at word 176.

Bits 15-23 not used at present.

13.12 FILE MGMT. BLOCK

The File mgmt. blocks, one for each open file up to fifteen, controls writing and reading of the files. This, along with document header information contained in the files themselves, permit the user programs to read the files and at the same time, properly control file processing. After determining which files belong to the Slave program as outlined under the "slave job mgmt. blocks" the user should test the file control block, for each file to be read, as follows:

13.12.1 Test word 4, bits 12-35 (total no. of blocks written on the file) for non zero. If this field is zero, do not open the file, as no data as yet has been written by DEP.

13.12.2 If the file had no blocks written reread DESCMM and repeat step 1., or test another file that may be read by the slave.

13.12.3 If word 4, bits 12-35 are non zero the file may be opened.

13.12.4 Open the file either as a random file or a sequential file. This should be known and the slave program designed to handle the desired files in either random or sequential fashion. However word 4, bits 0-5, will contain a 1 for sequentially written files, or a 1 for randomly written files. If a random file, a job index will be contained in the first block read.

13.12.5 After reading the first data block processing may proceed on a sorter basis or a file basis.

13.12.6 On a sorter basis the user should:

a) compare the header checkpoint sequence number for the given sorter against the corresponding sorter checkpoint sequence number in the file mgmt. block. If the header sequence number is less than the file mgmt. block sequence number, there is another batch of documents following and the batch identified by the file sequence number may be safely processed.

b) If the header sequence number is equal to the file mgmt. sequence number, DESCMM should be read until a later checkpoint is determined.

13.13 FILE MGMT. BLOCK, Cont'd.

13.12.6 Continued

*c) As long as the checkpoint sequence number is satisfactory and the number of blocks written per sorter are greater than the slave has read processing may proceed.

*Normally this type of file will be random and the user must find the next block by getting the points or chain address from characters 31-33 of the doc. header. This will be in the first and last record in the block.

13.12.7 On a file basis (non random) processing, reading, of the file may proceed on the basis of reading the file--as long as the DEP block sequence number count of blocks written is greater than the block sequence number found in the block read.

13.12.8 When the slave determines that a later message is needed it must read DESCMM. If the control block read is a later message, the message sequence number will have been incremented by DEP (word 4).

13.12.9 The DEP will inform the slave when it needs file space by putting the word "REL" in word 12, bits 0-17. The user slave should test this field each time a new message is read from DESCMM.

13.12.10 If the slave can release space, the number of 320 word blocks that can be released should be placed in word 12, bits 18-35. The entire message block from DESCMM should then be written to the Slave to DEP communication file.

13.12.11 If the Slave does not intend to release any space, at any time, or until end of job, the message "NOL" (no release) should be set to zero and the entire message block from DESCMM written to the slave to DEP communication file.

FILE MGMT. BLOCK

1	0 Ⓓ USER FILE NAME or IDES FILE NAME 35			
2	User File Code Ⓓ 0 11	DES File Code Ⓓ 12 17	File Seq No. Ⓑ 18 23	Slaves Using File Ⓑ 24 35
3	Current File Size Ⓑ 0	Max File Size Ⓑ 35		
4	File Type Ⓑ 0 5	No. Sorters Using File 6 11	Total # Blocks Written to File Ⓑ (Block Seq. No.) 12 35	
5	*No. of Blocks still Available for Writing Ⓑ 0 17	Next Available Address Ⓑ 18 35		
6	Checkpoint Seq. No. for Sorter #1 Ⓑ 17	18 Last Block Seq. No. Written for Sorter #1		
7	Checkpoint Seq. No. for Sorter #2	Last Block Seq. No. Written for Sorter #2		
8	Checkpoint Seq. No. for Sorter #3	Last Block Seq. No Written for Sorter #3		
9	Checkpoint Seq. No. for Sorter #4	Last Block Seq. No Written for Sorter #4		
10	Checkpoint Seq. No. for Sorter #5	Last Block Seq. No Written for Sorter #5		
11	Checkpoint Seq. No. for Sorter #6	Last Block Seq. No Written for Sorter #6		
12	Job No. if M file	User to DEP *Communication		

B = binary field

D = BCD field

File type (word 4 bits 0-5) 1 = sequential
2 = random

*includes any released spare by users of file.

13.13 SORTER MGMT BLOCK

	0			35
1	0	17 17	18 29	30 35
2	0 Block # Continuing Last Checkpoint	17	18 Block No. of 1st Block This Job 35	
3	Sorter Status 0 5		Next Available Address and Chain Address	
4	SEQ. NO. OF DOC. CONTAINING LAST CHECKPOINT			
5	CURRENT SEQUENCE NO.			
6	TOTAL # DOCS. PROCESSED THIS JOB BY SNUMG			
7	TOTAL NO. DOCS. PROCESSED ALL JOBS			
8	NO LATE POCKET SELECTS			
9	NO LATE READS			
10	NO FEEDER PKT ALERTS			

13.14 ABORT MGMT BLOCK

1	SORT JOB # 0 17	JOB SNUMB 18 29	File Seq. No. 30 35
2	BLOCK # CONTAINING LAST CHECKPOINT 0 17	BLOCK # AT 1st BLOCK THIS JOB	
3		NEXT AVAILABLE ADDRESS	
4	SEQ. NO. OF DOC. CONTAINING LAST CHECKPOINT 0 35		

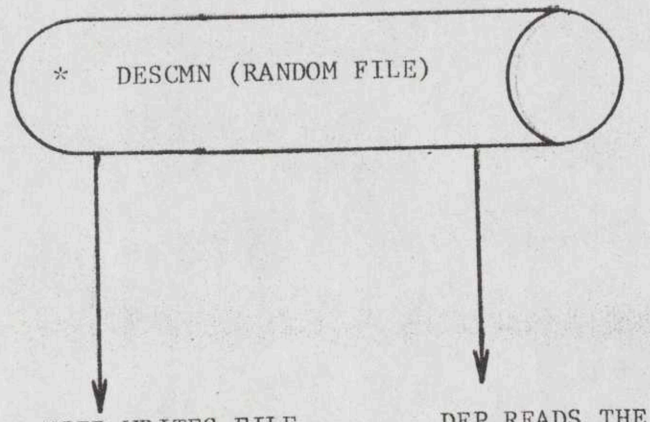
13.15 SLAVE TO DEP COMMUNICATION (DESCMN)

The user slave may communicate to the DEP program by writing to the DESCMN file. The DESCMN file will contain one block (320 words) for each user slave in the slave management area. Each slave will have an assigned block number to write to on the DESCMN file.

The DESCMN will be a random file containing 16 blocks (320 words). The DEP will ensure that each user slave is assigned a unique block address 1-16.

The user must inclde a $PRMFL card giving the name DESCMN with read and write permissions. This will allow the user to write the necessary communication information to the DEP. Also the user may use the file as a work file for the slave program.

USER TO DEP COMMUNICATION

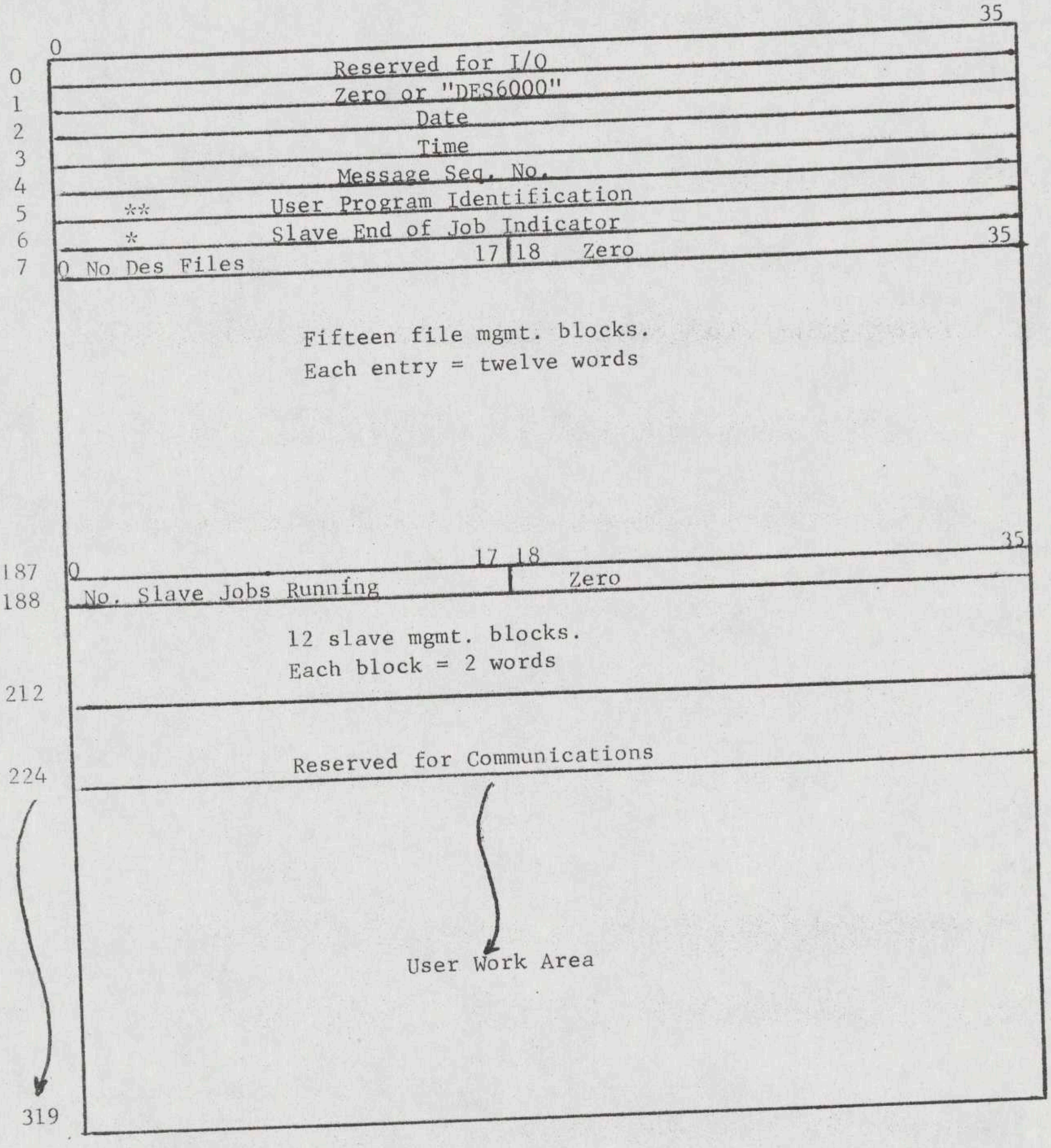


USER WRITES FILE
AND JOB STATUS
INFORMATION TO DEP
TO DESCMN USING
BLOCK NO. ASSIGNED
BY DEP. BLOCK NO
IS GIVEN IN THE
DESCMN FILE UNDER
SLAVE JOB MGMT.
BLOCK WORD 2, BITS
30-35.

DEP READS THE
FILE TO GET
USER SLAVE FILE
AND JOB INFORMATION.

*User may also read this file.

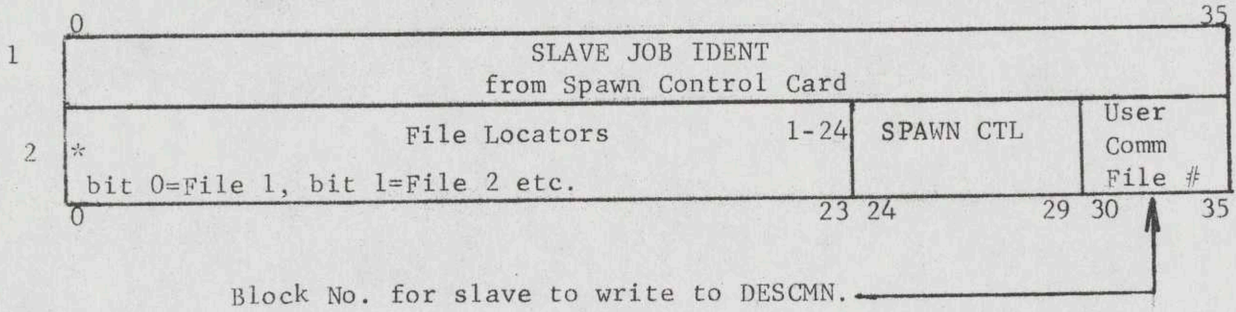
DESCMN COMMUNICATION - SLAVE TO DEP



*1 = Slave end of job
2 = Slave abort

**User must place his program ident in this word. Ident must be the same as given in the PSTG SPAWN control card.

SLAVE JOB MGMT. BLOCK



Block No. for slave to write to DESCMN.

*Bit 0 = File mgmt. block #1 located (starting) at word 8.

Bit 1 = File mgmt. block #2 located (starting) at word 20.

Bit 2 = File mgmt. block #3 located (starting) at word 32.

Bit 14 = File mgmt. block #15 located (starting) at word 176.

Bits 15-23 not used at present.

FILE MGMT. BLOCK

<table>
<tr><td>1</td><td colspan="5">0 Ⓓ USER FILE NAME or IDES FILE NAME 35</td></tr>
<tr><td>2</td><td colspan="2">User File Code
0 Ⓓ 11</td><td>DES File Code
12 Ⓓ 17</td><td>File Seq. No.
18 Ⓑ 23</td><td>Slaves Using File
24 Ⓑ 35</td></tr>
<tr><td>3</td><td colspan="3">Current File Size
0 Ⓑ</td><td colspan="2">Max. File Size
Ⓑ</td></tr>
<tr><td>4</td><td>File Type
0 Ⓑ</td><td>No. Sorters Using File</td><td colspan="3">Total # Blocks Written to File (Block Seq. No.)
Ⓑ</td></tr>
<tr><td>5</td><td colspan="3">*No. of Blocks still Available for Writing
0 Ⓑ 17</td><td colspan="2">Next Available Address
18 Ⓑ</td></tr>
<tr><td>6</td><td colspan="3">Checkpoint Seq. No. for Sorter #1
Ⓑ 17</td><td colspan="2">Last Block Seq. No. Written for Sorter #1
18 Ⓑ</td></tr>
<tr><td>7</td><td colspan="3">Checkpoint Seq. No. for Sorter #2</td><td colspan="2">Last Block Seq. No. Written for Sorter #2</td></tr>
<tr><td>8</td><td colspan="3">Checkpoint Seq. No. for Sorter #3</td><td colspan="2">Last Block Seq. No. Written for Sorter #3</td></tr>
<tr><td>9</td><td colspan="3">Checkpoint Seq. No. for Sorter #4</td><td colspan="2">Last Block Seq. No. Written for Sorter #4</td></tr>
<tr><td>10</td><td colspan="3">Checkpoint Seq. No. for Sorter #5</td><td colspan="2">Last Block Seq. No. Written for Sorter #5</td></tr>
<tr><td>11</td><td colspan="3">Checkpoint Seq. No. for Sorter #6</td><td colspan="2">Last Block Seq. No. Written for Sorter #6</td></tr>
<tr><td>12</td><td colspan="3">**</td><td colspan="2">User to DEP *Communication</td></tr>
</table>

B = binary field

D = BCD field

File type (word 4 bits 0-5) 1 = sequential
2 = random

*includes any released spare by users of file.

**contains "RES" or "REF"

RES = release the space given in 18-35 for rewriting.

REF = release the file space back to the system. File space given in word 3, bits 0-17 will be released. i.e., file removed from the system.

13.16 FILE TYPES

DEP files will be written in random fashion. There are three types of files the user may choose from for his file needs. These are defines as follows:

M-FILE - a multiplexed random-indexed single job file. From one to six sorters may write on this file so long as the same job number is being run on the sorter. Whenever a sort job is called for using an M-FILE the DEP will determine if a file already exists for the sort job; if so, the DEP will determine the next available write block on the existing file and start the job. The DEP, prior to starting the job, will place the job in the file index. The file index will consist of the first 320 word block on the file, as seen in "file index". Either DEP or the user may name this file depending on the user's choice, see "FILE Control Verbs."

N-FILE - a multiplexed random-indexed multiple job file. From one to six sorters running the same or different jobs may write to this file. Whenever a sort job is called using an N-FILE the DEP will determine if the file index will be updated and the job commenced. Data will be filed on the next available block in the file and continue in a random manner. The user must name this type of file DES 100 - DES 999, also see "FILE Control Verbs." The first block on this type of file (320 words) will contain the job file index.

S-FILE - a single job non-multiplexed file. Only one sorter with one job may access this file. It will contain a job file index. The user must let DEP name this file DES 001-099 and DEP must spawn a job to use the file sometimes between sort start and sort end of job.

SORTED INDEX FILE

The first block (320 words) of each Random file that DES has written will contain a file index. The file index will contain a two-word entry for each job on the file. The entry will contain the job number, a unique job snumb, the sorter the job was run on, the starting address in the file of the job, and a pointer to other entries which contain the same job number but run under a different snumb. The maximum number of entries will be 156 per file.

0 File Index 36

0	Reserved for I/O	
1	0 No. of Job Written to File 17	18 Next Available Entry Location 35
2	Zero	
3	Zero	
4	Zero	

JOB INDEX ENTRY

5	0 JOB NO. 17	18 SNUMB 29	30 Sorter # 1-6 35
	Relative Block No. of Starting Location	Chain Pointer to Next Entry for Same Job	

316

DES 6000 LOGICAL HEADER AND DATA RECORD

FIELD DESCRIPTION CODE	A	B		C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
FIELD NAME	SORT JOB No	SORT JOB SNUMB		YEAR	MONTH	DAY	SORT TIME	RUNST	DOC.T.	DOC.ST	USER.E.C.	NO CHAR IN DOC	PKT #	DOC SEQ NO.	FLOAT	SEND PT CODE	FILE SORTER CKPT SEQ. NO.	FILE SORTER BLOCK SEQ. NO..	FILE BLOCK PTR TO NEXT BLOCK	DATA
		DH #	SNUMB																	
LOC & NO CHARACTERS IN FIELD	1 2 3	4	5 6	7	8	9	10 12 11	13	14	15	16	17 18	19	20 22 24 21 23	25	26 27	28 30 29	31 33 32	34 36 35	*
STORAGE MODE	B C D	BCD	BIN	BIN	BIN	BIN	BIN	BCD	BCD	BCD	BIN	BIN	BIN	BIN	BIN	BIN	BIN	BIN	BIN	BCD
600-6000 Words	1			2				3					4		5			6		*

*S = DATA FROM WORDS 7-18 (37-108) FORMATTED OR PACKED

DES6000 HEADER FIELD DESCRIPTIONS

A. SORT JOB NO. The number which identifies the sort job called by the operator at the TTY servicing the document handler.

B. SORT JOB SNUMB. Unique number assigned by the DES sort program to identify the run to a particular handler and run. A new SNUMB will be assigned each time the job No. given under A is called. Character four of this field will contain the document handler, 1-6, and five and six a unique number within the job under A. SNUMBS are assigned in ascending sequence.

C., D., and E.

YEAR, MONTH and DAY. From the GECOS operating option the day is obtained via MME GETIME then converted to binary. See MME GETIME page 204 CPB 1518B.

F. SORT TIME. The time is obtained from MME GETIME. See CPB 1518B page 204.

G. RUN STATUS. This field will always contain a zero if the sort run is continuing satisfactorily, i.e., no restarts, no aborts etc. However, if the run has had to be restarted or the job has been killed by the operator this field will contain the following codes.

*1. = A restart has been initiated. The file has been rolled back to the last checkpoint.
NOTE: The user should always retain the block no. which contains the last checkpoint. This can be done by retaining field R from each previous block read.

2. = This run has been killed by the sorter operator.

3. = End of Job. i.e., last document this run.

*Normally the user will not detect a 1 code unless he is processing on a non checkpoint basis because after a roll back restart this field will again contain a zero (0).

NOTE: This field will be set by the TTY messages from the operator "KILL XXX-YY" and "RESTART XXX-YY".

H. DOCUMENT TYPES: The document type field will contain the following codes to indicate the following conditions:

DES6000 HEADER FIELD DESCRIPTIONS, Continued.

H. DOCUMENT TYPES, Cont'd.

Code	Description
1 =	A bill separator, I1 type, was placed in the pocket given in field L prior to this document being read.
2 =	Same as above for I2 type separator.
3 =	Same as above for I3 type separator.
* =	This is a checkpoint restart document.

NOTE: This field is set from the document status words. Word three bits 30 -33.

bit 30 on = checkpoint = *
bit 31 on = I1 separator = 1
bit 32 on = I2 separator = 2
bit 33 on = I3 separator = 3

I. DOCUMENT STATUS. This field will contain the following codes and meaning.

Code	Description
0	Data contained in 37-108 is in packed mode
1	Data contained in 37-108 is formatted per users sort logic.
2	A double feed has occurred document to garbage in packed mode.

NOTE: This field set from word three bit 34 of document status, or if a double feed from the read status word.

J. USERS ERROR CODE. Will contain a number 1-63 which the user has given in the sort logic to indicate an error condition. Zero indicates no error occurred in processing the document.

NOTE: This field is set from word three bits 24-29 of the document status.

DES6000 HEADER FIELD DESCRIPTIONS, Continued.

K. NO CHARACTERS IN DOCUMENT. The number of characters in the document 37-108 in the packed mode. If the document is formatted (37-108) the number of characters read in from the handler.

NOTE: This field is set from the document status words - word bits 10-17.

L. POCKET NUMBER. The pocket number which the document was selected to in the sort logic.

NOTE: This field is set from the document status words, word 3 bits 30-35.

M. DOCUMENT SEQ. NO. Contains the DES document seq. number. Each document will contain a seq. number beginning with 10 and incremented by 10 until all documents in the given job and SNUMB have been processed.

N. FLOAT. Float code obtained from the FRD-ABA master table - from 1-63.

NOTE: This field obtained by DEP from document status word 4 bits 0-5.

O. SEND POINT CODE. The send point code from 001-1000. This field is obtained from the FRD-ABA master table.

NOTE: This field obtained by DEP from document status word 4 bits 6-17.

P. FILE-SORTER CHECKPOINT SEQ. NO. The checkpoint sequence number for this sorter on this file. Each time a checkpoint is taken for a sorter writing to the file the checkpoint seq. no. will be incremented by one.

Q. FILE SORTER BLOCK SEQ. NO. This field will be incremented by one each time a sorter delivers a 320 word block of data to the file. Each sorter block number will commence with one for each sorter 1-6.

R. FILE BLOCK POINTER TO NEXT BLOCK. The location of the next block of data on the file for the job and sorter. This must be used as the random address for the next read.

S. DATA. Either formatted or packed.

DØCUMENT ENTRY SYSTEM - SCAN RECORD INPUT

JOB RUN CONTROL Card Type	Job Run No.	Loop Time	OPERATION Sequence	Action Verb	Switch Redefine	FIELD DEFINITION Type	Name I.D.	Locators Start Digit Q=1 Qs	Q=2 QUE	Stop Digit Q=1 Qs	Q=2 QUE	Length Maximum	Minimum	Process	Subfield Scale Ind.	SUBFIELD DEFINITION Location Within Field	Length	Name I.D.	TEST VALIDATION CONTROL Format Control	Error Accept Count LRA	TCD Mod	Type	Control <>	Test Comparison Success Pkt No.	Xfer Index	Failure Pkt No.	Xfer Index	Error Code	OPERAND TESTED SEARCHED Table Name / Constant Name / Literal Value / Special Value	Value Type Ind.
1	2 3 4	5 6	7 8 9	10 11 12 13 14 15 16 17	18	19	20 21 22 23 24 25 26 27	28	29	30	31	32 33	34 35	36 37	38	39 40	41 42	43 44 45 46 47 48	49 50 51 52 53	54 55	56 57	58	59	60 61	62 63 64	65 66	67 68 69	70 71	72 73 74 75 76 77 78 79	80
1	001	.	001	* IN COL.			1 IS A	COMMENTS CARD																						
Ⓐ	Ⓑ	Ⓒ	Ⓓ	Ⓔ	Ⓕ	Ⓖ	Ⓗ	Ⓘ	Ⓙ	Ⓚ	Ⓛ	Ⓜ	Ⓝ	Ⓞ	Ⓟ	Ⓠ	Ⓡ	Ⓢ	Ⓣ	Ⓤ	Ⓥ	Ⓦ	Ⓧ	Ⓨ	Ⓩ	①	②	③	④	⑤
*	001		002	THIS IS A SAMPLE SORT for DES-600																										
*	001		003	THIS IS A SORT RUN 001 which will breakout -																										
*	001		004	ON-US ITEMS AND SORT TRANSIT																										
*	001		005	SET UP VERBS MUST PRECEDE THE STARTUN VERB																										
*	001		006	*** DEVICE TYPE FOR RUN 001 IS THE MRS200 ***																										
S	001		007	DEVICE MRS200																										
*	001		008	*** THE SETVALUE VERB IS USED FOR POCKET CONTROL ***																										
*	001		009	SETVALUE M (00-11) = ALL POCKETS IN USE AND AN M TYPE SEPARATOR																										
*	001		010	WILL BE PUT IN EACH POCKET WHEN AN M SEPARATOR IS INDICATED																										
S	001		011	SET VALU	M		(00-11)																							
*	001		012	SET VALUE	I		(02,280.300)(03,280.300)(04,280.300) = KILL POCKETS.																							
*	001		013	and an I type document must be placed in these pockets																										
*	001		014	when the count is between 280 - 300.																										
S	001		015	SET VALU	I		(02,280.300)(03,280.300)(04,280.300)																							
*	001		016	SETVALUE	C		(350) = Turn on full pocket light when pocket count = 350.																							
*	001		017	SETVALUE	R		(11) = Reject Pocket																							
S	001		018	SET VALU	R		(11)																							

CE 326

DØCUMENT ENTRY SYSTEM - SCAN RECØRD INPUT

JOB RUN CONTROL			OPERATION			FIELD DEFINITION										SUBFIELD DEFINITION			TEST VALIDATION CONTROL										OPERAND TESTED SEARCHED	
								Locators				Length							Format		TCD		Test Comparison							
								Start		Stop														Success		Failure				
Card Type	Job / Run No.	Loop Time	Sequence	Action Verb	Switch / Redefine	Type	Name I.D.	Digit to Q=1 #Qs	Q=2 QUE	Digit Q=1 #Qs	Q=2 QUE	Maximum	Minimum	Process	Subfield Scale Ind.	Location Within Field	Length	Name I.D.	Format Control	Error Accept Count / LRA	Mod	Type	Control - <>	Pkt No.	Xfer Index	Pkt No.	Xfer Index	Error Code	Table Name / Constant Name / Literal Value / Special Value	Value Type Ind.
1	2 3 4	5 6	7 8 9	10 11 12 13 14 15 16 17	18	19	20 21 22 23 24 25 26 27	28	29	30	31	32 33	34 35	36 37	38	39 40	41 42	43 44 45 46 47 48	49 50 51 52 53	54 55	56 57	58	59	60 61	62 63 64	65 66	67 68 69	70 71	72 73 74 75 76 77 78 79	80
*	001		019	** BEGIN RUN HERE - START PROCESSING 1ST ITEM ****																										
Ⓐ	Ⓑ	Ⓒ	Ⓓ	Ⓔ	Ⓕ	Ⓖ	Ⓗ	Ⓘ	Ⓙ	Ⓚ	Ⓛ	Ⓜ	Ⓝ	Ⓞ	Ⓟ	Ⓠ	Ⓡ	Ⓢ	Ⓣ	Ⓤ	Ⓥ	Ⓦ	Ⓧ	Ⓨ	Ⓩ	①	②	③	④	⑤
S	001		020	START RUN																										
*	001		021	**GET THE TRANSIT FIELD. TRANSIT QUE = . TRANSIT IS REDEFINED 3 TIMES																										
S	001		022	GET FIELD	1	D	TRST	.		.				9																
S	001		022	" "	2	D	"	.		.				9																
S	001		022	" "	3	D	"	#		.				9												10		20		
*	001		023	**CHECK THE TRANSIT FIELD FOR 4 DIGITS A DASH & 4 DIGITS - ACCEPT A NON-NUMERIC FOR A DASH.																										
S	001		024	CKFORMAT	1	D	TRST												44	01						10		21		
*	001		025	**TEST THE TRANSIT FIELD FOR ON-US ABA NUMBER. DO NOT USE THE DASH **																										
S	001		026	TEST			TRST																				047		ON-US	K
*	001		027	**TEST SWITCH 0 TO SEE IF BLOCK HEADER PREVIOUSLY READ.**																										
S	001		028	TEST SW	0																						070			
*	001		029	** BLOCK HEADER PREVIOUSLY READ, TEST SWITCH 1 TO SEE IF BATCH HEADER READ. **																										
S	001		030	TEST SW	1																						069			
*	001		031	** BATCH HEADER PREVIOUSLY READ - GET ACCT FIELD AND PERFORM TCD **																										
S	001		032	GET FIELD	1		ACCT *			1	*	2		8																
S	001		032	GET FIELD	2		ACCT *			1	*			8																
S	001		032	GET FIELD	3		ACCT .				*															10		25		

DOCUMENT ENTRY SYSTEM - SCAN RECORD INPUT

JOB RUN CONTROL			OPERATION			FIELD DEFINITION									SUBFIELD DEFINITION				TEST VALIDATION CONTROL										OPERAND TESTED SEARCHED	
								Locators				Length							Format		TCD			Test Comparison						
								Start Digit		Stop Digit														Success		Failure				
Card Type	Job Run No.	Loop Time	Sequence	Action Verb	Switch Redefine	Type	Name I.D.	Q=1 =Qs	Q=2 QUE	Q=1 =Qs	Q=2 QUE	Maximum	Minimum	Process	Subfield Scale Ind.	Location Within Field	Length	Name I.D.	Format Control	Error Accept Count LRA	Mod	Type	Control = < >	Pkt No.	Xfer Index	Pkt No.	Xfer Index	Error Code	Table Name Constant Name Literal Value Special Value	Value Type Ind.
1	2 3 4	5 6	7 8 9	10 11 12 13 14 15 16 17	18	19	20 21 22 23 24 25 26 27	28	29	30	31	32 33	34 35	36 37	38	39 40	41 42	43 44 45 46 47 48	49 50 51 52 53	54 55	56 57	58	59	60 61	62 63 64	65 66	67 68 69	70 71	72 73 74 75 76 77 78 79	80
*	0 0 1		0 3 3	**GET BRANCH BANK # FROM ACCT. FIELD ALSO CHECK FORMAT OF ACCT. FIELD. **																										
Ⓐ	Ⓑ	Ⓒ	Ⓓ	Ⓔ	Ⓕ	Ⓖ	Ⓗ	Ⓘ	Ⓙ	Ⓚ	Ⓛ	Ⓜ	Ⓝ	Ⓞ	Ⓟ	Ⓠ	Ⓡ	Ⓢ	Ⓣ	Ⓤ	Ⓥ	Ⓦ	Ⓧ	Ⓨ	Ⓩ	①	②	③	④	⑤
S	0 0 1		0 3 4	CKFORMAT	1	D	ACCT												34	01						10		26		
S	0 0 1		0 3 5	GETS UBF		N	ACCT								S	01	03	BR NH												
S	0 0 1		0 3 6	GETS UBF		N	ACCT								S	05	04	ACT 1												
S	0 0 1		0 3 7	CHK TCD			ACT 1														10	N				10		30		
*	0 0 1		0 3 8	TCD TEST O.K. TCD WAS MOD 10 NATURAL																										
*	0 0 1		0 3 9	**VALIDATE THE DOLLAR FIELD AND TEST FOR HI DOLLAR ITEM **																										
S	0 0 1		0 4 0	VALIDATE	1	N	DOLLAR	$		$				10																
S	0 0 1		0 4 0	VALIDATE	2	N	DOLLAR	$		#				10																
S	0 0 1		0 4 0	VALIDATE	3	N	DOLLAR	#		$				10																
S	0 0 1		0 4 0	VALIDATE	4	N	DOLLAR	Q		$				10																
S	0 0 1		0 4 0	VALIDATE	5	N	DOLLAR	A		$		15	10				10									10		28		
S	0 0 1		0 4 1	TEST			DOLLAR																>			10		27	ZERO	K
*	0 0 1		0 4 2	** TEST DOLLAR FIELD FOR HI DOLLAR AMOUNT **																										
S	0 0 1		0 4 3	TEST			DOLLAR																>		6	7			HI DOLLAR	K
S	0 0 1		0 4 4	END TEST																										

DØCUMENT ENTRY SYSTEM - SCAN RECØRD INPUT

JOB RUN CONTROL: Card Type	Job Run No.	Loop Time	OPERATION: Sequence	Action Verb	Switch / Redefine	FIELD DEFINITION: Type	Name I.D.	Locators Start Digit Q=1 #Qs	Start Q=2 QUE	Stop Digit Q=1 #Qs	Stop Q=2 QUE	Length Maximum	Minimum	Process	Subfield Scale Ind.	SUBFIELD DEFINITION: Location Within Field	Length	Name I.D.	TEST VALIDATION CONTROL: Format Control	Error Accept Count / LRA	TCD Mod	Type	Test Comparison Control = < >	Success Pkt No.	Success Xfer Index	Failure Pkt No.	Failure Xfer Index	Error Code	OPERAND TESTED SEARCHED: Table Name / Constant Name / Literal Value / Special Value	Value Type Ind.
1	2 3 4	5 6	7 8 9	10 11 12 13 14 15 16 17	18	19	20 21 22 23 24 25 26 27	28	29	30	31	32 33	34 35	36 37	38	39 40	41 42	43 44 45 46 47 48	49 50 51 52 53	54 55	56 57	58	59	60 61	62 63 64	65 66	67 68 69	70 71	72 73 74 75 76 77 78 79	80
*	0 0 1		0 4 5	*** TEST FOR BLOCK AND BATCH HEADER AND SEPARATORS ***																										
Ⓐ	Ⓑ	Ⓒ	Ⓓ	Ⓔ	Ⓕ	Ⓖ	Ⓗ	Ⓘ	Ⓙ	Ⓚ	Ⓛ	Ⓜ	Ⓝ	Ⓞ	Ⓟ	Ⓠ	Ⓡ	Ⓢ	Ⓣ	Ⓤ	Ⓥ	Ⓦ	Ⓧ	Ⓨ	Ⓩ	①	②	③	④	⑤
*	0 0 1		0 4 6	*** SET SW 0-ON IF BLOCK HEADER, SET SW 1 ON IF BATCH HEADER ***																			=	11			050		HDR1	K
S	0 0 1		0 4 7	TEST .			TRANST																							
S	0 0 1		0 4 8	SET SW	0																									
S	0 0 1		0 4 9	END TEST																			=	11			054		HDR2	K
S	0 0 1		0 5 0	TEST			TRANST																							
S	0 0 1		0 5 1	SET SW	1																									
S	0 0 1		0 5 2	END TEST																										
*	0 0 1		0 5 3	**TEST FOR A SEPARATOR IF NO SEPARATOR TEST TRANSIT FIELD FOR END OF JOB.***																			=	M			057		SEP1	K
S	0 0 1		0 5 4	TEST			TRANST																							
S	0 0 1		0 5 5	END TEST																										
*	0 0 1		0 5 6	**TEST FOR I SEPARATOR & $ FIELD - SEARCH M20 TRANSIT TABLE																			=	I			064		SEPT2	K
S	0 0 1		0 5 7	TEST			TRANST																							
S	0 0 1		0 5 8	END TEST																										
S	0 0 1		0 5 9	VALIDATE	1	N	DOLLAR				$			10																
S	0 0 1		0 5 9	VALIDATE	2	N	DOLLAR				$			10																
S	0 0 1		0 5 9	VALIDATE	3	N	DOLLAR				$			10													9		30		
S	0 0 1		0 5 9	VALIDATE	4	N	DOLLAR				$			10																

CE 326

Honeywell

Rev. A

43A177886

DØCUMENT ENTRY SYSTEM – SCAN RECØRD INPUT

JOB RUN CONTROL			OPERATION			FIELD DEFINITION										SUBFIELD DEFINITION			TEST VALIDATION CONTROL										OPERAND TESTED SEARCHED	
								Locators				Length							Format		TCD		Test Comparison						Table Name	
								Start Digit	to	Stop Digit														Success		Failure			Constant Name Literal Value Special Value	
Card Type	Job / Run No.	Loop Time	Sequence	Action Verb	Switch / Redefine	Type	Name I.D.	Q=1 Qs	Q=2 QUE	Q=1 Qs	Q=2 QUE	Maximum	Minimum	Process	Subfield Scale Ind.	Location Within Field	Length	Name I.D.	Format Control	Error Accept Count / LRA	Mod	Type	Control < = >	Pkt No.	Xfer Index	Pkt No.	Xfer Index	Error Code		Value Type Ind.
1	2 3 4	5 6	7 8 9	10 11 12 13 14 15 16 17	18	19	20 21 22 23 24 25 26 27	28	29	30	31	32 33	34 35	36 37	38	39 40	41 42	43 44 45 46 47 48	49 50 51 52 53	54 55	56 57	58	59	60 61	62 63 64	65 66	67 68 69	70 71	72 73 74 75 76 77 78 79	80
S	0 01		0 6 0	TEST			DOLLAR																>			9		31	ZERO	K
Ⓐ	Ⓑ	Ⓒ	Ⓓ	Ⓔ	Ⓕ	Ⓖ	Ⓗ	Ⓘ	Ⓙ	Ⓚ	Ⓛ	Ⓜ	Ⓝ	Ⓞ	Ⓟ	Ⓠ	Ⓡ	Ⓢ	Ⓣ	Ⓤ	Ⓥ	Ⓦ	Ⓧ	Ⓨ	Ⓩ	①	②	③	④	⑤
S	0 0 1		0 6 1	SRCH			TRST																=			9			M10	T
S	0 0 1		0 6 2	END TEST																										
*	0 0 1		0 6 3	** TEST FOR END OF JOB																										
S	0 0 1		0 6 4	TEST			TRST																=				067		EOJ	K
S	0 0 1		0 6 5	END RUN																										
*	0 0 1		0 6 6	**TEST FOR PREVIOUS BLOCK AND BATCH HEADER***																										
S	0 0 1		0 6 7	TEST SW	0																						070			
S	0 0 1		0 6 8	TEST SW	1																					059				
S	0 0 1		0 6 9	HALT DH																									BATCH-H	C
S	0 0 1		0 7 0	HALT DH																									BLOCK-H	C
*	0 0 1		0 7 1	***FILE, CONTROL CARD WITH FILE CODE, FILE SIZE																										
S	0 0 1		0 7 2	FILE CTL	A1,,25,50																									
S	0 0 1		0 7 4	END																										

DOCUMENT ENTRY SYSTEM - MASTER TRANSIT TABLE INPUT

MASTER IDENT.	SEQ. NUMBER	FRB-ABA			* MULTIPLE ABA'S	PRIME RUN NUMBER	PRIME POCKET NO.	S.P. IND.	SEND PT. CODE	FLOAT	SECONDARY RUN NUMBERS AND POCKETS												BIG-ITEMS				SEND POINTS-COLLECTION TIME CODES-																		
		FRB		ABA							1			2			3			4			AMT																						
		DISTRICT	CITY STATE								RUN NO.	POCKET NO.	S.P. IND	RUN NO.	POCKET NO.	S.P. IND	RUN NO.	POCKET NO.	S.P. IND	RUN NO.	POCKET NO.	S.P. IND	10,000	1,000	100	POCKET NO.	T 1	T 2	T 3	T 4	T 5	T 6	T 7	T 8	T 9	T 10	T 11	T 12	T 13	T 14	T 15	T 16	T 17	T 18	
1 2 3	4 5 6 7	8 9	10 11	12 13 14 15	16	17 18 19	20 21	22	23 24 25	26 27	28 29 30	31 32	33	34 35 36	37 38	39	40 41 42	43 44	45	46 47 48	49 50	51	52	53	54	55 56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75 76 77 78 79 80
M																																													
100	001	12	21	010	0	001	02		005	05																																			
100	002	12	21	020	0	001	03		6	07																																			
A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W		X	Y											Z	Z								

CE 325

DOCUMENT ENTRY SYSTEM - GENERAL TABLES INPUT

IDENT	RUN NO.	SEQ. NO.	TABLE NAME	SORT IND.	TABLE TYPE	TCD ID No.	Length of Arg. in Chars.	TCD Div. or Time Codes 01-18	28 — TABLE DATA →			Pocket Type	Pocket No.	Alt Pocket
									COLLECTION TIMES TIME OF DAY					
									FROM 00 2400	TO 00 2400				
1 2	3 4 5	6 7 8 9	10 11 12 13 14 15 16 17	18	19 20 21 22	23	24 25	26 27	28 29 30 31	32 33 34 35	36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76	77	78 79	80
GT			ONUS		MISC		09		1221	0500				
T			HDR1	"S" OR BLANK	MISC		09		9999	9999				
T			HDR2		MISC		09		9999	3333				
T			SEP1		MISC		09		8888	8888				
T			SEP2		MISC		09		8888	7777				
T			EOJ		MISC		09		9999	5555				
A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
T			ZERO		MISC		10		0000	0000				
			HI											
			DOLR		MISC		10		0005	0000				

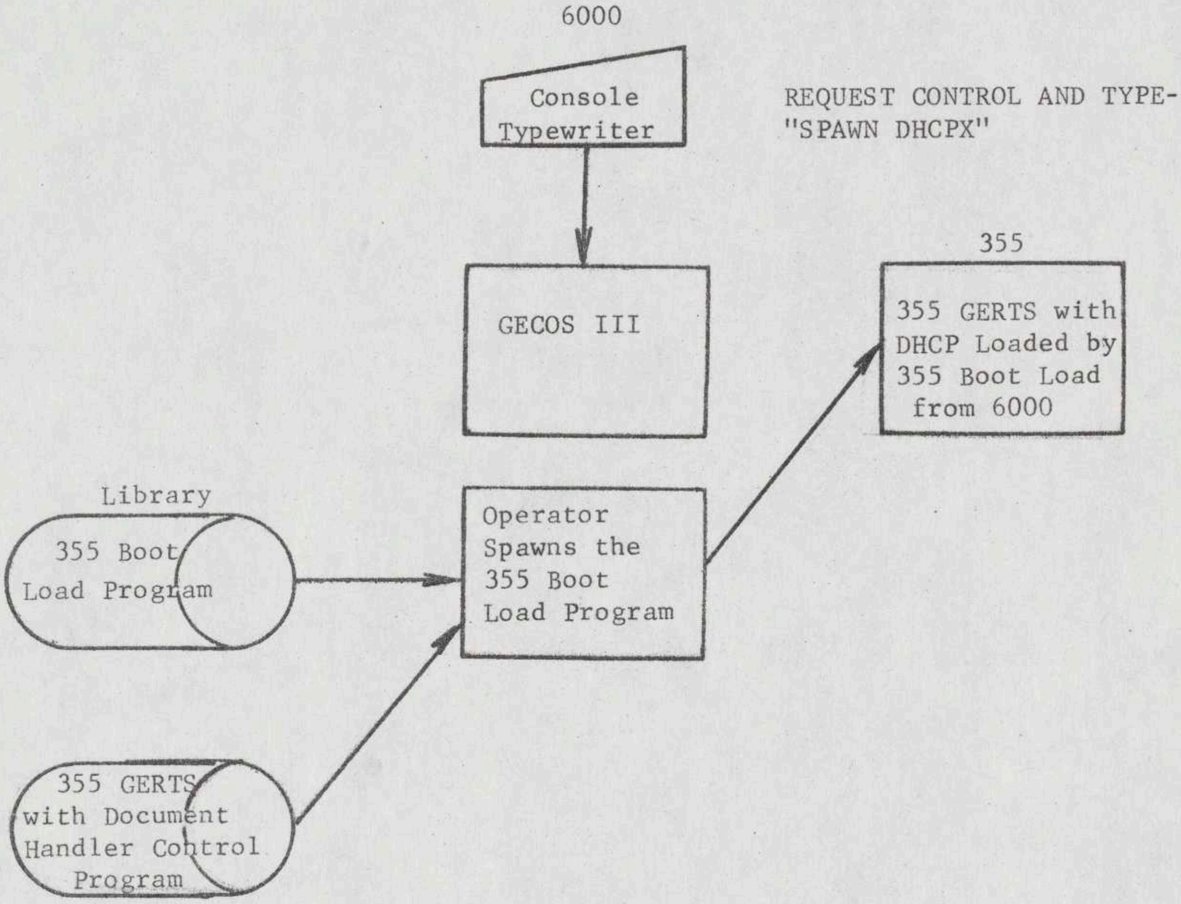
CE 324

Honeywell

Rev. A

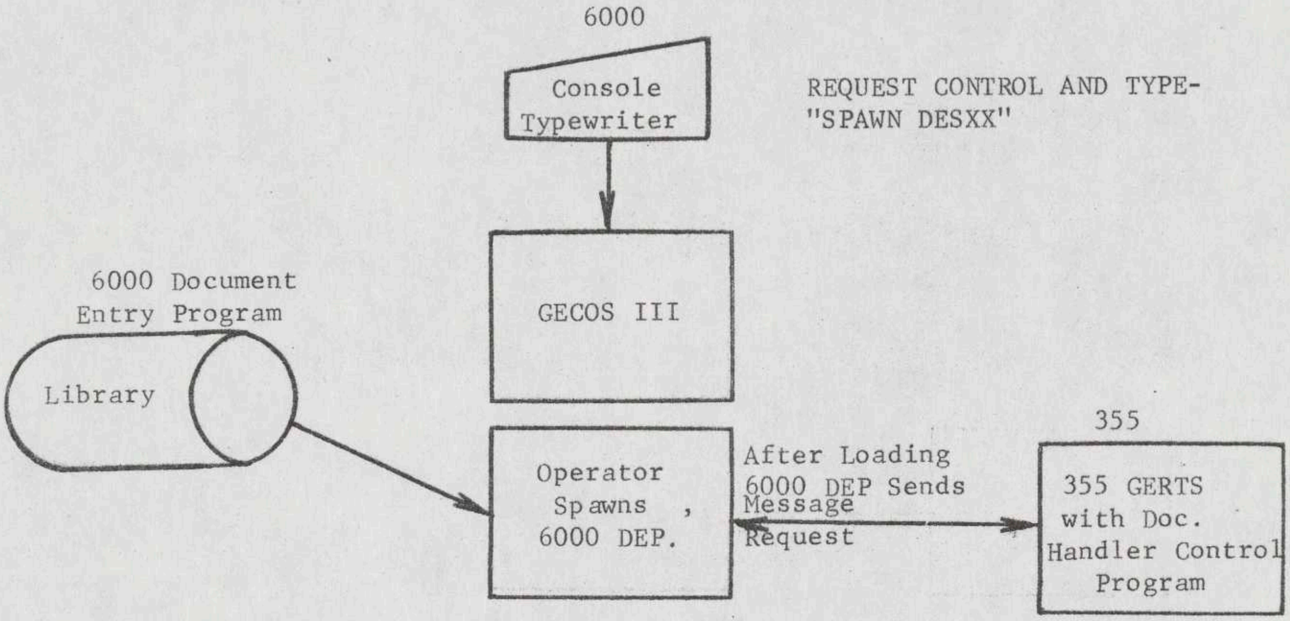
43A177886

LOADING 355 GERTS WITH DOCUMENT HANDLER CONTROL PROGRAM

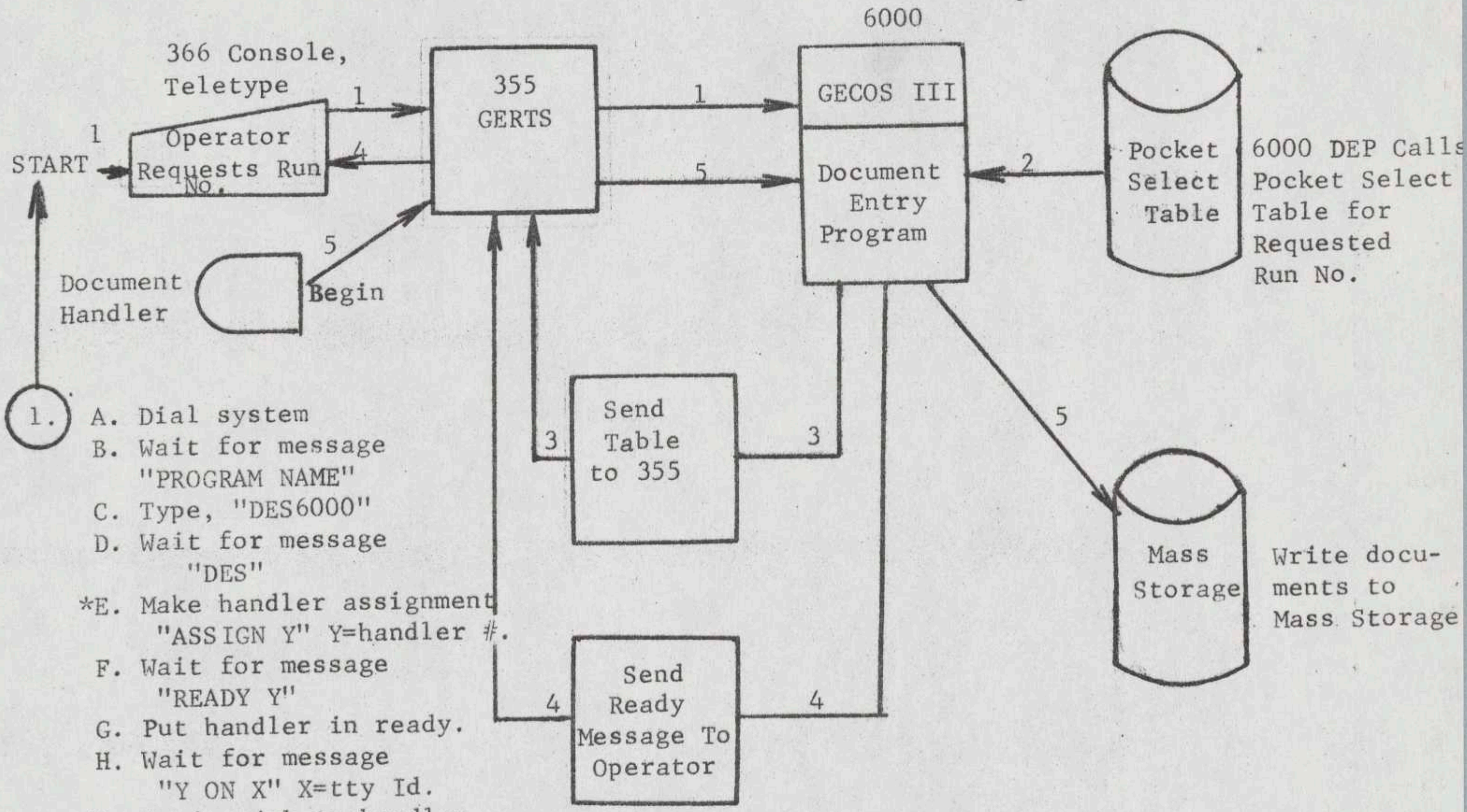


REQUEST CONTROL AND TYPE-
"SPAWN DHCPX"

LOADING THE 6000 DOCUMENT ENTRY PROGRAM



INITIATING A RUN ON THE DOCUMENT ENTRY SYSTEM



- A. Dial system
- B. Wait for message "PROGRAM NAME"
- C. Type, "DES6000"
- D. Wait for message "DES"
- *E. Make handler assignment "ASSIGN Y" Y=handler #.
- F. Wait for message "READY Y"
- G. Put handler in ready.
- H. Wait for message "Y ON X" X=tty Id.
- *I. Assign job to handler "NEW:ZZZ,Y" Z=job #, Y=handler #
- J. Wait for message "zzz accepted"
- *K. Type message "START Y"
- L. See message section

*Depress break key and wait for ?? before typing message.

13.19 DES MESSAGES

The 6000 Document Entry Program will accept and issue messages from and to the TTY's necessary for operator control, error discovery and recovery, test and diagnostics aids and system and job status. This will require the 6000 DEP to maintain a sizeable message table which will be used to verify, interrogate and respond to incoming messages from the TTY's.

Other messages that will be issued to the TTY's automatically from the 6000 DEP will also be required. These messages will communicate error thresholds, restart conditions, etc. to the operator.

The message which will be required, their meaning, their action or reaction on the system, are listed by category below.

Operator control messages will provide the means by which an operator via a TTY can initiate a job and control it through "end of Job." This requires the ability to correct error conditions which may occur within the job, restart the job, stop the job, kill the job, etc.

The sign on (initial request from the console teletype or typewriter for the DES-6000, procedure will require that the operator request control, dial on, break, etc. depending on the type of TTY or typewriter configured). The 355 GERTS will respond with a request for a program name "Program name?". The operator will then request the "DES".

The "DES" message will be sent to 6000 GECOS via the 355 and the connection established for the terminal to the 6000 DEP direct access program. The 6000 DEP will respond to the requesting terminal with the message "DES-6000 RDY". At this point the DES-6000 system is ready to accept the various command for setting up the document entry jobs.

a. Program - Name - This message will be issued by the 355 DHCP (GERTS) when initial request is received from the console on TTY terminals.

b. DES - This message will be entered on the TTY to request the 6000 DEP in the 6000.

c. DES-6000 rdy - This message will be issued by the 6000 DEP in response to the DES message. It will verify to the user (operator) that the Document Entry System (6000 and 355) is ready to run.

13.19 DES MESSAGES, Continued

13.19.1 ASSIGN yyzz

yy represents the TTY terminal and zz is the document handler number. This message permits the operator to assign the TTY (yy) to act as the message control console for one or two or more document handlers. When entered it will cause the 6000 DEP to issue all messages for the jobs running on the handler zz to the assigned TTY (yy).

13.19.1.1 RDY - zz

This message is issued by the 6000 DEP in response to "assigne yyzz". The operator is informed to put the document handler assigned by zz in ready. This will cause the 355 DHCP to respond to the special interrupt caused by putting the handler in ready and connect the document handler terminal to the 6000 DEP. The 6000 DEP will verify that the terminal identification (ID) for the document handler is correct. If correct, message 13.19.1.2 will be issued. If incorrect, message 13.19.1.6 will be issued.

13.19.1.2 "zz ON yy"

6000 DEP responds to the "assign yyzz with this message if the assignment can be made. This confirms that the document handler is assigned to the TTY designated by yy.

13.19.1.3 INVALID zz ON yy

In the event the document handler zz was previously assigned to another TTY, the 6000 DEP issued this message to indicate the current assignment cannot be made.

13.19.1.4 REASSIGN yyzz

This message permits the operator to change TTY's for a previously assigned handler.

13.19.1.5 zz BUSY

This message will be issued when an attempt is made to reassign TTY's to a handler

13.19 DES MESSAGES, Continued

13.19.1.5 Continued

and the handler is in use i.e. running a job. Whenever the handler is not busy and the reassignment can be made zz on yy will be issued. The message "zz busy" may be issued for a variety of other reasons as well.

13.19.1.6 CK-CONF-RDY-zz

The identification (ID) of the document handler readied does not match the assigned ID (zz) from the TTY. The operator should check the configuration deck and either ready the correct zz or change the assignment to correspond to the configuration.

13.19.2 TEST xxxx

This message is entered on the TTY when the operator wishes to run a test job on a particular document handler prior to production or for test diagnostics purposes. The purpose of the test job is to determine if, a handler is operating in a satisfactory manner.

Test xxxx causes the 6000 DEP to verify the test job (where xxxx is the test no.). Then extract any needed sort control data from the table file for running the test job and transfer it to the 355 DHCP. When the test is ready to run the 6000 DEP will issue a message, "Test xxxx rdy".

The 6000 DEP program will monitor all data (documents) coming from the document handler on which the test is being run. At end of test a message, E.O.T. xxxx, will be issued to the TTY servicing the document handler. Then totals for the test will be printed which will include the number of documents read followed by error totals by category.

No data will be written to tape or mass storage on test jobs.

13.19.2.1 TEST xxxx RDY

The 6000 DEP issued this message when the requested job, denoted by xxxx, is valid and the sort data and sort tables have been transferred to the 355 DHCP, i.e. the job is ready to run.

13.19 DES MESSAGES, Continued

13.19.2.2 xxxx INVALID

This message will be issued by the 6000 DEP when a test job, denoted by xxxx, is invalid.

13.19.2.3 EOT xxxx

The 6000 DEP will issue this message when the test job is finished. It will be followed by error totals as shown under 13.19.2.4.

13.19.2.4 TOT. NO. DOC. 9999

Test xxxx errors - If there is no detected errors the Test xxxx errors will be followed by the number 0 (zero). If there are errors, the errors will be listed as follows:

MRS 200 or DRD 200

Test xxxx errors -

hardware - feed failures -

data alerts -

T/Timing -

S/echo -

software -

13.19.2.5 DH-NO AND ZZ - XXX see 13.19.3.4 and 13.19.3.5

13.19 DES MESSAGES, Continued

13.19.3 NEW xxxx

The operator will enter this message through the TTY to request a new job (xxxx represents the job no.). The job will be run on the document handler previously assigned to the TTY by the "assign" message. When there are more than one document handlers assigned, to the TTY the 6000 DEP will automatically issue a message requesting the operator to assign a document handler for the job. (D.H. - no.?) if the job number requested is valid, the 6000 DEP will extract the necessary control and sort data from the table file and transfer it to the 355 DHCP. When the 355 DHCP returns a ready to run status for the job, the operator "Job xxxx rdy". In the event the requested job number is invalid the 6000 DEP will issue a message "xxxx invalid" and wait for the operator to enter a new job request: "new xxx".

13.19.3.1 22 xxxx RDY

After the 6000 DEP has 22 - xxx verified that the handler and job number, denoted by, is valid and all sort control data sent to the 355 DACP with a return status that the 355 and the disc handler are ready to run the requested job; this message will be issued to the operator.

13.19.3.2 xxx INVALID

This message is issued when the requested job no., denoted by xxx is invalid. The operator must repeat message 3. "New xxx" giving corrected job number.

13.19.3.3 EOS - zz - xxx

The message is issued at end of job.

13.19.3.4 DH - NO?

This message is issued when a job is requested from a TTY to which more than one document handler has been assigned. The operator must respond by typing the document handler identification (ID) and the job number zz-xxx, where zz is the document handler and xxx the job number.

13.19 DES MESSAGES, Continued

13.19.3.5 zz - xxx

The operator types the message in response to SB.

zz - document handler ID

xxx - job number

13.19.3.6 EOR zz-xxx

The documents have all been read but no E.O.J. control document. Job will not be closed until a positive close is given either from the console or a resuming procession with E.O.J. document.

13.19.4 START zz

This message is entered by the operator when he is ready to start a run. It must have been preceded by an "assign yyzz" and a "new xxx" or "test xxx". When this message is received by the 6000 DEP and is verified to be valid, i.e. the job number is correct and the 355 DHCP is ready to run the job, the 6000 DEP issues a command to the 355 DHCP to commence feeding and reading to the document handler on which the job is loaded.

13.19.4.1 zz NOT RDY

This message is issued to the operator when the 355 DHCP determines that the handler to which the job is assigned is not ready. The 355 DHCP returns the not ready status to the 6000 DEP which in turn issues the message. The operator must ready the handler after which message 13. 19.3.1 will be issued. ZZ represents the handler number. This should not occur unless the handler drops ready or the operator taxes the handler out of ready inadvertently.

13.19 DES MESSAGES, Continued

13.19.5 STOP zz

When the operator wishes to temporarily, stop a run on a document handler, he will issue the message zz represents the handler ID. This will cause a controlled stop of the handler and all documents following the stop of the handler will be read and pocketed by the 355 DHCP. Normal operation will be resumed by issuing a start zz message.

13.19.6 CLOSE zz - xxx

The operator can bring a job to a normal end of job or test by the use of the close message. Normally this message will be used when all documents have been processed for a job but there was no final control document to cause a normal end of job. ZZ represents the handler ID on which the job was running.

13.19.7 KILL zz - xxx

This message is used when it is necessary to terminate a run along with any files created by it. Data files created by the job will not be saved. When the kill option is used the job will have to be rerun.

13.19.8 ABORT zz

This message will be used when it is desired to terminate a run but save the data files created by the job. The job can be restarted at a later time by using the restart xxx. Processing will resume from the last restart point retained in memory by the 6000 DES. The 6000 DES will not stop the job until an appropriate restart point has been established. This prevents the need to reposition the data file at restart. When the abor zz is issued to the 6000 DES the job number running on zz is placed in a suspense table. When restart xxx is issued the suspense file is searched for a matching job number and if found the 355 DHCP is reinitiated for the given job and processing resumed from the retained check point.

13.19 DES MESSAGES, Continued

13.19.9 RESTART xxx-zz

This message is used to restart a job that was previously terminated by an abort zz or through a hardware failure. xxx is the job to be restarted and zz is the handler on which the job was aborted.

zz-xxx ABORTED

This message verifies that the requested job has been terminated and the handler on which it was running is available for another job.

NOTE: When a job is aborted and later restarted; it will be possible to restart the job on a different handler provided that: the handler is of the same type and not in use. This is made possible because sort tables and control information are reissued to the 355 DHCP. Also all control totals in the 6000 DEP will be saved after an abort and when the job is restarted carried through to end of job.

13.19.10 AUDIT xxx

Typing the audit xxx message will cause the 6000 DEP to print on the requesting TTY all the job totals for the job number denoted by xxx.

13.19.10.1 WW = Pocket XX KL SEP WW

If the condition arises where the number of documents directed to kill pockets prior to encountering a kill separator exceeds the authorized limit, the above message is typed to the operator.

13.19.10.2 ZZ ATTN

This message reflects a hardware status condition, (full pocket or empty feed hopper), that requires manual intervention.

13.19.10.3 ZZ JAM

Whenever the transport mechanism is implied to the point of preventing successful reading and pocketing of documents, or an endorser malfunction occurs, the above message is typed to the operator.

13.19 DES MESSAGES, Continued

13.19.10.4 ZZ SHUT DOWN

This message is typed to the operator whenever late pocket and late read commands exceed system limitations, or error conditions have occurred which require immediate handler shutdown.

13.19.10.5 ZZ REQ SEPRS

If the number of Batch Separators are insufficient for proper pocket allocation, the above message is typed to the operator.

13.19.10.6 ZZ NON-HOMO

Whenever the documents contained within a batch are, according to run types and transaction codes, not applicable to that run, the above message is typed to the operator.

13.19.10.7

ZZ $ ERRS	(Dollar amount errors)
ZZ TR ERRS	(Transaction code errors)
ZZ ACCT ERRS	(Account Number errors)
ZZ RTE ERRS	(Transit Number errors)
ZZ TCD ERRS	(Transposition Check Digit Errors)

The system maintains counts of the above errors encountered during processing. Whenever the error count exceeds an allowable limit the system so notifies the operator through one of the above messages.

13.19.10.8 ZZ CHANALERT
ZZ DATALERT
ZZ CMND REJ
ZZ CHANBSY

The above messages reflect hardware status conditions and will follow the ZZ SHUT DOWN message.

13.19.11 ERRORS zz

This message will be provided to assist T&D. Entering this message on the TTY will cause the 6000 DEP to print all hardware and/or possible hardware errors by category on the requesting TTY. Errors will be kept by job and by total jobs for each handler.

13.19 DES MESSAGES, Continued

13.19.12 REVIEW xxx

This message will be provided to enable an operator to get a printout of the jobs table sort and control data. Entering this message on the TTY will cause the 6000 DEP to printout the control information on the requesting TTY for the job number denoted by xxx. Review xxx may be issued prior to a start xxx to ensure that the table sort parameters are correct.

13.19.13 SWITCH zz TO mm

This message allows an operator to switch a run from one handler to another. ZZ represents the from handler and mm represents the to handler.

TITLE: Engineering Product Specification, Part 1
Document Handler Software for the DES6000

Approved by: Cw Dix 6/16/71
C. W. Dix, Director - Engineering Date

Reviewed by: JC Richter 5-29-71
J. C. Richter, Manager - Industry Application
Industry Marketing Date

Approved by: 6/16/71
R. F. Stevens, Manager - Operating Systems
Engineering Date

Approved by: L I Wilkinson 6/16/71
L. I. Wilkinson, Manager - Systems Engineering Date

Prepared by: CD Bolejack 8/18/71
C. D. Bolejack, System Software Design Date