

**DMERT OPERATION SYSTEM GENERIC UPDATE FACILITIES  
SOFTWARE SUBSYSTEM DESCRIPTION  
3B20D COMPUTER**

	PAGE		PAGE
1. GENERAL . . . . .	1	B. Output Messages . . . . .	12
2. OVERVIEW . . . . .	2	C. Error Codes . . . . .	13
3. SYSTEM UPDATE . . . . .	3	4. ABBREVIATIONS AND ACRONYMS . . . . .	15
SYSTEM UPDATE REQUIREMENTS . . . . .	3	<b>Figures</b>	
A. SUPR Requirement . . . . .	3	1. System Update Scenario . . . . .	6
B. DMERT Requirements . . . . .	4	2. SUPR Log Output . . . . .	9
C. Application Requirement . . . . .	6	3. System Update Menu Page Screen . . . . .	10
SYSTEM UPDATE PROCEDURES DESCRIPTION . . . . .	6	4. ENTER Arguments Page 1 Screen . . . . .	11
A. SUPR Enter Request . . . . .	6	5. ENTER Arguments Page 2 Screen . . . . .	12
B. SUPR Proceed Request . . . . .	7	6. SUPR PDS Input Messages . . . . .	12
C. SUPR Commit Request . . . . .	7	7. SUPR PDS Output Messages . . . . .	14
D. SUPR Backout Request . . . . .	7	<b>Tables</b>	
E. SUPR Restore Request . . . . .	7	A. SUPR Error Codes . . . . .	3
F. SUPR Utilities . . . . .	7	<b>1. GENERAL</b>	
SYSTEM UPDATE MENU PAGE DESCRIPTION . . . . .	8	1.01 This section provides a general functional description and information related to the system update portion of the program update facilities. The duplex multienvironment real-time (DMERT) operating system used by the 3B20D computer provides these facilities.	
A. System Update Menu Page Screen . . . . .	8	1.02 The reasons for reissuing this section are listed below. Revision arrows are used to emphasize the more significant changes.	
B. ENTER Arguments Page 1 Screen . . . . .	10	(a) Rewrote paragraph 2.04 and put SURR error codes into Table A.	
C. ENTER Arguments Page 2 Screen . . . . .	11		
SYSTEM UPDATE MESSAGE AND ERROR CODES . . . . .	11		
A. Input Messages . . . . .	11		

- (b) Added input message to paragraph 2.06.
- (c) Added paragraph 2.07 to describe the use of the system update menu page screens.
- (d) Added paragraph 3.07 to paragraph 3.06.
- (e) Deleted file system integrity reference from paragraph 3.10.
- (f) Deleted paragraph 3.21 and reference to file system integrity.
- (g) Deleted references to force disk from paragraph 3.27 and paragraph 3.29.
- (h) Revised paragraph 3.31 for clarity.
- (i) Added paragraphs 3.39 through 3.50 to describe the system update menu page.
- (j) Changed Figure 2 sequence option from single to double.
- (k) Added Figures 3, 4, and 5 to show the system update menu page screens.
- (l) Revised message in Figure 6.
- (m) Omitted messages referring to force disk from Figure 7.
- (n) Deleted reference to system update errors in paragraph 3.62 because they are listed in paragraph 2.04.
- (o) Revised Table A because SUPR error codes have been changed.

**1.03** System updates consist of applying new generics and generic issues to the 3B20D computer.

**2. OVERVIEW**

**2.01** The system update facilities provide a safe and reliable means for the craft to apply new versions of DMERT and application software while minimizing service disruption. These facilities allow installation of one or more partitions of DMERT or the entire system.

**2.02** Requirements for system update can be divided into three categories.

- (1) **System Update Procedure (SUPR) Requirements:** The SUPR requirements describe the capabilities that SUPR provides to assist with an update.
- (2) **DMERT Requirements:** The SUPR processes depend on certain DMERT facilities to provide the necessary update support.
- (3) **Application Requirements:** Each application must fulfill a variety of responsibilities to support the update since the system update normally introduces a new generic consisting of both DMERT and application software.

**2.03** The SUPR requirements support the 3B20D system update by providing a separate process for each step of the update procedure. The SUPR processes are associated with the following five input messages.

- SUPR ENTER
- SUPR PROCEED
- SUPR COMMIT
- SUPR BACKOUT
- SUPR RESTORE

**2.04** There are six types of errors that can cause an SUPR process to fail. These errors are listed in Table A.

**2.05** The SUPR requirement supports two update methods: off-line disk method and backup partition method. Using the off-line disk method, any number of partitions may be updated. Using the backup partition method, only those partitions having backups may be updated.

**2.06** If the new generic is faulty, the update may be reversed and the original system may be restored. If a step (process) fails, an error code containing information about the problem is provided in an output message (Table A). The input message **STOP:GEN[;UCL]!** is used to stop a step if something goes wrong.

♦TABLE A♦  
SUPR ERROR CODES

ERROR	ERROR CODES
UNIX* system call error	101xxy - 153xxy
Input data error	201xxy - 226xxy
Internal system update error	301xxy - 326xxy
Execution error	401xxy - 423xxy
Application processing error	501xxy - 502xxy
DMERT error	601xxy - 623xxy

\* Trademark of AT&T

♦2.07 All SUPR commands that can be executed using craft input messages can also be executed using the system update menu page. The system update menu page, one of three SUPR interface screens, provides a visual and user-friendly procedure for installing system updates. This procedure helps to eliminate problems caused when typing lengthy input commands. In addition to SUPR processes, an application process is executed from the menu page. The application process can be executed at the same time that a SUPR process is running.♦

### 3. SYSTEM UPDATE

#### SYSTEM UPDATE REQUIREMENTS

##### A. SUPR Requirement

3.01 Due to the major impact on an application during a system update, the update procedure is divided into several distinct steps. This division allows the craft to have direct control over the procedure, and this control enables the craft to choose the best time to begin each successive step for the update. This control also provides a means to cancel the update at any step of the procedure.

#### Backup Partition Method

3.02 The basis of the backup partition method is that each partition containing generic data is "backed up" by another partition containing the same data (unrelated to the duplication associated with duplex disk). One copy is designated as the primary partition and the other copy as the backup partition. The two partitions form a primary/backup partition pair.

3.03 At any given time, only one copy of a partition is in use. The new generic can be written into the unused partitions resulting in one set of old generic partitions and one set of new generic partitions. These two sets become the same again after completion of the update procedure. An example of a successful update using the backup partition method follows.

- (1) **Enter New Generic:** After determining which set of partitions are not in use, read the new generic data into those unused partitions.
- (2) **Proceed With New Generic:** Prepare for booting from the new generic partitions.

(3) **Boot From New Generic Partitions:** Boot system from the new generic partitions using the emergency action interface.

(4) **Commit To New Generic:** Copy the new generic partitions into the old generic partitions resulting in both copies containing the new generic.

**Off-Line Disk Method**

3.04 The off-line disk method involves writing new generic data onto one disk of a duplex disk pair. For each duplex disk pair updated, one disk is removed from normal operation and new generic data is written onto that disk. Upon completion of the update procedure, each pair of duplex disks contain the new generic. The off-line disk method relies upon the existence of duplex disk pairs.

3.05 The key to the off-line disk method is to always keep the disk(s) containing the generic not currently in use in the off-line state. The reason for keeping the nonactive disk(s) off-line is to keep them from being restored. The generic data on an off-line disk is protected from being overwritten by the generic data on its active mate.

3.06 If a system-initiated boot occurs during the update procedure, the system boots from the generic it was running on previously. Therefore, new generic disks are marked off-line in the incore equipment configuration database (ECD) as well as the ECD on disk. After the PROCEED step, the control unit is simplexed so that a specific system disk is forced in conjunction with a control unit. An example of a successful update using the off-line disk method follows.

(1) **Enter New Generic:** For each duplex disk pair to be updated, mark one of the disks off-line and read the new generic data onto the off-line disk.

(2) **Proceed With New Generic:** Prepare for booting from the new generic system disk. Specifically, mark the old generic disk(s) off-line in the new generic ECD. Each generic ECD now considers the disk(s) containing the other generic to be off-line.

(3) **Boot From New Generic:** Manually boot the system from the system disk containing

the new generic using the emergency action interface.

(4) **Commit To New Generic:** Restore the old generic disk(s) resulting in each duplex pair of disks containing the new generic.

**New Generic Data Integrity**

3.07 The SUPR verifies integrity of the new generic data in several ways. The new generic data input is analyzed for possible damage since its generation. The SUPR monitors the new generic data input for consistency within itself and with respect to the update method being used. A check is made to ensure new generic data is written onto the appropriate disk(s).

**Bootting From New/Old Generic**

3.08 Booting from either the new or old generic is done manually since some of the boot options on the emergency action interface page change from their previous state. The specific boot option used depends upon the current configuration and the update method. The SUPR provides the craft with instructions indicating the boot options to select to boot from the appropriate generic.

**B. DMERT Requirements**

3.09 The SUPR processes rely on support from other software within DMERT to provide the update facilities. Other than the general operating system capabilities; such as, those provided by the UNIX operating system supervisor, SUPR requires existence of the following DMERT facilities.

- Volume table of contents
- Generic input
- Disk partition access
- Configuration information
- Off-line disk support
- Partition copy.

### Volume Table of Contents

**3.10** The SUPR processes use disk-related information from the volume table of contents. Besides knowing the location and size of each partition on the disk, SUPR determines whether a given partition is a primary or backup partition and which partition is its backup (if it has one). Also, the volume table of contents contains an identification of the logical content of the disk (which set of partitions are defined by the volume table of contents). This ensures that the new generic data is being targeted for the appropriate disk.

### Generation of Generic Input

**3.11** The main input to SUPR is generic data in the load-disk-from-tape format. When input for SUPR is generated, certain facilities must be available to the application. The application must:

- Specify a partition as primary or backup
- Define partition pairs
- Specify which partition to place on the same disk
- Define contents of each partition
- Generate new generic data for multiple disks
- Identify which partitions to include in the new generic data.

### Disk Partition Access

**3.12** The SUPR processes write the new generic data onto disks on a partition basis. This means that SUPR accesses a specific partition on a specific disk using information from the volume table of contents for that disk.

### Configuration Information

**3.13** The SUPR processes access software configuration information. This includes whether the system is running on primary or backup (root or broot); which boot file is being used (dmert/appdmert); and on which disk the system is currently running.

### Off-Line Disk Support

**3.14** The provision of the off-line disk method relies upon a variety of DMERT capabilities. The following paragraphs summarize these capabilities.

**3.15** When new generic data is received for a set of partitions targeted for the same disk, a means of identifying which duplex disk pair is involved must exist. The SUPR then selects one of the disks to mark off-line and to receive the new generic data.

**3.16** In order to use the disk, SUPR makes state transitions. These transitions in the incore ECD include from active to out of service to off-line and then back again. For the ECD on disk, these transitions include from out of service to off-line and then back again.

**3.17** The off-line disk is not restored to service after a generic update since that would destroy the data.

**3.18** Off-line disks are easily accessible on a partition basis. Easily accessible means that read and write operations are performed in a manner similar to that for an active disk.

**3.19** The SUPR processes mark disks off-line in the incore ECD and record that state in the display administration process display.

**3.20** The SUPR processes update the ECD on an off-line disk. The old generic disk(s) must be marked off-line in the new generic ECD to protect the old generic data.

### Partition Copy

**3.21** The ability to "back up" partitions is an essential part of the backup partition method. However, "backing up" a partition implies making a copy of a partition that is in use. Since partitions in use may change, it is difficult to get a valid copy of a partition without having some way of holding the partition content constant for a period of time. Hence, SUPR needs a mechanism to copy partitions in that a snapshot of a partition can be taken.

**C. Application Requirement**

**3.22** The application is responsible for determining which update method to use for a given system update. Also, the application ensures that criteria for the update are satisfied.

**3.23** The application engineers the disk usage and makes proper preparations for the update to use the SUPR update facilities. The application performs application-dependent processing once for each step of the update procedure and supplies any software for application-dependent processing. If the application decides to update any process outside of the "hook" provided within SUPR, the application must furnish input message(s) for the craft to initiate these processes.

**3.24** The major responsibility of the application is to integrate the system update with the application services. This provides a smooth transition of services across generics.

**SYSTEM UPDATE PROCEDURES DESCRIPTION**

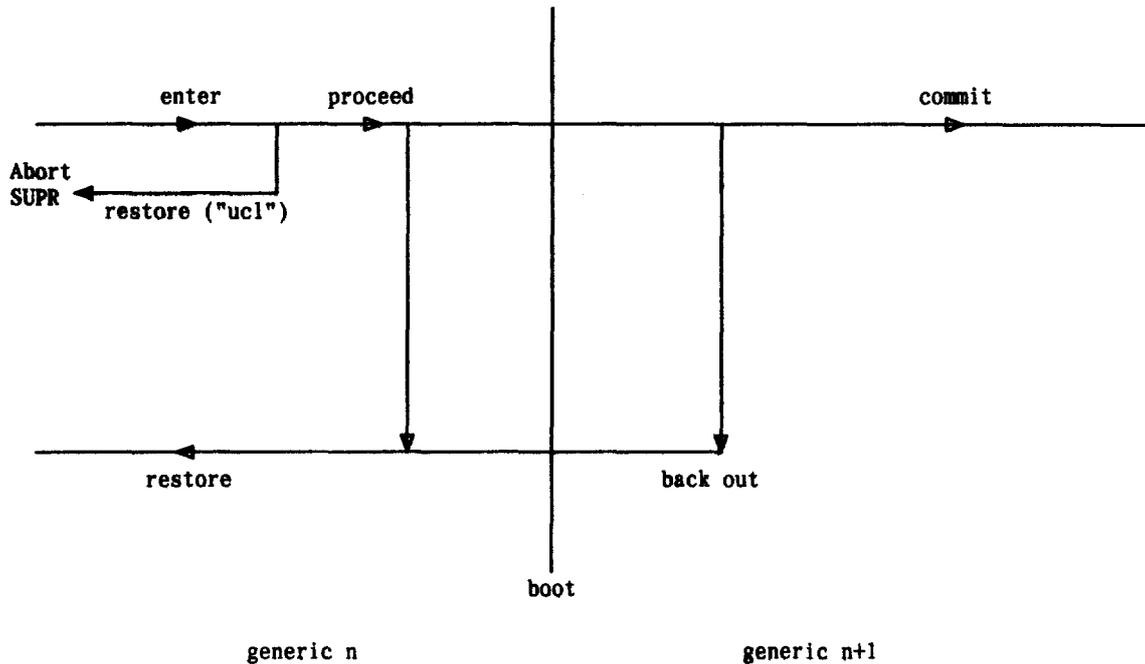
**3.25** The SUPR provides a separate process for each step of the update procedure. Applica-

tion-dependent processing is done by another process that is invoked by each SUPR process. The SUPR utilities also provide for additional interaction with SUPR. Figure 1 shows the normal command sequencing expected along with the correlation of each SUPR command to a generic.

**A. SUPR Enter Request**

**3.26** This procedure introduces the new generic into off-line disk or backup partitions. At the beginning of the update, both disks (of each duplex disk pair) must be active for the off-line disk method. The input message specifies update method, source (type unit) path name, and destination (volume table of contents) path name.

**3.27** The ENTER process tells the craft to mount tapes, to acknowledge each request, and to continue the update (using the "CONTINUE" utility). If not acknowledged within 30 minutes, the request is repeated once with an additional 30 minutes to wait. If still not acknowledged, the ENTER process stops with the error code for craft acknowledgment time-out. The ENTER process periodically reports its progress as number of blocks written.



**Fig. 1—System Update Scenario**

**B. SUPR Proceed Request**

**3.28** This procedure is used when the ENTER process is successful. The PROCEED process prepares for booting the system under the new generic. For the off-line disk method, this process prompts the craft to boot the system on the "primary root" of the off-line disk. For the backup partition method, this process prompts the craft to boot the system on the opposite "root" (the "root" the system was not up on originally).

**C. SUPR Commit Request**

**3.29** This procedure is used when the PROCEED process (and boot) is successful. Both disks (of each duplex disk pair) must be active for the backup-partition method. The COMMIT process copies the new generic over the old (onto the originally active disks or partitions). This completes the system update (new generic installed).

**D. SUPR Backout Request**

**3.30** This procedure is used when the new generic appears to be faulty. The BACKOUT process prepares for booting the system under the old generic. This process prompts the craft to boot the system on the originally active disk and "primary" or "backup" root (the "root" on which the system was booted before beginning the SUPR).

**Caution:** *If any process fails, use the "readlog" utility to output the SUPR log. If an SUPR process stops, the CONTINUE utility will try to restart the execution.*

**E. SUPR Restore Request**

**3.31** This procedure is used when BACKOUT (and boot) is successful or when ENTER or PROCEED is unsuccessful (using "UCL" option). A "DISK ENTER" entry (in the SUPR log) exists for every disk updated. The RESTORE process copies the old generic over the new (ie, onto the original off-line disk or "backup" partitions). This completes the system update (old generic remains installed).

**F. SUPR Utilities**

**3.32** In addition to the five SUPR processes used to implement the separate steps of the update procedure, three SUPR utilities provide for external

interactions with those SUPR processes. These utilities also provide assistance in troubleshooting.

**SUPR CONTINUE Utility**

**3.33** The CONTINUE utility allows craft interaction with SUPR processes. If an SUPR process requires craft actions to be coordinated with its processing, the process issues the appropriate instructions and then waits for an acknowledgment. If an SUPR process stops, the CONTINUE utility is used to restart the process at the last continuation point reached. The CONTINUE utility determines this from log information.

**3.34** When invoked, the CONTINUE utility sends an acknowledgment message to the SUPR process running. The message contains enough information for message verification, a stamp to ensure that it is a recent acknowledgment, and a special SUPR acknowledgment identifier to ensure that it is the correct acknowledgment. If an SUPR process stops, the CONTINUE utility tries to restart the execution.

**SUPR Stop Utility**

**3.35** The stop utility provides a means to gracefully halt an SUPR process. The stop utility sends a termination signal to the SUPR process running and waits to see if it terminates.

**3.36** Unless the SUPR process is insane, it terminates itself since each SUPR process catches both termination and interrupt signals. When the SUPR process receives such signals, it performs only final cleanup processing prior to terminating. This final processing is the same as that done if an error occurs. If the SUPR process fails to respond to the first termination signal, a second one is sent. What happens next if the SUPR process still fails to halt, depends on the execution mode (conditional or unconditional) of the stop utility. If the stop utility is executing conditionally, it will exit with an error code indicating the termination signal was ineffective. If the stop utility is executing unconditionally, it uses the **pkill** command to stop the SUPR process. When this happens, neither the SUPR process nor DMERT performs any clean up processing. Thus any reserved devices (such as a disk) may still be reserved.

### SUPR Readlog Utility

**3.37** The readlog utility provides human-readable information pertaining to the execution of SUPR processes. Each SUPR process makes several entries in the SUPR log file. In case of failure (or halting) of a step, the information in the log may be useful. The readlog utility outputs this information.

**3.38** The COMMIT and RESTORE processes automatically clear the SUPR log 3 minutes after either completes successfully. If the SUPR log output (see Fig. 2 for an example) is desired when such completion occurs, the log file `/etc/log/suprlog` must be copied to another location.

### SYSTEM UPDATE MENU PAGE DESCRIPTION

**3.39** The system update menu page, one of three different screens displayed on the maintenance terminal, is used as the interface between the user and the system update processes. The main screen is called the "System Update Menu Page." The other two screens are called "ENTER Arguments Page 1" and "ENTER Arguments Page 2." All actions are performed on the System Update Menu Page except specifying the ENTER command line arguments which are performed on the two "ENTER Arguments" screens. These screens allow the user to install system updates by executing commands with their appropriate arguments.

**3.40** Input commands that are executed using the system update menu page are initiated via pokes (a number or variable text string that is typed to request an operation displayed on the menu page). Some commands take optional and/or mandatory arguments as input. These arguments are specified in one of two ways: the user requests the arguments using pokes, or the user is prompted to input variable text strings. After a command and its arguments have been specified, the corresponding input command line appears on the maintenance terminal. While the command is executing, the input command remains displayed on the maintenance terminal until the command has completed its execution.

**3.41** During the system update, the menu page remains flexible. From the menu page, the craft can go to another page, execute nonsystem update commands, and then return to the menu page without losing the status of the update. If an SUPR process and/or the application process is executing

when another menu page is specified, the SUPR process and/or the application process along with the menu page will continue to execute. Also, the menu page may be terminated at any time as long as an SUPR process or the application process is not executing. If the menu page is terminated before the update was completed, the menu page may be brought back up with all of the update data still intact. The update is resumed from the point of termination.

**3.42** The SUPR interface screens provide a visual and user-friendly display for the execution of SUPR processes. On each of the three screens, a prompt (CMD<) is located in the upper left hand corner of the screen. A blinking underscore cursor is placed after the prompt to designate where to type the desired menu number(s). Below the "CMD<" line, a "Messages Line" will appear to report four types of messages: input errors made by the user, warning messages, instructions concerning system update procedures, and information indicating the status of the update. Error, warning, and procedural messages are displayed in reverse video. Status messages are displayed in normal video. The last two lines of each screen are reserved for the scrolling line (used to display output) and the manual input line (used to execute PDS/MML input commands), respectively.

**3.43** Each screen has a set of menu numbers that are typed by the user to request various actions. The user may request only the menu numbers that are displayed on the screen that is currently being viewed. Menu numbers from other screens not currently being viewed are considered to be invalid. There is an exception to this rule. If the cursor is positioned on the "CMD<" line, the user may request a menu number from the index menu page (page 100) to display a page other than the system update menu page.

#### A. System Update Menu Page Screen

**3.44** All actions except specifying the ENTER command line arguments are performed on this screen (Fig. 3 ). The menu numbers 701 through 750 are arranged on the screen according to function. The first three columns, numbers 701 through 709, executes the SUPR processes and the application process. The first three numbers (701, 702, 703) represent the three SUPR processes that must be executed to install a system update. They are listed in the order in which they must be executed. The fourth column,

```

OP GEN READLOG
FIRST
  ENTRIES: 19
BEGIN ENTER
  METHOD: OFLDISK
  EXECUTION: CONDITIONAL
  INPUT FILE: /dev/mt08
  DESTINATIONS: /dev/vtoc
  DFC NUMBER: 0
  SEQUENCE OPTION: DOUBLE
  ECD: /appecd
  ECD PARTITION: 16
  OLD LOG: /etc/log/suprlog
  OLD LOG PARTITION: 14
  NEW LOG: /etc/log/suprlog
  NEW LOG PARTITION: 13
  PID: 196665
  Mon Sep 9 13:52:22 1982
DISK ENTER
  NEW GENERIC: MHD 0
  OLD GENERIC: MHD 1
  DISK PAIR: 1
  PID: 196665
  Mon Sep 9 13:55:16 1982
TAPE ENTER
  SEQUENCE NUMBER: 0
  SG ID: Mon Sep 4 13:43:01 1982
  GENERIC-VERSION: 2.1
  SEQUENCE: FIRST
  PID: 196665
  Mon Sep 9 13:55:17 1982
WRITE ENTER /* vtoc */
  PARTITION: 0
  BACKUP PARTITION: 0
  PID: 196665
  Mon Sep 9 13:55:19 1982
WRITE ENTER /* other partitions */
.
.
.
APPLHOOK ENTER
  PID: 196665
  Mon Sep 9 14:03:27 1982
END ENTER
  FILE SYSTEM: BACKUP
  BOOT FILE: APPDMERT
  BOOT DISK: MHD 1
  COMPLETION: SUCCESS
  ERROR CODE: 0
  UNIX ERROR NUMBER: 0
  CONTINUABLE: NO
  PID: 196665
  Mon Sep 9 14:43:54 1982

OP GEN READLOG
BEGIN PROCEED
  METHOD: OFLDISK
  EXECUTION: CONDITIONAL
  PID: 262201
  Mon Sep 9 14:44:36 1982
APPLHOOK PROCEED
  PID: 262201
  Mon Sep 9 14:46:52 1982
END PROCEED
  FILE SYSTEM: BACKUP
  BOOT FILE: APPDMERT
  BOOT DISK: MHD 1
  COMPLETION: SUCCESS
  ERROR CODE: 0
  UNIX ERROR NUMBER: 0
  CONTINUABLE: NO
  PID: 262201
  Mon Sep 9 14:52:43 1982
BEGIN COMMIT
  METHOD: OFLDISK
  EXECUTION: CONDITIONAL
  PID: 293446
  Mon Sep 9 16:26:39 1982
DISK COMMIT
  NEW GENERIC: MHD 1
  DISK PAIR: 1
  PID: 293446
  Mon Sep 9 16:31:45 1982
APPLHOOK COMMIT
  PID: 293446
  Mon Sep 9 16:32:03 1982
END COMMIT
  FILE SYSTEM: PRIMARY
  BOOT FILE: APPDMERT
  BOOT DISK: MHD 0
  COMPLETION: SUCCESS
  ERROR CODE: 0
  UNIX ERROR NUMBER: 0
  CONTINUABLE: NO
  PID: 293446
  Mon Sep 9 16:32:19 1982
OP GEN READLOG COMPLETED

```

Fig. 2—SUPR Log Output

numbers 720 through 735, are affiliated with the SUPR processes and/or the application process. Numbers 720 and 721 represent arguments that may be desired on some of the SUPR process command lines. Number 735 requests that an SUPR process or the application process be executed. In the fifth column, numbers 740 through 750, perform actions concerning SUPR interface and its screens. For example, number 740 displays the ENTER Arguments Page 1 screen and number 741 displays the ENTER Arguments Page 2 screen. Number 750 terminates the system update menu page.

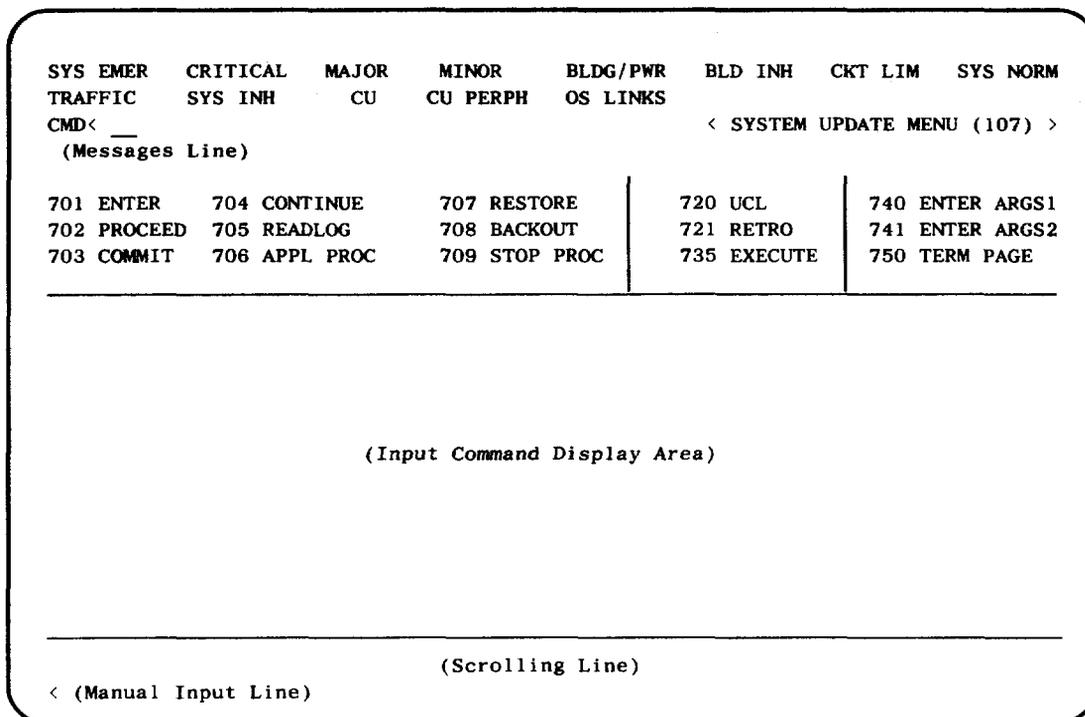
**3.45** The input command lines corresponding to the SUPR processes and the application process, that are requested using menu numbers 701 through 709, are displayed in the "Input Command Display Area."

**B. ENTER Arguments Page 1 Screen**

**3.46** The menu numbers 760 through 764 request arguments that are to be specified on the ENTER command line (Fig. 4). All ENTER command

line arguments are specified on this screen except the disks to be updated and their corresponding tape sequence options which are specified on the ENTER Arguments Page 2 screen. To the right of the menu numbers and their descriptions are the possible values that may be specified for each argument along with their default values (if any). Between the menu descriptions and their possible values, input fields are reserved for the user to specify the argument values. When a poke to an argument is requested, an underline is drawn at the appropriate input field. The length of the underline visually represents the maximum length allowed for that argument value. A blinking underscore cursor is placed at the beginning of the underline to indicate where to input the desired argument value. For the tape device argument, the underline is drawn one line below the description.

**3.47** The menu numbers 792 and 793 control paging. Number 792 displays the ENTER Arguments Page 2 screen. Number 793 displays the System Update Menu Page screen and automatically writes the ENTER input command line into the display area.



◆Fig. 3—System Update Menu Page Screen◆

SYS EMER	CRITICAL	MAJOR	MINOR	BLDG/PWR	BLD INH	CKT LIM	SYS NORM
TRAFFIC	SYS INH	CU	CU PERPH	OS LINKS			
CMD< _				< ENTER ARGUMENTS PAGE 1 of 2 >			
(Messages Line)							
760	UPDATE METHOD:			(B = BACKUP PARTITION, 0 = OFF-LINE DISK)			
761	DFC PREFERENCE:			(0 = DFC0, 1 = DFC1, 2 = NO PREF <DEFAULT>)			
762	UPDATE BACKUP PARTITIONS:			(Y = YES <DEFAULT>, N = NO)			
763	UNCONDITIONAL EXECUTION:			(N = NO <DEFAULT>, Y = YES)			
764	TAPE DEVICE:			(SPECIFY FULL PATHNAME)			
-----							
792	DISPLAY ENTER ARGUMENTS PAGE 2			(SPECIFY DISKS AND SEQUENCE OPTIONS)			
793	DISPLAY SYSTEM UPDATE MENU PAGE			(VIEW ENTER INPUT COMMAND LINE)			
-----							
(Scrolling Line)							
< (Manual Input Line)							

Fig. 4—ENTER Arguments Page 1 Screen

### C. ENTER Arguments Page 2 Screen

**3.48** Near the top of the screen, DEST and SEQ OPT are defined and their possible argument values are specified (Fig. 5).

**3.49** The menu numbers 770 through 779 request the disks that are to be updated. The menu numbers 780 through 789 request their corresponding sequence options. After one of these arguments has been requested, an underline is drawn immediately after its description. A blinking underscore cursor at the beginning of the line indicates where to input the desired argument value. The length of the line visually represents the maximum length allowed for that argument value.

**3.50** The menu numbers 791 and 793 control paging. Number 791 displays the ENTER Arguments Page 1 screen. Number 793 displays the System Update Menu Page screen and automatically writes the ENTER input command line into the display area.

### SYSTEM UPDATE MESSAGE AND ERROR CODES

#### A. Input Messages

**3.51** The input messages for the SUPR fall into three categories: SUPR command invocation, craft response, and craft intervention.

**3.52** The SUPR command invocation category contains input messages that invoke each step of the SUPR. The craft is allowed to initiate the appropriate SUPR process using one of these input messages.

**3.53** The craft response category includes those messages provided for the craft to acknowledge requests from SUPR for manual action. The craft uses these messages to indicate that a request from SUPR has been carried out.

**3.54** The craft intervention category includes input messages available for the craft to halt any SUPR process in progress. The halt request must correspond to a SUPR command invocation message

```

SYS EMER   CRITICAL MAJOR   MINOR   BLDG/PWR BLD INH   CKT LIM   SYS NORM
TRAFFIC    SYS INH   CU     CU PERPH OS LINK
CMD<  _____ < ENTER ARGUMENTS PAGE 2 of 2 >
      (Messages Line)

DEST =     DESTINATION - DISK TO BE UPDATED (FULL PATHNAME)
SEQ OPT =  SEQUENCE OPTION FOR DESTINATION (S = SINGLE, D = DOUBLE, T = TRIPLE)

770 DEST:                               780 SEQ OPT:
771 DEST:                               781 SEQ OPT:
772 DEST:                               782 SEQ OPT:
773 DEST:                               783 SEQ OPT:
774 DEST:                               784 SEQ OPT:
775 DEST:                               785 SEQ OPT:
776 DEST:                               786 SEQ OPT:
777 DEST:                               787 SEQ OPT:
778 DEST:                               788 SEQ OPT:
779 DEST:                               789 SEQ OPT:

-----
791 DISPLAY ENTER ARGUMENTS PAGE 1      (SPECIFY OTHER ENTER ARGUMENTS)
793 DISPLAY SYSTEM UPDATE MENU PAGE    (VIEW ENTER INPUT COMMAND LINE)
-----

      (Scrolling Line)

< (Manual Input Line)

```

◆Fig. 5—ENTER Arguments Page 2 Screen◆

input earlier. The SUPR input messages are shown in Figure 6.

**B. Output Messages**

3.55 The SUPR output messages are issued by SUPR processes or utilities in response to

SUPR input messages. These output messages are grouped into four categories: completion information, craft requests, status reports, and requested log output. Also, an input message is provided that allows the craft to output the contents of the SUPR log file in a readable form. This capability is generally used only if an SUPR process failed or stopped. To

```

UPD: GEN; ENTER [SGLSEQ|DBLSEQ|TPLSEQ] [ ; UCL ] : {BKPRT|OFLDISK [ a
      [ , NOBKUP ] } , SRC "b" , {DEST "c" | DEST ("c" [ , "d" ] . . . ) } !
UPD: GEN; PROCEED [ ; UCL ] !
UPD: GEN; COMMIT [ ; UCL ] !
UPD: GEN; BACKOUT [ ; UCL ] !
UPD: GEN; RESTORE [ ; UCL ] !
UPD: GEN; CONTINUE !
STOP: GEN [ ; UCL ] !
OP: GEN; READLOG !

```

Fig. 6—SUPR PDS Input Messages

determine appropriate action about executed steps, more information is needed.

**3.56** The completion information category consists of two completion messages (success/failure). One of these output messages is issued when an SUPR process or utility completes.

**3.57** The craft request category consists of output messages that request a craft manual action. If a manual action is required during any step of the update procedure, an output message instructs the craft which actions to take.

**3.58** The status report category is made up of output messages that convey periodic status information to the craft concerning update processing progress.

**3.59** The requested log output category is made up of output messages containing lines of text displaying the log entries. These output messages are generated in response to the craft requesting log output.

**3.60** The SUPR output messages are shown in Figure 7. These output messages are sent to all maintenance terminals, to the terminal issuing the SUPR input message, and to the receive only printer (ROP). Messages

### C. Error Codes

**3.61** When any SUPR process fails, it performs final cleanup processing and then issues an error code as part of the completion message to help pinpoint the trouble. A six decimal digit error code is printed on the receive only printer (ROP) to identify the problem. The first three digits indicate the nature of the error. The next two digits (xx) indicate the module in which the error occurred. The last digit (the error number, y) is used to differentiate errors of the same error code that can occur multiple times in the same module. To locate exactly which error in the module failed, each error is given its own unique error number. The following paragraphs lists the six types of errors which can cause an SUPR process to fail.

**3.62** All SUPR processes run as UNIX operating system user processes. If any UNIX operating system call fails, SUPR fails and outputs an error code and error number. The error code indicates the

error type as a UNIX system call error. The error number identifies the specific system call that failed. The UNIX system calls are used mostly for data transferral and for message communication with other processes. If the system is stable, UNIX system call errors should not occur.

**3.63** Data errors are caused by inconsistent new generic data. If SUPR encounters abnormalities in the new generic data, it fails with an error code indicating the error type as a data error and with the error number identifying the problem found in the data. These errors generally indicate the use of faulty new generic input.

**3.64** All system update errors indicate that some internal SUPR data was garbled. When SUPR encounters an erroneous piece of internal data, it fails and outputs an error code and error number. The error code indicates the error type as a system update error. The error number identifies the specific data causing the error. The SUPR internal errors should not occur. If they do, it is possible that incompatible versions of SUPR processes are being used or that the SUPR log file is corrupt.

**3.65** Execution errors are caused by mistakes in using the SUPR processes. Note that manually halting an SUPR process with the "stop" utility causes an "execution error." If an execution error occurs, SUPR fails with an error code indicating the error type as an execution error and with the error number identifying the mistake. The errors are listed by error type and error number only.

**3.66** The SUPR processes interface with the application only through the application "hook," a UNIX operating system user process. If the application process fails to set a time limit for its processing or if the application process terminates unsuccessfully, SUPR fails. An error code follows indicating the error type as an application error, while the error number associated with the error code specifies the error.

**3.67** The SUPR processes interface with various areas of DMERT outside the jurisdiction of its UNIX operating system supervisor. If any request made to another part of DMERT fails, SUPR fails and outputs an error code and error number. The error code designates the error type as a DMERT error. The error number identifies the specific DMERT request that failed. The DMERT error category

UPD GEN ENTER	MOUNT TAPE AND CONTINUE
UPD GEN ENTER	MHD x SELECTED FOR NEW GENERIC
UPD GEN ENTER	CHANGING MHD x TO OFFLINE
UPD GEN ENTER	MHD x OFFLINE
UPD GEN ENTER	n BLOCKS WRITTEN
UPD GEN ENTER	TRANSFERRING CONTROL TO APPLICATION
UPD GEN ENTER	APPLICATION COMPLETION EXPECTED WITHIN n SECONDS
UPD GEN ENTER	UPDATING PARTITION b IN MHD x
UPD GEN ENTER	COMPLETED
UPD GEN ENTER	STOPPED WITH ERROR CODE xxxxx
UPD GEN PROCEED	TRANSFERRING CONTROL TO APPLICATION
UPD GEN PROCEED	APPLICATION COMPLETION EXPECTED WITHIN n SECONDS
UPD GEN PROCEED	UPON COMPLETION BOOT SYSTEM SELECTING PRIMARY ROOT
UPD GEN PROCEED	UPON COMPLETION BOOT SYSTEM SELECTING BACKUP ROOT
UPD GEN PROCEED	UPON COMPLETION BOOT SYSTEM SELECTING MHD { 011}
UPD GEN PROCEED	COMPLETED
UPD GEN PROCEED	STOPPED WITH ERROR CODE xxxxx
UPD GEN COMMIT	TRANSFERRING CONTROL TO APPLICATION
UPD GEN COMMIT	APPLICATION COMPLETION EXPECTED WITHIN n SECONDS
UPD GEN COMMIT	COMPLETED
UPD GEN COMMIT	STOPPED WITH ERROR CODE xxxxx
UPD GEN BACKOUT	TRANSFERRING CONTROL TO APPLICATION
UPD GEN BACKOUT	APPLICATION COMPLETION EXPECTED WITHIN n SECONDS
UPD GEN BACKOUT	UPON COMPLETION BOOT SYSTEM SELECTING PRIMARY ROOT
UPD GEN BACKOUT	UPON COMPLETION BOOT SYSTEM SELECTING BACKUP ROOT
UPD GEN BACKOUT	UPON COMPLETION BOOT SYSTEM SELECTING MHD { 011}
UPD GEN BACKOUT	COMPLETED
UPD GEN BACKOUT	STOPPED WITH ERROR CODE xxxxx
UPD GEN RESTORE	TRANSFERRING CONTROL TO APPLICATION
UPD GEN RESTORE	APPLICATION COMPLETION EXPECTED WITHIN n SECONDS
UPD GEN RESTORE	COMPLETED
UPD GEN RESTORE	STOPPED WITH ERROR CODE xxxxx
UPD GEN CONTINUE	COMPLETED
UPD GEN CONTINUE	STOPPED WITH ERROR CODE xxxxx
STOP GEN	COMPLETED
STOP GEN	STOPPED WITH ERROR CODE xxxxx

Fig. 7—SUPR PDS Output Messages

includes errors that develop when interfacing with the following:

- Disk driver
- Recent change and verify system
- System integrity monitor
- Maintenance interface request administrator
- ECD manager.

If the system is stable, DMERT errors should not occur.

**4. ABBREVIATIONS AND ACRONYMS**

4.01 The following is a list of abbreviations and acronyms used in this section.

<b>ABBREVIATIONS</b>	<b>TERMS</b>
DFC	Disk File Controller
DMERT	Duplex Multienvironment Real-Time
ECD	Equipment Configuration Database

**ABBREVIATIONS**

**TERMS**

ECDMAN	Equipment Configuration Database Manager
ID	Identification
LDFT	Load Disk From Tape
MIRA	Maintenance Input Request Administrator
ROP	Receive Only Printer
SG	System Generation
SIM	System Integrity Monitor
SUPR	System Update Procedure
UCL	Unconditional
VTOC	Volume Table of Contents