ULT

SYSTEM MANAGEMENT AND SUPPORT

THE ULTIMATE CORP.

SECTION 1

SYSTEM CONCEPTS AND COMPONENTS

SECTION 1

SYSTEM CONCEPTS AND COMPONENTS (continued)

1.1 OVERVIEW OF ULTIMATE'S VIRTUAL MEMORY SYSTEM

The Ultimate operating system is designed to run on a virtual machine, in which the virtual memory (on disk) is directly addressable as if it were in main (RAM) memory. The system architecture comprises, depending on the implementation, a combination of hardware, firmware, and software. This architecture is designed to optimize the data base and data manipulation capabilities and to implement a virtual memory management system at the machine level.

The virtual memory of an Ultimate system resides on disk. The disk is organized into blocks of 512 bytes each, called "frames". Frames are identified by frame-id (FID) numbers, from 1 to the maximum FID (MAXFID); the value of MAXFID depends on the size of the disk(s). Figure A shows the typical layout of a disk subsystem immediately after a file restore; that is, in an initialized condition.

ABS Section

The first 1024 frames of the disk are reserved for the Ultimate operating system and user assembly code. This section is the ABS section of the disk; it consists of executable assembly object code supplied by Ultimate. Users may modify only the frames reserved for user assembly code.

PROCESS Workspace Section

The initial PROCESS workspace section follows the ABS section in virtual memory. The size of this section depends on the number of ports (terminals) on the system. Each terminal line requires a separate workspace, as do the Spooler and Warmstart Processes, and the two UltiNet Processes (see Figure B). This allows multi-programming, whereby multiple programs can run at the same time, sharing CPU resources. Each Process workspace is initially allocated 416 total frames (32 contiguous frames of primary workspace plus 384 linked frames of secondary workspace), as shown in Figure C.

Additional Process workspaces are obtained as needed from the Available Space (Overflow) pool, such as for processing system commands that have a BASIC EXECUTE statement. The total disk space allocated to Process workspaces is dependent on the system configuration. For example, systems with UltiNet have one or two additional phantom process workspaces allocated. (Space is reserved on all systems, but not used unless UltiNet is installed.)

FILES Section

The FILES Section follows the (initial) PROCESS section. The first file is the SYSTEM file. The frame at which the SYSTEM file begins is called SYSBASE. The SYSTEM file is followed by the account master dictionaries, the file dictionaries, and the data file areas.

Available Space (Overflow) Section

The disk space between the last user file and the maximum frame-id (MAXFID) is the Available Space or Overflow section.

Figure D shows the virtual memory FID layout and contents of the major sections of the disk. Frames 1-399 and 600-1023 are reserved for Ultimate system software. Frames 400-599 are available for user-written assembly language programs. Initially (at logon time), frame 1024 (400 hex) is normally the first frame of line 0's Primary Workspace (32 contiguous frames). The first workspace frame contains the line's Process Control Block (PCB). PCBs for succeeding processes are at 32-frame intervals; line 1's PCB is at frame 1056 (420 hex), line 2 is at frame 1088 (440 hex), etc. The linked secondary workspaces begin at the frame called WSSTART:

1024 + (#processes * 32) = WSSTART

and continues until all Processes have been allocated their Secondary Workspaces (384 frames each).

NOTE: Once a user logs onto a line, software may cause a transfer to different workspace frames, depending on the processing required. When a user logs off, the system resets the line to use the reserved initial workspace for that line.

The FILES section begins at the frame called SYSBASE, the base of the SYSTEM file. Its frame-id (FID) can be calculated as follows:

1024 + (#processes * 416) = SYSBASE

For example, a system with 16 communications lines plus the UltiNet File Server and Communications Server, the Phantom line and the Spooler line (20 total Processes) will have as the starting FID of its FILES section:

1024 + (20 * 416) = 9056 (SYSBASE)

On a newly regenerated or restored system, all other files on the system immediately follow the SYSTEM file. Also, the Available Space frames are a contiguous block which continues to the last usable FID (MAXFID). On a running system, the Available Space "pool" becomes fragmented into a series of linked as well as contiguous blocks, as frames are taken from and returned to the pool.

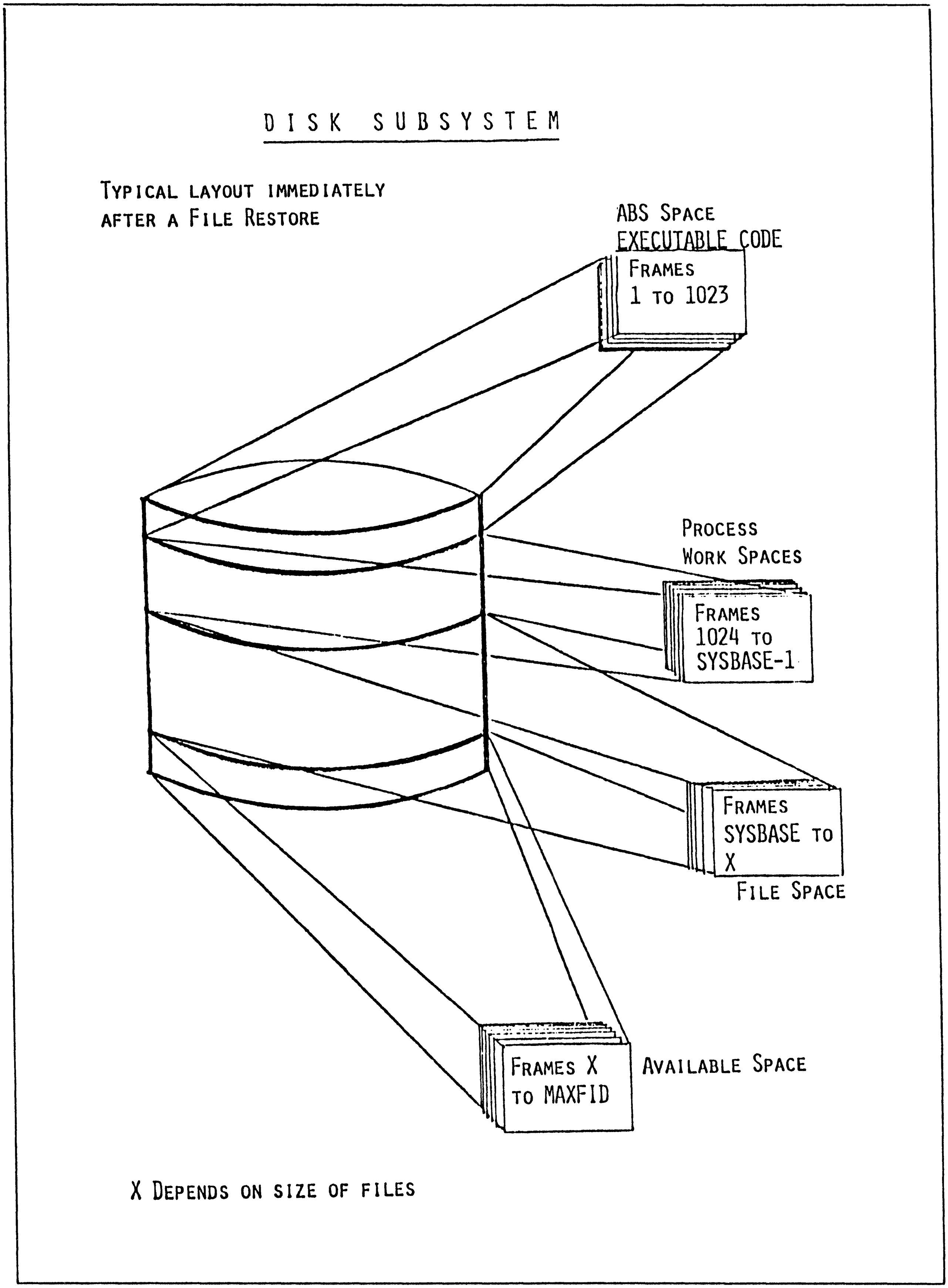

Figure A. Layout of Disk Subsystem After File Restore

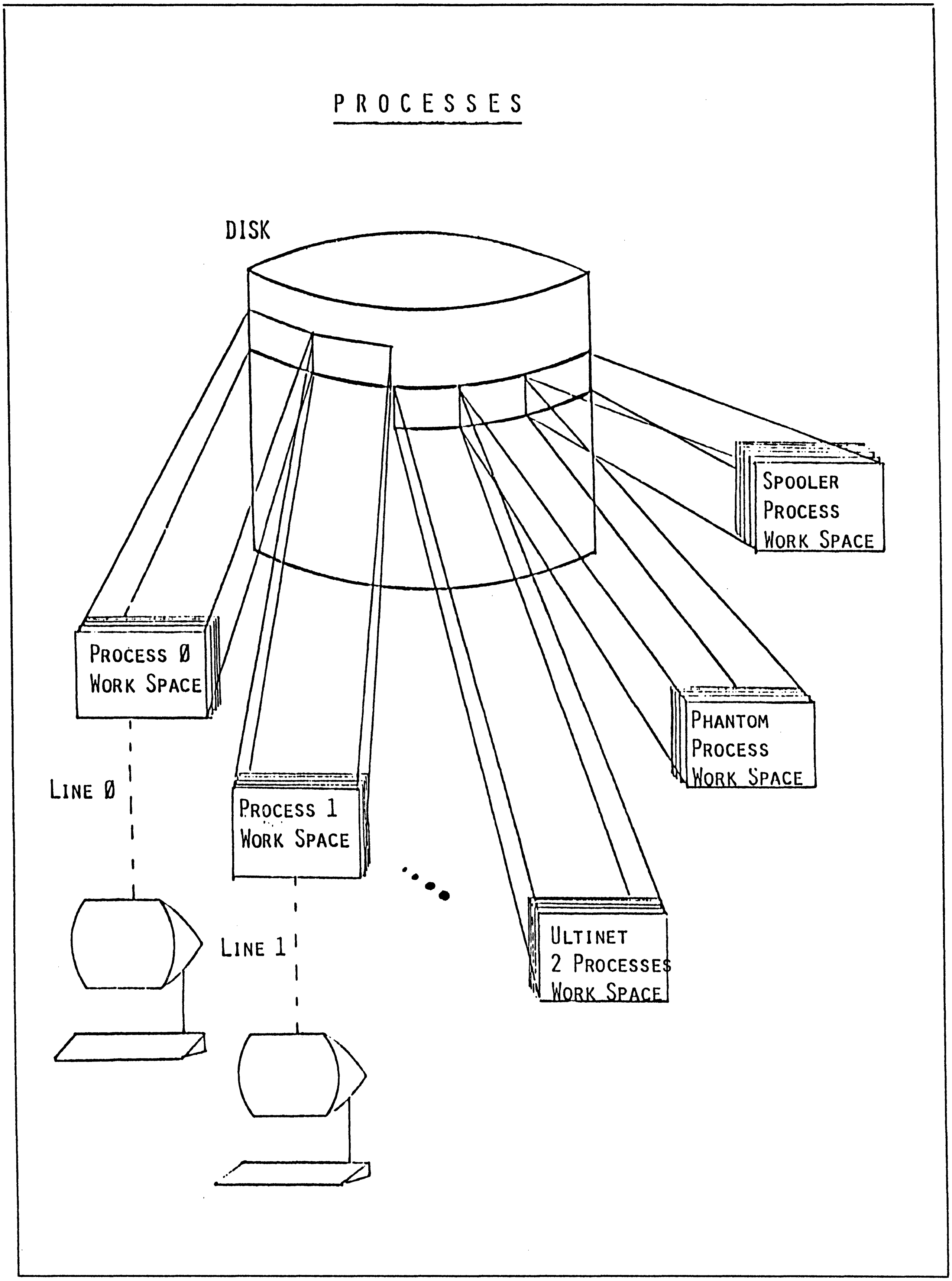

Figure B. Processes

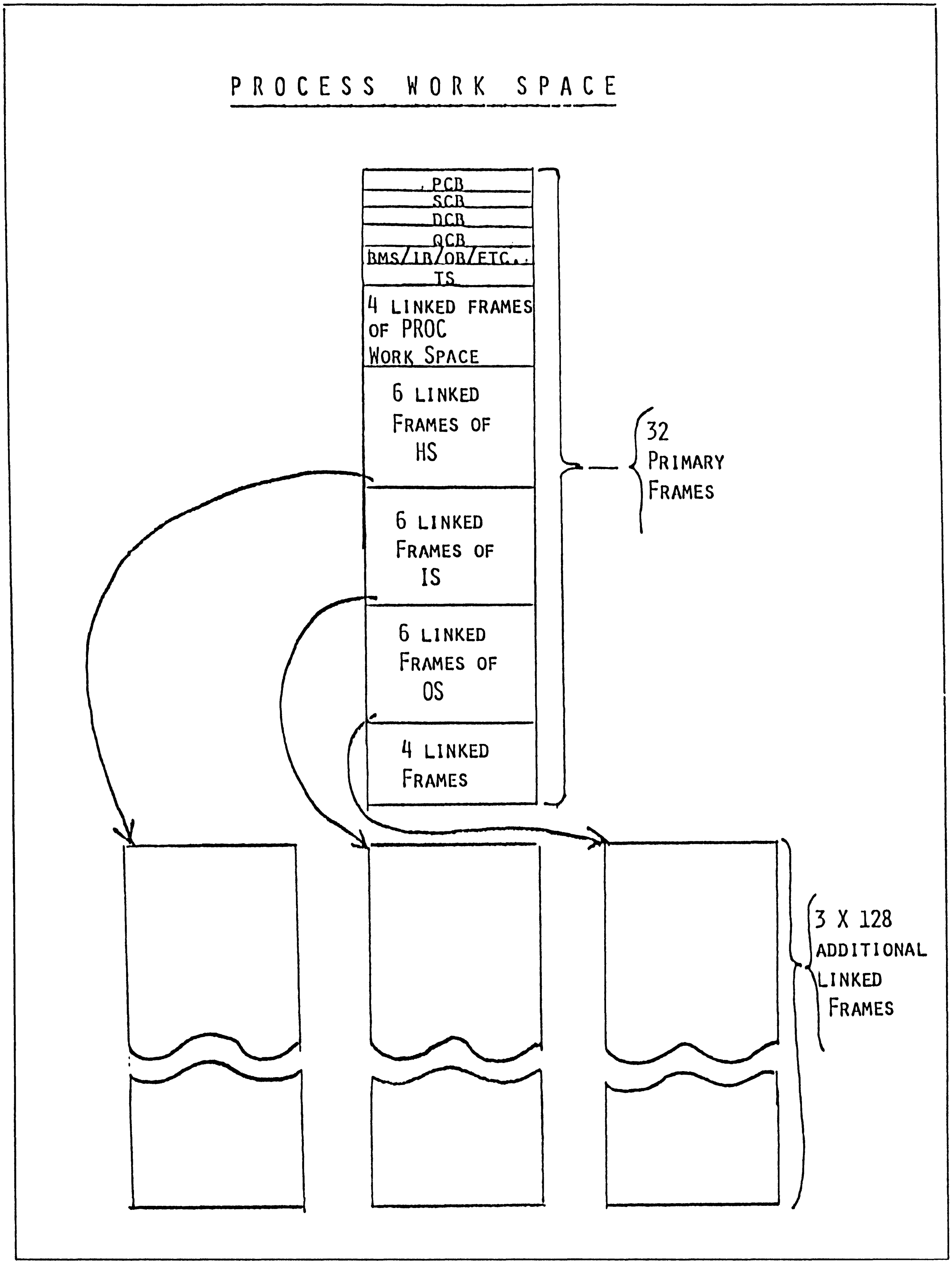

Figure C. Process Workspaces

```
Frame                  Description
                                                              +
1-399                  Operating System Software              |
                                                              A
400-599                User Assembly Code                     B
                                                              S
600-1023               Operating System Software              |
                                                              +
-------------------------------------------------------------
                                                              +
1024-1055              Line 0 PCB & Primary Workspace         |
1056-1087              Line 1 PCB & Primary Workspace         W
                       .                                      O
                       .                                      R
                       .                                      K
                       Spooler PCB & Primary Workspace
(lines+4) * 32 =       Line 0 Secondary Workspace             S
                       Line 1 Secondary Workspace             P
                       .                                      A
                       .                                      C
                       .                                      E
                       .                                      |
                       Spooler Secondary Workspace            +
-------------------------------------------------------------
                                                              +
                       'SYSTEM'                               |
                          'ACC'                               |
                          'BLOCK-CONVERT'                     |
                          'GAMES'                             F
                          'PROCLIB'                           I
                          'SYSTEM-ERRORS'                     L
                          'SYSPROG'                           E
                          'ERRMSG'                            S
                          'NEWAC'                             |
                          'SYSPROG-PL'                        |
                          'STAT-FILE'                         |
                                                              |
                       User Account                           |
                         Program                              |
                         Proc                                 |
                         Dictionary                           |
                           Data                               +
-------------------------------------------------------------

  |                   Free Space
  |
  v
-maxfid
```

Figure D. Virtual Memory Layout

1.2 SOFTWARE COMPONENTS

The Ultimate system typically comprises hardware, firmware, and software running a multi-user virtual memory operating system. In most configurations the Ultimate system controls the computer, and requires no host software. In other cases, such as on the DEC VAX computer, the Ultimate system runs in conjunction with a host operating system.

The Ultimate system software can be divided into two main components: virtual code (such as the ABS code), and monitor code, called the Kernel.

Virtual code is executed by virtual Processes. A Process may be defined as an ongoing task that executes a sequence of instructions, and is typically associated with a particular terminal or line. Monitor code is not associated with a particular line; it manages the execution of the virtual Processes, and interacts with the computer hardware (or host operating system).

Virtual code resides in virtual memory and is brought in from disk as needed, whereas monitor code resides permanently in main memory. Figure A shows a diagram of the main memory layout.

The monitor code (Kernel) is the program responsible for Process scheduling, memory management, disk scheduling, input/output, interrupt handling, and execution of Monitor calls. A Process makes Monitor calls when the Process does not have the capability of performing a certain small task on its own. Then it calls on the Monitor to perform the task, such as getting the current time or current status of the tape drive.

A Process Identification Block (PIB) is an element in the fixed portion of Main Memory. PIBs contain the current status of a Process, the FID of its Primary Control Block (PCB), and pointers to its Terminal I/O Buffer.

The Buffer Table is an element in the fixed portion of Main Memory. It identifies which frames are currently in the Virtual portion of Main Memory and their memory address location.

A Terminal I/O Buffer is an element in the fixed portion of Main Memory. Input Buffers hold the characters that a user types on the keyboard of a terminal. Output Buffers hold the characters to be displayed prior to the actual screen display.

The Virtual portion is divided into 512-byte buffers (each equal to one disk frame). When a Process begins executing, the disk frames containing its Process workspace are accessed, as needed, and copied into the Virtual buffers. Such access is made possible by a "frame fault", which alerts the system that a Process has attempted to use a frame which is currently not in main memory (see Figure B). Once a frame is in main memory, it is accessible to all Processes; no Process is given its own copy.

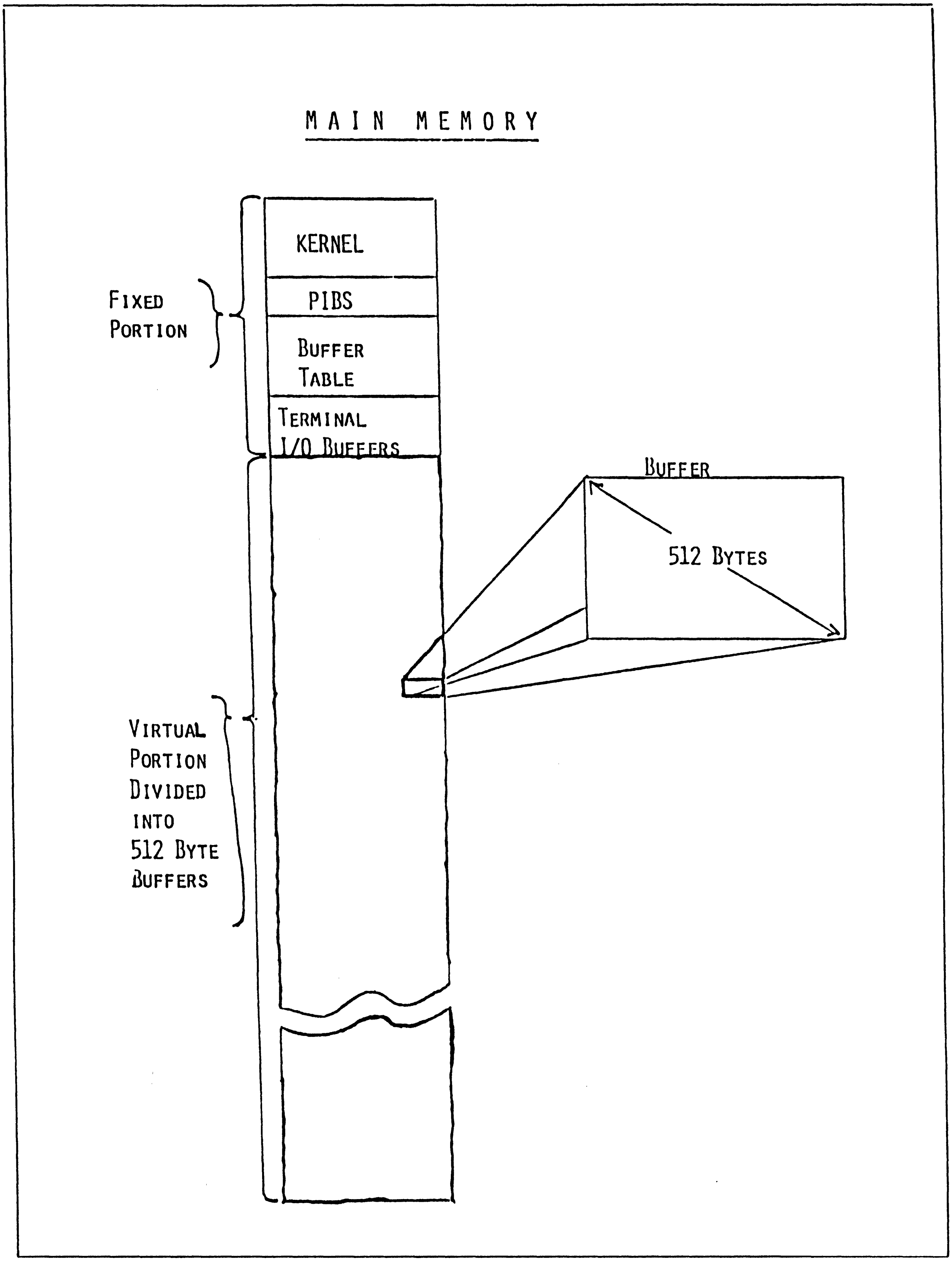

Figure A. Main Memory

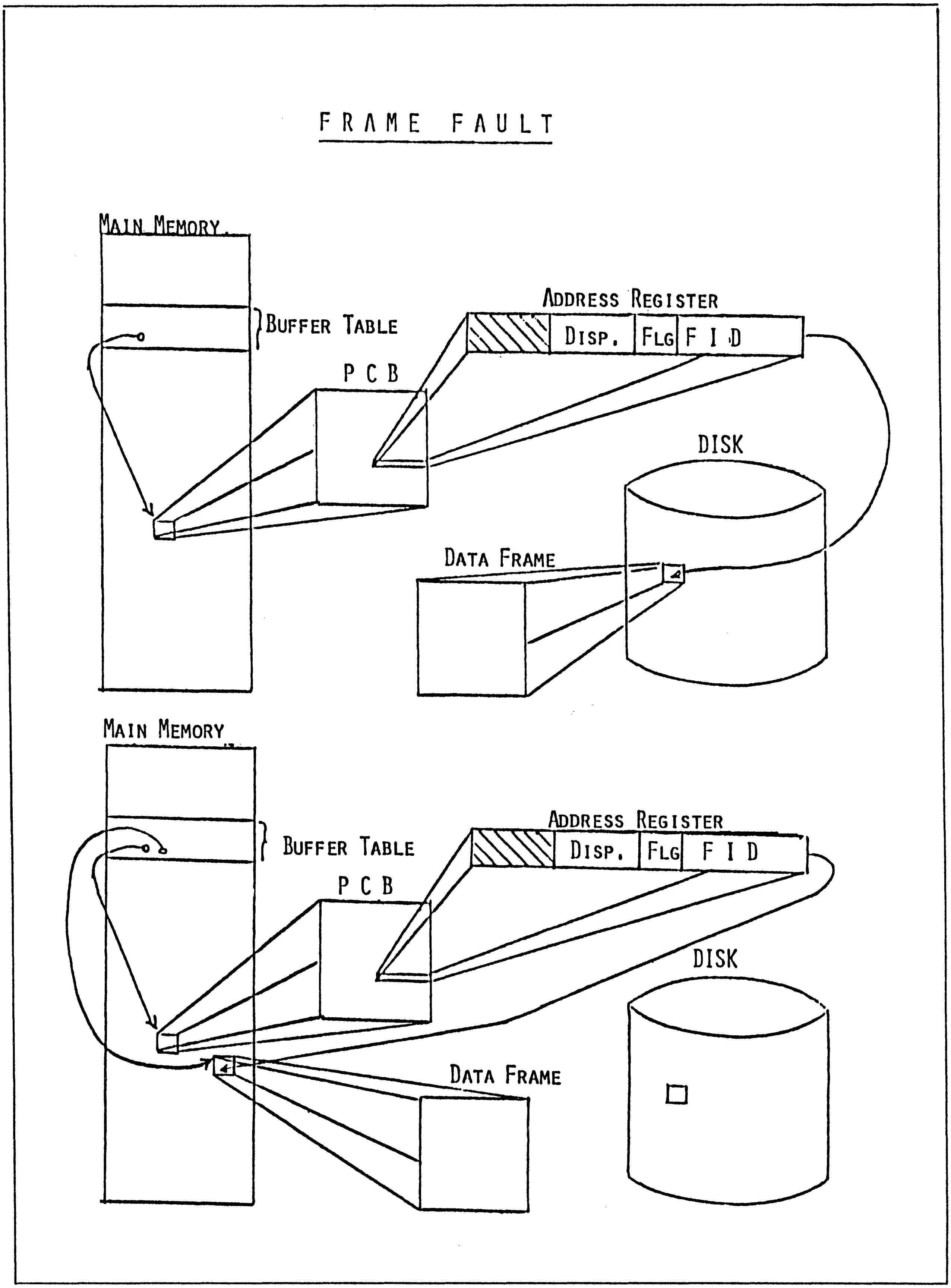

Figure B. Frame Fault

1.3 FRAME FORMATS AND THE ALLOCATION OF AVAILABLE SPACE

As data areas are needed to perform system operations and user tasks, virtual memory space is allocated frame-by-frame to the Process requesting the space. Each 512-byte frame has either a linked or unlinked format, depending on its use. A frame is always referenced by its frame-id (FID).

Frames may have a linked or unlinked format. Figure A shows these two formats and their associated address register format (address registers are discussed below). Linked frames are used to define data areas which may be greater than one frame in length. The data bytes in a frame (of both types) have no pre-defined format.

Frames in the ABS section of the disk subsystem have an unlinked format and may contain object-code programs. Frames in the PROCESS section are linked and unlinked; an example of a Process workspace format is shown in Figure B. The first frame in the workspace is the Primary Control Block (PCB), shown in Figure C.

The PCB maintains all address registers for the Process. These registers function as pointers to other specific bytes within frames throughout the main and virtual memory system, thereby controlling the Process operations. The address register field format is shown in Figure D.

All frames in the FILES section are in the linked format. A set of linked frames in a file is called a "group". Groups may expand as data files expand. When the end of a frame is reached, another frame is obtained from the Available Space pool and linked to the end of the group. The first 12 bytes of a linked frame contain (in hexadecimal):

Byte	0	Reserved byte.
	1	(NNCF) Number of Next Contiguous Frames; the count of frames whose FIDs are sequentially linked forward from this one.
	2-5	FID of next (logically) forward linked frame.
	6-9	FID of previous (logically) backward linked frame.
	A	(NPCF) Number of Previous Contiguous Frames; the count of frames whose FIDs are sequentially linked backward from this one.
	B	Reserved byte.

The first frame of a linked set of frames has an NPCF value of zero (0), and zero as the backward link FID. The last frame of such a set has an NNCF value of zero (0), and zero as the forward link FID.

The NNCF and NPCF fields are normally zero, except in the linked workspace allocated to each Process, and in files that have a separation greater than one. (A file's "separation" is the number of frames per group, which is usually one.)

NOTE: After a file restore, there is one principal block of contiguous Available Space. As the system obtains and releases frames from the Available Space pool (and as files are created and deleted) the Available Space becomes fragmented into blocks of contiguous frames and chains of linked frames.

Available frames will be placed in the linked Available chain only when there are 31 sets of contiguous Available Space (representing the maximum that the system space management routines can maintain).

For suggested procedures for checking the system's Available Space pool, see Sections 3 and 4 of this manual, which cover system restore procedures and user account and file creation procedures. See also the POVF command in the System Commands Guide.

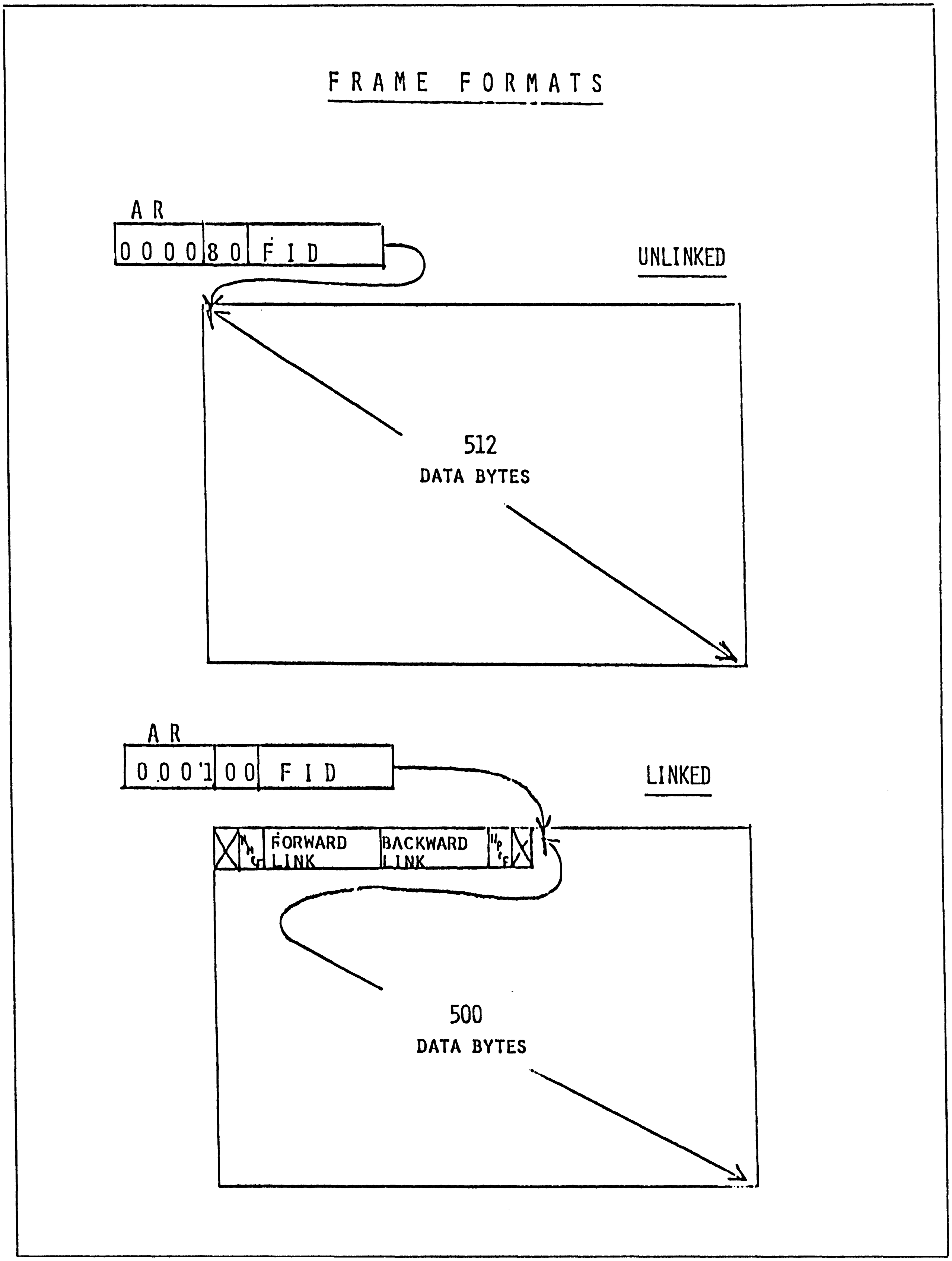

Figure A. Frame Formats

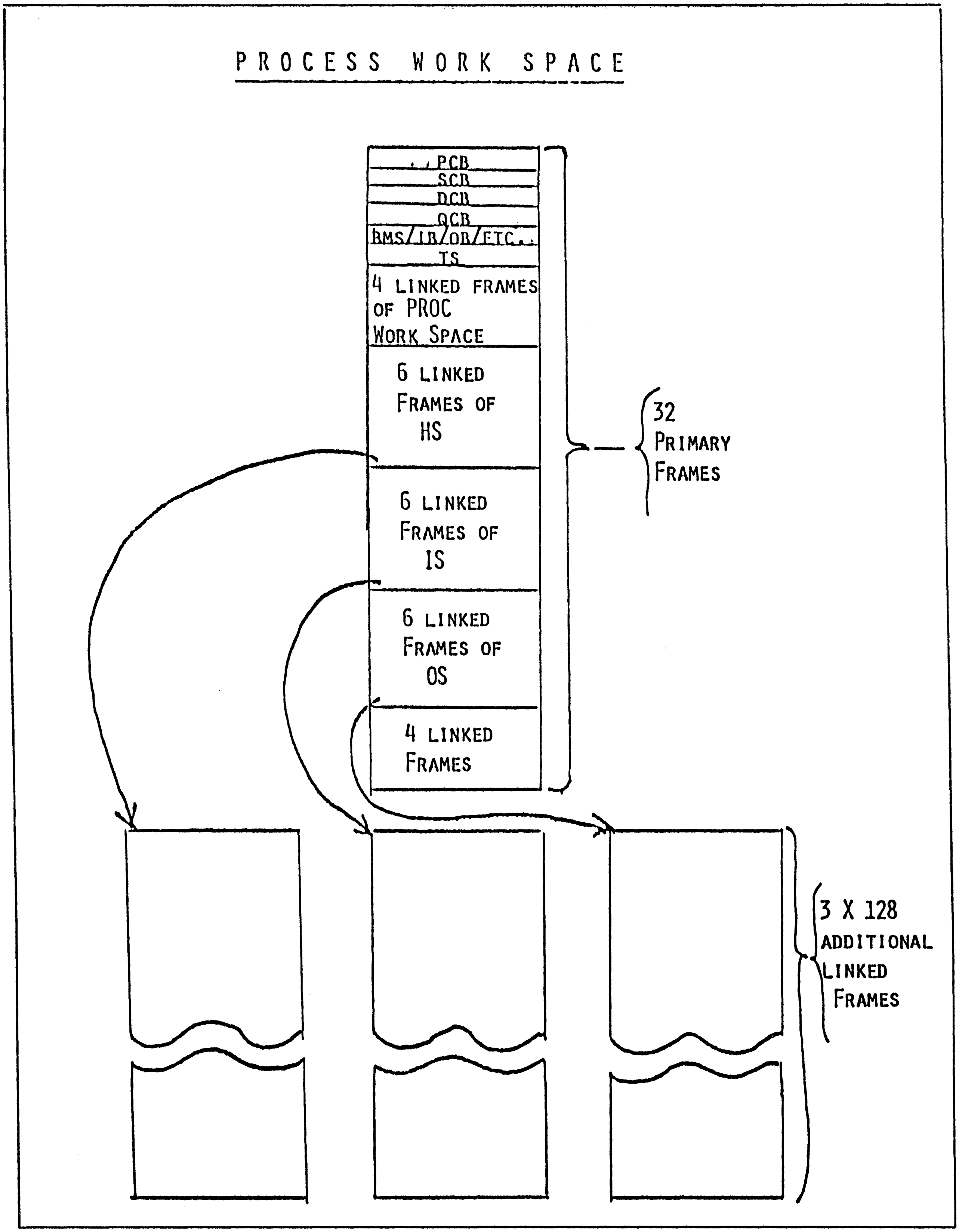

Figure B. Process Workspace Format

PCB

```
 ---------------------------------------------
|                        |  Accumulator        |
|                        ---------------------|
|           Various Storage Elements           |
|                                              |
|                   Bits                       |
|                Characters                    |
|               Half Tallies                   |
|                  Tallies                     |
|             Double Tallies                   |
|             Storage Registers                |
|                                              |
|---------------------------------------------|
|                                              |
|                      16                      |
|                   Address                    |
|                  Registers                   |
|                                              |
|                                              |
|                                              |
|                                              |
|---------------------------------------------|
|         11 Entry Return Stack                |
|---------------------------------------------|
|                                              |
|         Various Storage Elements             |
|                                              |
 ---------------------------------------------
```

NOTE: The Primary Control Block is the first frame of each Process Work Space

Figure C. PCB Format

Address Register Fields

Main Memory Address

Reserved	Disp	Flags	FID

Register Field	Valid when Attached	Valid when Detached	Where Loc-ated	No. of Bits	Range
FID	Yes	*	PCB	24	1 to 16777215
Flags	Yes	Yes	PCB	8	***
Displacement	No	Yes	PCB	16	-32768 to 32767
Main Memory Address	Yes	No	**	**	0 to **

*   FID not valid when register is Linked and Displacement is less than 0 or greater than 500.

**   Location, Size, and Bit Configuration of the Main Memory Address is dependent upon the specific hardware.

*** Flags field of X'80' indicates Unlinked format; Flags field of X'00' indicates Linked format.

Figure D. Address Register Field Format

1.4 HARDWARE COMPONENTS: CPU AND PERIPHERALS

The Ultimate system hardware includes the CPU(s) and peripheral equipment such as printers, terminals, and tape drives. The system Spooler can support up to 40 printers (4 may be parallel). Up to 250 terminals and multiple tape drives may also be configured (depending on hardware model).

Figure A shows a typical Ultimate machine implementation. The upper portion of the diagram is implemented in hardware and firmware; the lower portion is software.

An Ultimate system may be configured with two or three CPUs. In such installations, usually one CPU is used for the monitor code (Kernel) operations and one or two CPUs are used for the virtual Processes.

Terminals are attached to system ports or channels (lines) and defined to the system via system (TCL) commands such as SET-TERM, SET-BAUD, and TERM. An item in the ACC dictionary file can also be used to define terminal type and baud rate on a per-line basis, as well as the location of the terminal. See the Ultimate System Commands Guide for details on using the commands. Section 1.10, THE ACCOUNTING HISTORY (ACC) FILE, in this manual discusses the format of the ACC dictionary items, and Section 4, PROCEDURES FOR SECURITY, USER ACCOUNTS, TERMINALS, AND PERIPHERAL EQUIPMENT, discusses terminal set-ups.

The line or letter quality printer(s) are controlled by the Spooler software. Printers are attached to the Spooler and defined to the system via system (TCL) commands beginning with the letters SP (for example, SP-ASSIGN, SP-LPTR, etc.). The SYSPROG Main Menu also allows selection of commonly used Spooler procedures. Section 4.22, SPOOLER SET-UP AND MAINTENANCE, discusses printer set-ups. Also see the System Commands Guide for details on using the commands.

An Ultimate system may have multiple tape units, which are used primarily for system backup and restore procedures. Tape units are handled by the Tape commands. Section 4.23, OVERVIEW OF TAPE CONTROL COMMANDS, gives a summary of the Tape commands; the System Commands Guide details each command as a separate topic.

```
                ----------------
                | Standard CPU |
                |   & Memory   |
                ----------------
                        |
                        |                       -----------
                        |               ----->|   Tape    |
                        |               |      ----------
                        v               |
                ----------------------  |        ----------
                |   Ultimate         |<----      | Virtual|
      ------->|   Co-processor CPU |<---------->| Memory |
      |         |                    |<----      | (Disk) |
      |         ----------------------  |        ----------
      |                                 |
      |                                 |        -----------
  ------------                          |
  | Virtual  |                          ----->  | CRTs     |
  |Assembler |                          |        -----------
  ------------                          |
      |                 ------------    |        --------------
      |                 | Ultimate |    ----->  | Printer(s) |
      ----------->|    O/S   |            --------------
                        ------------
                         ^    ^   ^
                         |    |   |
         ------------    |    |   --------------------
         |                    |                     |
  ----------        --------------           ------------
  |BASIC   |        |  Utilities |           |  Recall  |
  |Runtime |        |            |           |  Compiler|
  ----------        --------------           ------------
      ^                                           ^
      |                                           |
  ----------                                  ------------
  | BASIC  |                                  |  Recall  |
  | Object |                                  |  Command |
  | Code   |                                  |          |
  ----------                                  ------------
      ^
      |
  -----------     ----------
  | BASIC    |    | BASIC  |
  | Compiler|<---| Source |
  -----------     ----------
```

Figure A. Ultimate Machine Implementation

1.5 ULTIMATE SYSTEM SYS-GEN TAPES

An Ultimate system SYS-GEN tape must be used to generate an Ultimate system. A SYS-GEN tape must contain all three basic software components of the system: the bootstrap section with the monitor code (Kernel) and Configurator, the ABS section, and the FILES section. One, two, or all three of these components may be saved to tape via a file-save, depending on the type of save specified. Only the SYS-GEN proc, however, creates a SYS-GEN tape that may be used to bootstrap (generate) and restore an Ultimate system.

A current system release level SYS-GEN tape must be available at all times to generate an Ultimate system from scratch, if necessary. A SYS-GEN tape is provided when a new system is installed and started up (bootstrap and coldstart). With each new release of the Ultimate system, a new SYS-GEN tape is provided to users.

Users may create backup SYS-GEN tapes by using a special file-save proc: the SYS-GEN command. The SYS-GEN tape may then be used to recover from system failures, restore after system maintenance, or transfer data to another system.

An Ultimate system can be generated from a SYS-GEN tape by using the Boot button on the CPU or by entering appropriate system (TCL) commands (or a combination of both). Booting, coldstarting, and creating backup SYS-GEN tapes are explained in Section 2, PROCEDURES FOR STARTUP AND SHUTDOWN OF SYSTEM.

A SYS-GEN tape contains five sections, each delimited by an end-of-file mark (EOF):

1 & 2. SYSTEM BOOT 1 and SYSTEM BOOT 2. Bootstrap section (one for DEC-based systems, one for Honeywell-based systems) with Configurator.

3. COLD LOAD. Monitor and some virtual program frames needed to boot the system, which are also loaded into Bootstrap section of virtual memory.

4. ABS LOAD. ABS section (system software). This section is preceded by a tape label that states the system release level, and is terminated by an end-of-file (EOF). The ABS section software makes up the bulk of the Ultimate operating system, the system software (BASIC, EDITOR, RECALL, PROC, ASSEMBLY), and the system utility programs.

5. DATA. FILES section containing at least the minimum set of Ultimate files. Included are the SYSTEM dictionary, a SYSPROG account, and system files ACC, SYSTEM-ERRORS, ERRMSG, etc. Several additional files are T-DUMPed after the system files; these are used for the selective upgrade method and updating the Word Processing (WP)

account. User accounts, if present in the FILES section, are preceded by a tape label containing the account name and are terminated by an end-of-file (EOF). The last account on the tape is terminated by an end-of-data (EOD) mark, which is two EOFs.

Note that normal file-save tapes contain only the FILES section, and have neither Bootstrap nor ABS sections. A Boot tape, on the other hand, contains only the Bootstrap and ABS sections, and has no FILES section.

Figure A summarizes the Ultimate system commands that are associated with creating or using a SYS-GEN tape. Each command is detailed in the Ultimate System Commands Guide. A SYS-GEN tape can also be created from the SYSPROG Main Menu and loaded from the CPU front panel button(s).

NOTE: The SYS-GEN command saves the system as it stands at the time SYS-GEN is executed. It is not guaranteed to produce a standard SYS-GEN tape as would be supplied by the Ultimate Corp. T-COPY could be used to copy a standard SYS-GEN tape, if the hardware includes two tape drives. (Only if you restore a system from a SYS-GEN tape (including files) and immediately execute the SYS-GEN command, will the resulting tape be a duplicate of the original.)

Figure B summarizes the files on a standard SYS-GEN tape.

SYSTEM COMMAND	PURPOSE
:FILELOAD	Loads FILES section from SYS-GEN tape or other file-save tape; allows loading additional data tapes, if needed. Does not reload ABS, monitor code, or other parts of the Bootstrap section.
SYS-GEN	System-save. Dumps Bootstrap, ABS, and FILES sections, creating a SYS-GEN tape.
SAVE (D,F,G,S,T,Z)	Full file-save. Dumps FILES section only.

Figure A. System Commands Using SYS-GEN Tape

The first five "files" of a SYS-GEN have the following format; each is delimited by an EOF mark:

File 1	SYSTEM BOOT 1
File 2	SYSTEM BOOT 2
File 3	COLD LOAD
File 4	ABS LOAD
File 5	FILE DATA (including the following accounts):
and up	SYSTEM
	SYSTEM-ERRORS
	BLOCK-CONVERT
	PROCLIB
	SYSLIB
	SYSTEM (Synonym)
	SYSPROG
	ERRMSG
	SPSYM
	GAMES
	ACC
	SYSTEM (Q Pointers, etc.
	T-DUMP of STAT-FILE

Several files are T-DUMPed at the end of the SYS-GEN tape. These files are used for the selective upgrade method and updating the WP account. The FILES section is subject to change with each release, as new system accounts may be added. Refer to the release documentation for the exact number. To position the tape at the first file, do an appropriate number of T-FWDs from Load-point. The files are dumped as follows:

SYSPROG MD
DICT NEWAC
ERRMSG
DICT SYSLIB
SYSLIB
DICT TERMDEF
CONVERSION,DOC
CONVERSION,PGM
CONVERSION,PROC
CONVERSION,TABLE
OSYM
PSYM
DICT STAT-FILE
DICT BLOCK-CONVERT
USER-MODES
USER-MODES,SOURCE
DICT SYSPROG-PL
DICT ACC
DICT SYS-ERRS
DICT DOCUMENT
DOCUMENT
DOCUMENT,DELTA
DOCUMENT,UPG
DOCUMENT,SYSTEM
DICT UFOS
UFOS
DICT REALLOC-FILE
WP Master Dictionary (MD)
DICT WP-PROGS
WP-PROGS,TABLES
DICT SPSYM
SPSYM
SPSYM,DF
account save of WP
ULTINET MD
ULTINET BP

Figure B. Format and Contents of SYS-GEN Tape

1.6 THE ULTIMATE FILE STRUCTURE AND SYSTEM FILES

The Ultimate system has a hierarchical file structure composed of four levels of files: the System Dictionary (SYSTEM), the user Master Dictionaries, the file level dictionaries, and the data files.

Figure A shows the four-level Ultimate file structure. In this hierarchy, the files at each level point to multiple files at the next lower level. For example, the SYSTEM file contains user account names and other system-level pointers. The user Master Dictionaries contain file names, command names, and other user-level pointers. The file dictionaries contain data definitions and other file-level definitions. Lastly, the data files themselves contain no pointers, just data items (commonly known in other systems as "records").

In Ultimate systems, the "dictionary" concept has special meaning. The data in a file must be accessed via the dictionary associated with it; the dictionary is like an index to the file. Since the dictionary is itself a file, it contains items just like a data file does. But the items in a dictionary serve to define lower level dictionaries or data files.

The system can contain any number of files, and files can contains any number of items. Files can expand to any size. Items are of variable length and can contain any number of attributes (commonly known in other systems as "fields") and characters--but the total length of an item may not exceed the maximum of 32,267 bytes (or characters).

<u>Level 0: The SYSTEM Dictionary</u>

The highest level dictionary is called the SYSTEM file. It contains all valid user Logon names and other information relevant to each user account. Each user account is a separate item (that is, file definition item) that points to the user's Master Dictionary.

<u>Level 1: User Master Dictionaries</u>

Each user account has a user Master Dictionary (abbreviated "MD" or "M/DICT"). It contains the definitions of the account's "vocabulary" (system command verbs, PROCs, etc.) as well as pointers to the files which the account may use.

When an account is created, a standard set of "vocabulary" items is stored in its Master Dictionary. Users may create synonyms to, and abbreviated forms of, these vocabulary elements by creating copies of the elements. New procedures (PROCs) may be added. New data files may be created, and pointers to existing files may be added.

<u>Level 2: File Dictionaries</u>

Each data file, or logical group of files, has a file dictionary that describes the structure of data within the associated data file(s) and items. In some cases, such as the UltiWord dictionary, WP-DOCUMENTS, multiple user or data file names may share the same dictionary. (These are known as "shared dictionary" files, and have the filename format DICT,FILE; for example, WP-DOCUMENTS,GREG).

Some dictionaries do not have an associated data file. These are known as "single-level" files, where the data is stored within the dictionary itself.

<u>Level 3: Data Files</u>

Data files contain the actual data, which is stored in attributes (fields) and items (records) of variable length. Each item has a name, which is known as the "item-id" and serves as a key to retrieving the item's data. Special delimiters are used to separate values in an attribute and attributes in the item. Values themselves can be composed of multiple sub-values instead of a single value. File items can be accessed sequentially (all items) or directly (specified items, using the "item-id" as the key). Items can also be "selected" for processing, optionally in a sorted sequence.

```
Level 0        SYSTEM DICTIONARY              One per system
            ---------------------------
            | Account names with      |
            | passwords, file access  |
            | codes, accounting data  |
            ---------------------------

Level 1     USER MASTER DICTIONARY            One per account
            ---------------------------
            | Vocabulary items such   |
            | as verbs, modifiers,    |
            | etc., and filenames     |
            ---------------------------

Level 2        FILE DICTIONARY           One per data-file(s)
            ---------------------------
            | Data definitions and    |
            | inter-relationship      |
            | definitions             |
            ---------------------------

Level 3          FILE DATA
             --------------------------
             | data items             |
             --------------------------
```

Figure A. Four-Level Ultimate File Hierarchy

The FILES section of an Ultimate-supplied SYS-GEN tape contains the essential system files. These include the SYSTEM dictionary file, some system maintenance files, and the SYSPROG and SECURITY accounts and their files.

Figure B shows typical files included on the SYS-GEN tape, and the file hierarchy levels in which they belong.

SYSTEM Dictionary File

The SYSTEM dictionary is the highest-level file in the Ultimate file hierarchy, the master cross-reference to all system and user data. It contains the file pointers to every account in the data base, as well as pointers to the system-level files such as ACC, PROCLIB, etc.

The initial entries in the Ultimate-supplied SYSTEM file define the special system-level files necessary for the Ultimate software. As user accounts and files are created, the system adds to the SYSTEM dictionary the appropriate pointers to the user's Master Dictionary (MD). These pointer items are also known as user identification items.

User identification items are either file definition (D) or file synonym (Q) items. When a user account is created, a D-pointer item is stored for that user name. The item-id is the account name that the user will enter to log on to that account. When a new user name is defined as synonymous with another existing user account name (via the EDITOR or a COPY command), a Q-pointer item is created with the new user name as its item-id.

The format of user identification items is discussed in Section 1.7, USER IDENTIFICATION ITEMS. Before attempting to change any items in the SYSTEM file, consult this topic.

A SYSTEM dictionary item with the item-id "LOGON" contains the request to log on to the system (typically, "Logon please"). An item in the ERRMSG file with the item-id "LOGON" may contain a user-defined message that is displayed after a user has logged on to an account.

ACC (Accounting History) File

The ACC files have a tri-level structure, with an ACC account, an ACC dictionary and an ACC data file section.

The ACC account is a user identification item in the SYSTEM file. The account points to the ACC dictionary file. The ACC dictionary (DICT ACC) contains two types of items: (1) those that define the data file attributes (fields), and (2) those that define the system lines (ports). These formats are detailed in Section 1.10, THE ACCOUNTING HISTORY FILE.

The ACC data section has two types of items: (1) those that indicate the actively logged-on users, and (2) the accounting history data items that keep track of the usage statistics of each user Process. These formats are detailed in Section 1.10, THE ACCOUNTING HISTORY FILE.

BLOCK-CONVERT File

The BLOCK-CONVERT file is a single-level file that contains two unrelated types of items:

1. Items that define the format used in the characters displayed when the BLOCK-PRINT verb is used.

2. Items used to print a descriptive message on the compile listing when a BASIC program is compiled with the 'A' (assembly code) option.

These formats are detailed in Section 1.11, THE BLOCK CONVERT FILE.

ERRMSG File

The ERRMSG file is a single level file that contains all system message numbers (error and informative messages) and the message text associated with them. Each account's Master Dictionary has a Q item called ERRMSG that points to this system file; the Q item is created by the CREATE-ACCOUNT command. This format is detailed in Section 1.12, THE ERRMSG FILE.

PROCLIB File (Standard System PROC Library)

The PROCLIB file is a single-level file that contains commonly used PROCs, such as CT (Copy to Terminal), LISTU (List active users), etc. Each account's Master Dictionary has a Q item called PROCLIB that points to this system file, and command name items (CT, LISTU, etc.) that call the corresponding PROCs in PROCLIB.

SYSPROG Account and Files

The SYSPROG account and files are designed to control access to certain system commands and thereby protect system integrity. This account is described in Section 1.8, THE SYSPROG ACCOUNT, MENUS, AND FILES.

SECURITY Account and Files

The SECURITY account and files are designed to protect the system as well as specific terminal lines against unauthorized access, and to ensure user account security. This account is described in Section 1.9, THE SECURITY ACCOUNT, MENU, AND FILES.

SYSTEM-ERRORS File

The SYSTEM-ERRORS file is a tri-level file reserved for logging system/program messages or errors encountered when accessing data from disk.

SYSLIB File

The SYSLIB file contains object code for system routines implemented mainly in BASIC, such as CREATE-FILE.

SPSYM File

The SPSYM file is used by the UltiPlot commands: PIE, SPIE, PLOT, SPLOT.

LEVEL	FILENAME	FILE TYPE
0	SYSTEM	dictionary
1	ACC	account
	ERRMSG	dictionary
	BLOCK-CONVERT	dictionary
	PROCLIB	dictionary
	SYSPROG	account
	SYSTEM-ERRORS	account
2	ACC	dictionary
	SYSTEM-ERRORS	dictionary
3	ACC	data
	SYSTEM-ERRORS	data

Figure B. System Level Files

1.7 OVERVIEW OF USER IDENTIFICATION ITEMS

Each system file and user account has a user identification item stored in the System Dictionary (SYSTEM). This set of user-id items defines the user account names that may be used to log on to the system. User identification items are of two types: file/account definition (D) items or file synonym definition (Q) items.

A dictionary may have any of three types of items: file/account definition (D) items, file/account synonym (Q) items, and (for file dictionaries only) attribute (A) items. The item-ids of file/account definition items are the filenames of the files to which they point. Filenames must start with a non-numeric character, may be any length, and may contain any character except a comma (,) or a semi-colon (;). The difference between D and Q items is that a D item defines the physical extent of a lower level file while a Q item points to files in other accounts.

File/Account Definition (D) Items

The system files and accounts are pre-defined by Ultimate in the SYSTEM Dictionary. As user accounts are added to a system, file/account definition items (D pointer items) are created via the CREATE-ACCOUNT system command. (For details on creating user accounts, please see Section 4, PROCEDURES FOR SETTING UP NEW USERS.) These items may subsequently be updated via the Editor.

Figure A shows the format and example of a file/account definition (D) item. Each attribute is summarized below:

Attr#/*	Name	Purpose and Values
1*	D/CODE	Identifies the D item. May be D, DV, DW, DX, DY; secondary code is used for file save procedures.
2*	F/BASE	Base frame-id (FID) of file; this is inserted by the system.
3*	F/MOD	Modulo of file; this is the number of groups or "buckets" allocated for file. Account dictionaries are normally defined with 29 groups, file dictionaries with one (1).
4*	F/SEP	Separation of file; this is the number of sequential frames per group. Both dictionaries and data files normally have one frame per group.
5	L/RET	Names of other accounts that may access (retrieve) this account/file; null means "all".

6	L/UPD	Names of other accounts that may update this account/file; null means "all".
7	V/CONV	Password for this account; null means "no password".
8	V/CORR	Privilege level for this account; may be null, SYS1, or SYS2 (See Figure C.)
9	V/TYPE	Account update codes: U = update account item in ACC file at logon/logoff times R = restart at LOGON PROC from "END" at DEBUG (!) level I = inhibit BREAK key at LOGON or L = left justify (ignored)
10	V/MAX	Not used.
11		Not used.
12		Not used.
13	F/REALLOC	Reallocation codes.
14		Lines that may logon to this account; null means "all".

Starred (*) attributes are required; the others may be null or absent.

System privileges determine the account's capabilities in regard to full system usage. The value is null for level 0, SYS1 for level 1, and SYS2 for level 2 (see Figure B).

File Synonym Definition (Q) Items

File synonym definition items (Q pointer items) may be created by the SET-FILE system command, or by the EDITOR. Figure C shows the format and example of a file synonym definition (Q) item. Each attribute is summarized below:

Attr#/*	Name	Purpose and Values
1*	D/CODE	Identifies the Q item. Must be Q.
2	F/BASE	Account name in which file is found; null for user's own account.
3	F/MOD	File name to be retrieved by specifying the Q item-id; null for Master Dictionary.
4	F/SEP	System name (for UltiNet users); null for the local system.

Starred attributes are required; the others may be null or absent.

```
item SYSPROG in SYSTEM Dictionary

001 D <-------------------------- D/CODE
002 9837 <----------------------- Base FID
003 17  <------------------------ Modulo
004 1  <------------------------- Separation
005     <------------------------ Retrieval Code (L/RET)
006     <------------------------ Update Code (L/UPD)
007     <------------------------ Password
008 SYS2 <----------------------- System Privilege Level
009 U   <------------------------ Update Account File
010 10  <------------------------ Not used
```

Figure A. Sample File Definition (D) Item For SYSPROG Account

FACILITY	LOWEST PRIVILEGE LEVEL REQUIRED
Updating of M/DICT	SYS1
Use of magnetic tape	SYS1
Use of DEBUG (other than P, OFF, END and G commands).	SYS2
Use of DUMP command	SYS2
Use of Assembler and Loader	SYS2
Use of FILE-SAVE and FILE-RESTORE commands	SYS2

Figure B. Some Required System Privilege Levels

```
item MYACCOUNT in SYSTEM Dictionary

001 Q <-------------------------- D/CODE
002 Youracct  <------------------ Account name
003         <-------------------- File name
004 NJ <------------------------- System name
```

Figure C. Sample Synonym Definition (Q) Item for MYACCOUNT

1.8 THE SYSPROG ACCOUNT, MENUS, AND FILES

The System Programmer (SYSPROG) account is included on an Ultimate SYS-GEN tape. Users must be logged onto SYSPROG in order to perform a number of system management and maintenance functions.

The SYSPROG account provides an Ultimate system with a pre-defined account designed to be used only by responsible system personnel. By selecting a unique SYSPROG password and restricting the password to authorized users, the responsible personnel can maintain a level of control over certain system procedures and thereby protect system security and integrity.

To log on to the SYSPROG account, use the following formats:

Logon please: **SYSPROG**
Password: **PASSWORD**

>**LOGTO SYSPROG**
Password: **PASSWORD**

Logon please: **SYSPROG,PASSWORD** (not recommended)

The SYSPROG Main Menu is displayed (see Figure A). This menu allows you to initiate commonly used procedures by selection, without entering the equivalent system level (TCL) command. It also allows printing of additional system documentation (see Figure B). After the selected task is completed, the system returns to TCL, not the Main Menu. (Enter "SYSPROG" to redisplay the menu.)

The SYSPROG account is supplied with a number of files in the account. To list the files in an account, use the LISTFILES command. To list the items in any file, use the LIST command (for a list of item-ids), or the CT command with the "*" (all items) option (for item-ids and contents). For details on using these commands, refer to the command name, listed alphabetically in the System Commands Guide.

The SYSPROG account files include:

<u>NEWAC File</u>

The New Account (NEWAC) file is a prototype master dictionary used in creating new accounts. It contains verb (TCL command) definition items, pointers to system files like ACC and ERRMSG, and other master dictionary items used by system software such as RECALL and UPDATE. A new account's Master Dictionary is built from the NEWAC prototype to allow access to these essential files and system commands.

<u>SYSPROG-PL File</u>

The SYSPROG-PL file contains definition items for system commands and PROCs that are meant to be executed from the

SYSPROG account. These commands are used primarily for creating accounts, system startup, and system maintenance.

STAT-FILE File

The STAT-FILE file is designed to log file statistics for the File Statistics report generated after a file-save.

CONVERSION

The CONVERSION file is used when upgrading an account created on certain old or non-Ultimate operating system releases, using the ULT command.

DOCUMENT File

The DOCUMENT file contains the SYSPROG Main Menu as well as various on-line documentation available from the menu, such as notes about the current software revision.

SYM Files (OSYM, PSYM, TSYM)

The SYM files are used by the Assembler and Assembly Debugger software. Only OSYM and PSYM are supplied on the SYS-GEN tape. TSYM is used to store the variable names associated with an account's assembly programs; this file must be defined on a user account via the CREATE-FILE command.

POINTER-FILE

In order to use RECALL commands such as GET-LIST and SAVE-LIST, a POINTER-FILE must be defined on the user account. The dictionary section of this file contains the pointers to select-lists stored by the SAVE-LIST and EDIT-LIST commands; no data section is required.

REALLOC-FILE

The REALLOC-FILE is used for file reallocations.

RECEIVED-MSGS File

The RECEIVED-MSGS file is used to receive messages via the bisync (binary synchronous) communications software (RECEIVE command).

TERMDEF File

The TERMDEF file contains definitions for terminals and printers supported by the Ultimate system. Information such as function key code sequences and cursor positioning codes is loaded from this file into system tables at coldstart time or via the SYSPROG LOAD-TERMDEF command.

UFOS

The UFOS file is used to hold temporary operating system patches from Ultimate support personnel.

USER-MODES File

The USER-MODES file contains standard assembly code user routines supplied in an Ultimate base system. Those routines in frames 400-599 may be altered by the user. All routines in USER-MODES are automatically loaded on a coldstart or file-restore. Another data section called USER-MODES,SOURCE contains reference assembly language routines, but these are never automatically loaded.

```
SYSPROG MAIN MENU
(MFGR NAME-based Systems)

1.   File Save
2.   ALL-UPDATE-SAVE
3.   PART-UPDATE-SAVE
4.   Documentation Menu
5.   Spooler Menu
6.   Automatic File Reallocation Menu
7.   Load WP Account from SYS-GEN tape
8.   Create Boot tape

88.  Logoff
99.  Go to TCL

  ENTER SELECTION -_
```

Figure A. SYSPROG Main Menu

```
                  DOCUMENT DIRECTORY
              (MFGR. NAME-Based Systems)

Features                       Procedures & Utilities

1.   Update Save &             A.  Delta Document
     Transaction Logger        B.  Upgrade Procedure Document
2.   Supplemental Manual       C.  Auto. File Reallocation
3.   Bisync Document               Document
4.   Binary Save Document      D.  WP Delta Document
5.   Diags Document            E.  User Exit Document
6.   Term Def Document         F.  Boot Tape Creation Document
7.   PC Async Document

88.  Logoff
99.  Go to TCL

ENTER SELECTION -_
```

Figure B. Document Directory Menu (SYSPROG Menu Option 4)

1.9 THE SECURITY ACCOUNT, MENU, AND FILES

The Security (SECURITY) account is included on an Ultimate SYS-GEN tape. Users must be logged onto SECURITY in order to perform a number of system security and user account functions.

The SECURITY account provides an Ultimate system with a pre-defined account designed to be used only by responsible system personnel. By selecting a unique SECURITY password and restricting the password to authorized users, the responsible personnel can maintain control over system security, user file access capabilities, and account access capabilities for each terminal.

To log on to the SECURITY account, use the following formats:

Logon please: **SECURITY**
Password: **PASSWORD**

>**LOGTO SECURITY**
Password: **PASSWORD**

Logon please: **SECURITY,PASSWORD** (not recommended)

The SECURITY Main Menu is displayed (see Figure A). This menu allows you to initiate commonly used procedures relating to system, terminal, and account security by selection, without entering the equivalent system level (TCL) command. It also allows printing of the security system documentation to the terminal screen or printer. After the selected task is completed, the system returns to TCL, not the Main Menu. (Enter "SECURITY" to redisplay the menu.)

The SECURITY account is supplied with a number of files in the account. To list the files in an account, use the LISTFILES command. To list the items in any file, use the LIST command (for list of item-ids), or the CT command with the "*" (all items) option (for item-ids and contents). For details on using these commands, refer to the command name, listed alphabetically in the System Commands Guide.

The SECURITY account files include:

<u>OLD-LOCKS File</u>

The OLD-LOCKS file contains file access lock information generated by the SECURITY REMOVE-OLD-LOCKS command.

<u>LOGONH File</u>

The LOGONH file is used with logon monitoring. It contains one item per logon failure. These items are printed via the "Display Terminal Logon Failures" option on the SECURITY menu.

NOTE: The recommended procedures for using the SECURITY account are explained in Section 4, PROCEDURES FOR SECURITY, USER ACCOUNTS AND PERIPHERAL EQUIPMENT.

```
*******************************************
*      The ULTIMATE Accounts Manager      *
* Version #.#              Date           *
*******************************************

 1.  System Security Specifications
 2.  Terminal Security Specifications
 3.  Create/Update an Account
 4.  Rename an Account
 5.  Account Save to Tape
 6.  Account Restore from Tape
 7.  Delete an Account
 8.  Display Terminal Logon Failures
 9.  Security System Documentation

 Enter Selection, 'TCL', or 'OFF' _
```

Figure A. SECURITY Main Menu

1.10 USER ACCOUNTS AND FILE HIERARCHY

All user accounts and their associated files are maintained in the FILES section of an Ultimate system. As new users and accounts are defined, their programs, PROCs, dictionaries, and data files are added to the FILES section of the system.

Each user account name has a separate user identification item in the SYSTEM dictionary that points to the account's Master Dictionary (M/DICT or MD). Only valid account names may log onto an Ultimate system. Each user account can be protected by security features built into the system, and may use standard or customized logon messages and PROCs.

Users create their own user account files, as desired. Each user account may contain program files, PROC files, dictionary files, and data files. In order to save a select list via the SAVE-LIST command, the user account must have a POINTER-FILE created for that account. For details about setting up new user accounts, see Section 4, PROCEDURES TO SET UP AND MAINTAIN USER ACCOUNTS.

During system operation, the SYSTEM dictionary file is opened by the Ultimate operating system. When a user attempts to log on, the system searches the SYSTEM file for the entered account name. If the account name is found (and the appropriate password is entered), the account-defining item points to the first frame of the user's Master Dictionary (in the FILES section of virtual memory). This Master Dictionary is then opened, the user is logged on, and the stored logon program or PROC (if any) is executed. If the account name and/or password are not found, the user must attempt the logon procedure until valid entries are made.

Once a user account is logged on, the user may perform any tasks authorized on that account, work with any file to which the account has access, and log off the account when finished. While the account is logged on, the system automatically keeps statistics such as the connect time, charge-units, pages printed, and line (port) used. After logoff or logging to another account, the Accounting History (ACC) file is updated with these statistics, if the account's Accounting Option is set.

At logoff time, a logoff message normally displays the statistics of the current session (logon to logoff), such as connect time, charge-units, pages printed.

<u>File Hierarchy</u>

Figure A shows a sample file hierarchy of two user account Master Dictionaries, each with pointers to three file dictionaries. Two of the file dictionaries have multiple data files associated with them. These are known as "shared dictionary" files. When the diictionary file is not shared, it has the same item-id as the data file, as in BARB. The

BARB dictionary contains an item also called "BARB", which is the pointer to the BARB data file. With shared dictionaries, there are as many D-items as there are data files; each D-item points to one of the data files. The DEPT dictionary contains an item called "ACCT" and an item called "MAINT", each pointing to their respective data files.

For more details, see Section 4, PROCEDURES FOR SECURITY, USER ACCOUNTS, TERMINALS AND PERIPHERAL EQUIPMENT. Refer also to the Recall manual for background on user file retrieval and update capabilities.

```
                     --------------
                     |  SYSTEM    |
                     | Dictionary |
                     --------------
                         /     \
                        /       \
-------------   -------------  -------------   -------------
|   File    |___| Master    |  | Master    |___|   File    |
| Dictionary|   | Dictionary|  | Dictionary|   | Dictionary|
|  BARB     |   | for BAM   |  |  DCW      |   |  DONNA    |
-------------   -------------  -------------   -------------
     |             |    |          |      |          |
-------------      |    |          |      |    -------------
| Data File |      |    |          |      |    | Data File |
|   BARB    |      |    |          |      |    |  DONNA    |
-------------      |    |          |      |    -------------
          ---------     |          |    -----
          |             |          |        |
    --------------      |          |   ---------------
    |    File    |      |          |   |    File     |
    | Dictionary |      |          |   | Dictionary  |
    |   LETTERS  |      |          |   |  MEMOS      |
    --------------      |          |   ---------------
          |             |          |        |
    --------------      |          |   ---------------
    | Data File  |      |          |   |  Data File  |
    |  LETTERS   |      |          |   |  MEMOS      |
    --------------      |          |   ---------------
                        |          |
               -------------  -------------
               |   File    |  |   File    |
               | Dictionary|  | Dictionary|
               |  DEPT     |  |  JOBS     |
               -------------  -------------
                     |              |
    -----------------          -----------------
    |              |           |               |
----------- ------------   -----------    ------------
|Data File| | Data File|   | Data File|   | Data File|
|  ACCT   | |  MAINT   |   |  EDIT    |   | GRAPHICS |
----------- ------------   -----------    ------------
```

Figure A. File Hierarchy

1.11 THE ACCOUNTING HISTORY (ACC) FILE AND USER ACCOUNTS

> The basis for Ultimate system accounting history and statistics is the individual user account. The Accounting History (ACC) file is a mandatory system file. This file contains both the history of system usage and the current system status of logged on user accounts. It may also contain items that define the terminal types and baud rate currently associated with each system line (port).

The ACC file is a tri-level system file, with an ACC account level, a dictionary level, and a data level.

The ACC Account Level

The System dictionary (SYSTEM) contains the file definition item (item-id 'ACC') for the Accounting History File. This item is included with the Ultimate base system and has the standard D pointer item format (see Section 1.7, OVERVIEW OF USER IDENTIFICATION ITEMS).

The ACC Dictionary Level

The DICT ACC file contains attribute definition (A) items and may contain line definition items. The A items define the attributes associated with the ACC data file items so that they may be examined and listed via system commands. The item-id of each A item is the attribute name. They have the format shown in Figure A.

Users may create additional items in the DICT ACC file to define the terminal type and baud rate for each line (port) in the system. These items also specify the location of the terminal, which is displayed by the LISTU command. The item-id of each line definition item is the 3-digit line number. These items have the format shown in Figure B. Attribute 1 is the string describing the terminal location. Attribute 2 may be null, or contain 1-7 parameters (r,c,s,p,e,x,t) defining the baud rate (r=baud rate; c=char length; s=stop bits; p=parity; e=echo; x=xon/off; t=typeahead); see the SET-BAUD command in the System Commands Guide for the permissible values. Attribute 3 may be null, or contain the single-character TERM type, or may contain all parameters accepted by the TERM command; see the TERM or SET-TERM command in the System Commands Guide for the permissible values. For set-up procedures, see Section 4, PROCEDURES FOR SETTING UP NEW USERS AND TERMINALS.

The ACC Data Level

The ACC data file has two types of items: (1) active user statistics for logged-on user Processes, and (2) user Account Name/Line# accounting statistics. These items have the format shown in Figure C. They are maintained by the system via the logon and logoff procedures.

ACC Active User Items

Active user items are created when a user logs on to a Process line, and are deleted at logoff. The item-id of an active user item in the Accounting History File is the four-character hexadecimal FID of the PCB of the user's Process. If the PCB's start at FID=1024, (they proceed in steps of 32 frames from there on), we see that a user logged on to process zero will have an entry with an item-id '0400' (1024), while a user logged on to process one will have an entry with an item-id '0420' (1056), and so on. Attribute one of an active user item contains the name of the user (i.e., the item-id of the user identification item), attribute two contains the date logged on, and attribute three shows the time logged on.

ACC Accounting History Items

Accounting history items are created the first time a user logs onto a particular line, and are updated (if required; see NOTE below) after each logon session. The item-id of an accounting history item is the name of the user account (i.e., the item-id of the user identification item), with the line number concatenated by a "#". For example, if user 'SMITH' logs on to line 12, when he logs off, the item whose item-id is 'SMITH#12' in the ACC file will be updated. This allows one to keep track of system usage by user-id as well as line number. If multiple logoffs occur in the same day, the times, charge-units, and printer pages are stored for each logoff as subvalues.

Attributes 1, 2, and 3 are not used. Attribute 4 contains the date(s) logged on; each unique date is a value stored in Ultimate date format. Attribute 5 contains the logon time(s) represented in seconds past midnight (24-hour clock). Attribute 6 contains the connect time(s) as the number of seconds between logon and logoff. Attribute 7 is the charge-units, representing CPU usage. Attribute 8 is line printer pages (routed to printer during session).

Attributes 5, 6, 7, and 8 are control-dependent values which are controlled by attribute 4. (The RECALL manual discusses the control-dependent data set format.)

NOTE: The accounting history items are not automatically created or updated. The SYSTEM dictionary item for the user must have a 'U' in attribute 9 if the user is to have these items created and updated. This is indicated by requesting the Accounting Option in the CREATE-ACCOUNT or UPDATE-ACCOUNT menu.

If the SYSTEM dictionary data has been changed since logon or the history item to be updated is too large for the work-space, the message number 338 will be printed.

To avoid overflow, the account history items should be periodically cleared. For this procedure, see Section 4, PROCEDURES FOR NEW USERS AND SYSTEM MAINTENANCE.

Figure D shows a sample sorted listing of the active users via a LISTU command. Figure E shows a sample listing of the accounting history item for user SMITH via a LIST command.

```
     DATE                          Attribute Name

001  A  <-----------------------Attribute Definition Code
002  2  <-----------------------Attribute number
003     <-----------------------Heading or Tag
004     <-----------------------Structure Code
005     <-----------------------Reserved
006     <-----------------------Reserved
007     <-----------------------Conversion specifications
008     <-----------------------Correlative specifications
009  L  <-----------------------Justification code
010  12 <-----------------------Reserved

One item per defined attribute, as follows:

Attr#0    LINE or LOCATION.  Used in LISTU Command
Attr#1    NAME         Used in Active User Item
Attr#2    DATE.         "    "    "       "    "
Attr#3    TIME.         "    "    "       "    "
Attr#4    DATE         Used in Accounting History Item
Attr#5    TIME          "    "     "           "       "
Attr#6    CONN          "    "     "           "       "
Attr#7    UNITS         "    "     "           "       "
Attr#8    PAGES         "    "     "           "       "
```

Figure A. Format and Example of Attribute Definition Items for DICT ACC

```
     001     <--------------------Line number

001  LINE 1 AT BARBARA'S DESK <---Location of terminal
002  1200 <-----------------------Baud rate parameters
003  V <--------------------------Terminal type or parameters
```

Figure B. Format and Example of Line Definition Items for DICT ACC

```
Active User Items

     342A                FID of User Process PCB (hexadecimal)

001  DONNA  <----------------------User name
002  6449     <--------------------Date logged on
003  10394    <--------------------Time logged on


Accounting History Items

     SYSPROG#1           Account*Line

001       <-------------------------Not used
002       <-------------------------Not used
003       <-------------------------Not used
004  6449  <------------------------Dates logged on
005  10384\32178  <-----------------Times logged on
006  21790\17 <---------------------Connect times
007  241\4 <------------------------Charge-units
008  \  <---------------------------Number of printer pages
```

Figure C. Format of ACC Data Items

```
>LISTU <CR>

 CH# PCBF NAME...... TIME... DATE.... LOCATION..............

 000 0200 CM         11:02   03/22/86 CM's DESK
 001 0220 SYSPROG    12:10   03/22/86 COMPUTER ROOM
 002 0240 ELROD      09:11   03/22/86 LINE 2 IN WAREHOUSE
 003 0260 LC         06:59   03/22/86 LC's DESK
*006 02C0 CM         11:25   03/22/86 LINE 6
 007 02E0 BUGEYE     01:29   03/21/86 BUGEYE'S BENCH
```

Figure D. Sample Sorted List of Active User Items (via LISTU)

```
>LIST ACC = "SMITH]"        (selects items with item-ids
                            starting with the string "SMITH")

PAGE    1                             12:17:22  22 MAR 1986

ACC...... DATE. TIME... CONN... UNITS.. PAGES
                *       *       *       *
SMITH#0   01/13   16:56   00:04       9
          01/14   10:13   00:00       5
                  10:15   00:01     343
          02/06   17:02   00:18      41
          02/09   10:21   00:17     690
          02/23   07:58   00:01      27
          03/09   11:35   01:57     378
                  16:05   00:22      94
SMITH#5   01/13   12:48   02:25     160      5
                  15:20   00:05      14
                  15:25   00:00       2
                  15:28   00:17     110
                  16:20   02:55    2575     16
                  19:15   00:00      13
          01/16   09:41   06:13    1853      6
                  15:55   00:12      15

2 ITEMS LISTED.
```

Figure E. Sample List of Account History for User "SMITH"

1.12 THE BLOCK-CONVERT FILE

The BLOCK-CONVERT dictionary file defines all characters that can be printed with a BLOCK-PRINT command. It also contains items that correspond to the BASIC opcodes generated by the BASIC compiler, which are used when BASIC generates a listing of a program's object code.

The BLOCK-CONVERT file supplied in an Ultimate base system contains an item for each printable ASCII character (see Appendix B for a list of ASCII Codes). It also contains an item for each BASIC opcode (that is, all used codes in the range 00-FF).

Figure A shows an example of the two types of items from the BLOCK-CONVERT file. The first example is the item that forms the letter "S" in block-printing. The item-id is the letter itself. All block-printed letters have a height of 7 characters (there are 7 vertical rows of characters). The width is variable. The format for the attributes is:

```
1    n{c}
2-9  B/Cn{,n,n}
```

In attribute 1, 'n' is the width of the character matrix (the number of horizontal characters). The optional 'c' is a character to print instead of the specified letter in building the block print output; that is, a block print "S" is normally made up of S's, a "G" of G's, etc.

Attributes 2-9 specify the character layout of each of the 8 vertical rows of characters, top to bottom. The layout assigns either a blank (B) or a character (C) to each cell in the row, left to right. The 'B' or 'C' specifies filling the leftmost cell with a blank or a character. The 'n' specifies how many blanks or characters to print contiguously. A comma (,) switches from B to C or vice versa, and the 'n' specifies how many to print. The sum of 'n's must equal the character width defined in attribute 1. For example, to print a string of 7 S's and no blanks, the attribute would contain "C7" and to print 1 S, 5 blanks, and 1 S, the attribute would contain "C1,5,1".

The second type of data in the BLOCK-CONVERT file has a two-hexadecimal-digit item-id, corresponding to the BASIC opcode generated by the BASIC compiler; attribute 1 is the symbolic name for the opcode. These entries are used by the "A" option of the BASIC compiler to generate a listing of the BASIC object code.

Figure B is an example of BLOCK-PRINT command output.

Figure C contains a sample listing from a BASIC compiler command with the "A" option (Assembled code listing).

```
>COPY BLOCK-CONVERT S 8A (T) <CR>

Attribute         Meaning

    S             Item-id; defines format for character "S"
001 7             Defines character width as 7
002 B1,5,1        Specifies string " SSSSS " (1 blank, 5 S's,
                  and 1 blank)
003 C2,3,2        Specifies string "SS   SS"
004 C2,5
005 B1,5,1        Specifies string " SSSSS "
006 B5,2
007 C2,3,2
008 B1,5,1
009 B7            Bottom row is all blanks


    8A            Item-id (BASIC object-code byte)
001 STOP          Identifies object-code (STOP opcode).
```

Figure A. Sample Items from BLOCK-CONVERT File

```
>BLOCK-PRINT S <CR>

                           SSSSS
                          SS   SS
                          SS
                           SSSSS
                               SS
                          SS   SS
                           SSSSS
```

Figure B. Output Using BLOCK-PRINT Command

SOURCE CODE LINE	BASIC OBJECT CODE	PSEUDO ASSEMBLY CODE
001	01	EOL
002	03	LOADA I
003	FD	LOAD. 1
004	20	ONE
005	2D	SUBTRACT
006	5F	STORE
...		

Figure C. BASIC Compiler - Assembled Code Listing

1.13 THE ERRMSG FILE

The ERRMSG dictionary file supplied with the Ultimate system contains the standard system error messages generated by the system software, such as TCL commands, RECALL, BASIC, PROC, etc. System users may change these messages, add new messages, and create custom ERRMSG files for user accounts.

LOGON Item and Message

When a user logs onto an Ultimate system at a terminal, the system displays the message specified by an item called "LOGON" in the ERRMSG file supplied with the base system. However, this message may be replaced with any other message which should be received by all users on the system immediately upon logging on.

NOTE: This item must exist on file even if no general system message is displayed (that is, the item is null).

This item should be changed only when there are no users logged on to the system.

In addition, a user account may define an ERRMSG file specifically for that account. The account's ERRMSG file may also contain an item called "LOGON", which may contain any message which should be received by all users logging onto that account. The account's logon message will be displayed after the system's logon message.

The LOGON item in the ERRMSG file should not be confused with the LOGON item in SYSTEM, which contains the 'Logon please' message in ERRMSG format.

ERRMSG Item Format

The item-ids of the standard items in ERRMSG consist of the message number or name that identifies the error message contained in the item. Each line in an ERRMSG item must conform to a general format:

```
C{text}
```

where 'C' is a code that specifies an operation associated with the message. The valid options for 'C' are:

Code	Meaning
A	Inserts into the error message the next parameter in the list of parameters passed (along with the error number) to the error message program. The parameters may be specified by a BASIC program via a PUT, STOP or ABORT statement, or by other system software reporting a system-generated error message.
A(n)	Inserts the next parameter as above, but left justified in a field of 'n' blanks.
D	Places the current date in the output buffer.
H	Places the text following the 'H' in the output buffer, without a <CR> or line feed.
H+	Used at the end of the ERRMSG item only; same as 'H' and also suppresses the final <CR> and line feed that is normally output.
L	Prints the output buffer, with <CR> and line feed.
L(n)	As above, followed by n-1 blank lines.
R(n)	Inserts the next parameter as A (above), but right justified in a field of 'n' blanks.
S(n)	Sets the output buffer pointer to location 'n'.
T	Places the current time in the output buffer.
X	Skips a parameter in the parameter list passed to the error message program.

Connect Time Messages

The ERRMSG items "335" and "336" contain the connect time messages that are displayed when a user logs on or off the system (in the standard LOGON item). The connection times are retrieved from the Accounting History (ACC) data file.

To display an error message in the ERRMSG file, use the PRINT-ERR system command. (See the PRINT-ERR command, listed alphabetically in the System Commands Guide.)

Figure A shows some examples of error message processing.

Use of ERRMSG items in a BASIC Program

In a BASIC program, these lines:

```
FILE = "BP" ; ID = "1006"
OPEN FILE ELSE STOP 201,FILE
READ ITEM FROM ID ELSE STOP 202,ID
```

specify two possible error conditions and their associated message numbers in ERRMSG. An error condition could cause the program to stop with either of the following:

```
[201] 'BP' is not a file name.
'1006' not on file.
```

Use of the LOGON item in ERRMSG file defined in user account

If the item "LOGON" in the ERRMSG file for an account looked like:

```
HHello out there!
L
HIt's now
T
H and all's well!
```

then logging on to that account would display this message:

```
Hello out there!
It's now 11:22:33 and all's well!
```

Figure A. Sample Usage of the ERRMSG File

SECTION 2

PROCEDURES FOR STARTUP AND SHUTDOWN OF SYSTEM

2.1 BOOTSTRAPPING THE SYSTEM

Bootstrapping an Ultimate system usually requires using a SYS-GEN or Boot tape/disk to load the Bootstrap program into the computer. If the system was halted via a warmstop, it can be booted from stored memory on the disk. After booting, the system can then be started up (coldstart) or restarted at the current processing point (warmstart). Each manufacturer-based system initiates the bootstrap process from its front panel in a distinct way.

Bootstrapping must be done to bring up a new system, to update a system to a new release level, to restart after a warmstop for scheduled maintenance, to restart after system failure (crash), etc.

In order to bootstrap an Ultimate system, you must have a SYS-GEN and/or Boot tape/disk at hand to boot from. All boot dialogue is via the terminal on line zero. Bringing up the system requires three general steps:

1. Power on the CPU.
2. Press the appropriate buttons on the CPU.
3. Mount the SYS-GEN (or Boot) tape; wait until the tape is loaded and ready and press <CR>. Or, boot from disk, if a warmstop has been used and you wish to warmstart, or reboot, or enter the Diagnostics or Utilities Monitors.
4. Wait (the system may display the message "Wait") while the system boots itself until the System Startup Options are displayed. Then select the appropriate option to bring up the system.

<u>The System Generation (SYS-GEN) Tape/Boot Tape</u>

A SYS-GEN tape is supplied by the ULTIMATE Corp. or dealer for new installations, new release levels of the operating system, and when requested by a user. Ultimate-supplied SYS-GEN tapes contain everything needed to start up a system, but nothing the user has stored on the system (for example, user data files).

Old SYS-GEN tapes may deteriorate in time, so it is a good practice to create extra copies of the system for immediate restarting capability.

A Boot tape contains only the system programs necesary to boot the system (Bootstrap section) and coldstart (COLDSTART and ABS sections) or warmstart the system. It does not include a FILES section, which contains system accounts (for example, SYSPROG) and system files (for example, ERRMSG). (For more about the format of SYS-GEN tapes, see Section 1.5, ULTIMATE SYSTEM SYS-GEN TAPES.)

Users may create a Boot tape from the SYSPROG Main Menu (the Create Boot Tape option) or from the CREATE-BOOT system command. A SYS-GEN tape with all user data files can be created by the SYS-GEN system command. Procedures for creating and duplicating SYS-GEN tapes can be found in Section 2.7, CREATING A SYS-GEN OR BOOT TAPE, and Section 2.8, DUPLICATING SYS-GEN TAPES. (Also refer to the System Commands Guide for details on CREATE-BOOT and SYS-GEN commands.)

Booting a DEC-Based System

Most DEC-based systems have front panel switches that resemble the ones shown in Figure A.

The Boot button will bootstrap the system from tape. The system displays the message:

PUT TAPE ON-LINE, PRESS <CR>

Bring the SYS-GEN tape to load point and press <CR>. Or, to boot from disk after a warmstop, enter <CTRL/D>. The booted system then displays the System Startup options.

Booting a Honeywell-based System

The panel buttons for the larger (Models 6800 and up) systems resemble the one shown in Figure B. The smaller (Models 6000 to 6600) systems have only an Initialization lock similar to the one shown in Figure C. However, the Honeywell terminal has function keys that correspond to the panel buttons (also shown in Figure C).

In addition, Ultimate has a System Console Facility (SCF), used on line 0 on Honeywell systems without full control panels (that is, 6000, 6200, 6400, 7000, 7200, 7400). There are several operational modes; the Maintenance Mode can be used to boot as if from a full control panel. To use Maintenance Mode on line 0, do the following:

1. Enter <ESC> <ESC> # (where <ESC> = ESCAPE key) to get into the Command Mode. The screen will display the first mode: #_ENABLE PANEL.

2. Enter # to step through the mode selections until Maintenance Mode appears: #_ENABLE MAINTENANCE.

3. Press <CR> to select Maintenance Mode. Then the SCF boot commands (listed below) may be used.

4. To exit Maintenance Mode, enter # to return to Command Mode; step through the modes using # until Console Mode (#_ENABLE CONSOLE) is displayed. Press <CR> to return to normal terminal use.

To boot from the panel buttons or SCF, press in sequence:

Full Control Panel		SCF	
S	Step	H	Halt
CLR	Clear	^	Clear (^ = up arrow)
L	Load	L	Load
R	Ready	G	Go
E	Execute	X	Execute

Then wait for the traffic light to go out and again press:

E	Execute	X	Execute

The system then displays the System Startup options.

The small systems can be booted from disk or tape, initiated from the terminal or panel. Turning the key to UNLOCK enables the terminal function keys to be used as "panel buttons" in the sequence given above. Turning the key to ON (or UNLOCK) boots the system from disk (used only for warmstarts or diagnostics). For 6000 systems, to boot from tape, first turn the key to INITIALIZE, then ON, then press the EXECUTE function key to bring up the System Startup options. The 6200 and 6400 systems must go to Maintenance Mode and use the SCF 'X' (Execute) command.

Honeywell-based system users can boot the system from a user-specified disk channel address via the (B)oot option. This option is designed to allow users to boot a system from a channel address different from the default boot address (0400, hexadecimal).

To use this option, initialize the system with the SYS-GEN tape/pack mounted. Then, at the System Startup Options, type in "B". The system will ask for DISC CHANNEL to boot from. Mount the desired disk pack (for example, Honeywell TACPAC) on the appropriate disk drive and enter the corresponding channel address.

Booting a DEC VAX-Based System

On a DEC VAX system, the Ultimate operating system is transported to the VAX through a combination of hardware and software. The capability to run Ultimate is handled through VMS, the VAX's native operating system, and therefore the bootstrapping is done as a part of booting VMS.

In addition, coldstart procedures must be run to prepare the Ultimate system for users to logon. Please refer to Section 2.4, BRINGING UP THE ULTIMATE OPERATING SYSTEM ON A VAX, for VAX startup procedures.

The System Startup Options

The initial bootstrap process loads the software that displays the System Startup Options:

```
(B)oot
(C)oldstart               (omitted if boot is from disk)
(D)iagnostics Monitor
(F)ile Restore            (omitted if boot is from disk)
(O)ff-line Monitor        (Honeywell-based systems only)
(U)tilities Monitor
(W)armstart

Enter Option(s) or ? for help:
```

To select an option, enter the appropriate letter code and press <CR>. To display a Help screen that summarizes the options, type a question mark (?) and press <CR>.

The System Startup options do the following tasks:

B Boot system - allows booting again from a different tape or disk, if another one has been loaded. It loads the BOOT SYSTEM 1 and BOOT SYSTEM 2 (first two files of SYS-GEN tape) into the CPU. Then it displays the Startup options again.

C COLDSTART - loads the COLDSTART and ABS (third and fourth files of SYS-GEN tape) into the CPU and coldstarts system. See Section 2.2, COLDSTARTING THE SYSTEM, for details.

D DIAGNOSTICS MONITOR - is used by customer and field engineers for formatting disc packs, checking system configuration, booting other channels, creating T&V's (Honeywell Test and Verification procedures), and other diagnostic utilities. If this option is inadvertently entered, the user may recover by rewinding the tape and re-booting. (See Section 2.10, THE DIAGNOSTICS MONITOR for details.)

F FILE-RESTORE - performs a coldstart as in 'C' option above, and then loads the system and user files saved on a DATA tape (or the FILES section of the SYS-GEN tape); multiple tapes may be used in the restore. This is the option to restore an entire system (same as coldstart plus the :FILELOAD system command at TCL level.)

See Section 3.7, FILE-RESTORES, for details.

O OFF-LINE MONITOR (On Honeywell-based systems only) - used to create an off-line environment where only line 0 is activated. Functions such as disk to disk copy (that is, the DISK-COPY command) can be performed without endangering an on-line system.

U UTILITIES MONITOR. Used for off-line functions such as binary saving and restoring the system from tape (or disk) to tape and vice versa. (See Section 2.11, THE UTILITIES MONITOR for details.)

W WARMSTART - used to reload the Monitor (Kernel) without virtual system inititalization. Restarts each user Process without loss of work that was in progress at the time of a system warmstop. (Same as the :WARMSTART command at TCL level.)

If a separate software Monitor option is selected, that software is given control of the system. (See Section 2.10, DIAGNOSTICS MONITOR, or Section 2.11, UTILITIES MONITOR.)

If a COLDSTART, WARMSTART, or FILE RESTORE is selected, the configuration software establishes the following parameters:

1. Size of the disc drive
2. The amount of MOS Memory (Main Memory)
3. The number of communication (terminal) lines

These parameters are reported on the terminal (a sample report is shown in Figure D).

Note: The number of additional (secondary) workspace frames is fixed at 128 frames per process line.

The Configurator passes control to a virtual Process on line 0 which loads the system software and all of the files.

The Configurator loads the proper disc tables, calculates the maximum FID (MAXFID), and the SYSTEM dictionary base (SYSBASE), buffer tables, etc. After the configurator has finished running, the monitor code (Kernel) is started. This concludes the bootstrap portion of the system startup. The coldstart, warmstart, or file restore then begins. (See Section 2.2, COLDSTARTING, or Section 2.3, WARMSTARTING, or Section 3.7, FILE RESTORES.)

```
HALT     BOOT     LTC      POWER ON
____     ____     ____      ____
|__|     |__|     |__|      |__|

RUN      DC ON     ON        OFF
```

Figure A. Sample Front Panel on DEC-Based Systems

```
      PWR
      ___
      |_|

                |  L | C |                          | S | R |
                ---------------------------------------------
   CONTROL      |CLR|                        EXECUTE | E |
                -----                                -----
```

Figure B. Sample Front Panel on Large Honeywell-Based Systems

```
            ON
            ___
  UNLOCK    |_|   INITIALIZE

From UNLOCK position, the terminal function keys that
correspond to the panel buttons are:
                                                  CRT Type
 ____________________________________________________
|   ^    |  L  |            |  H  |         | G   |  X  |   non-Hon.
 ____________________________________________________
| CLEAR  |  F1  |           |  F4  |        | F6  | F7  |   Honeywell
-----------------------------------------------------
  Clear   Load              Step            Ready  Execute
```

Figure C. Sample Front Panel on Small Honeywell-Based Systems

```
          Disc Configuration

chan      set name        #  of
FF20      ULTIMATE LSI    1  2

          Disc Configuration

chan      set name        #  of
FF20      ULTIMATE LSI    2  2

Memory = 512KB
16 Comm. Lines

The date and time must be reset!!!
```

Figure D. Sample Bootstrap and Warmstart Completion Message

2.2 COLDSTARTING THE SYSTEM

If a system fails, or if any maintenance has been performed on the system it will be necessary to perform a coldstart. Coldstarting loads the COLDSTART and ABS sections from tape but does not load system or user files (FILES section). You may coldstart after booting the system, or use the COLDSTART system command (from the SYSPROG account), or logon to a special COLDSTART account.

The coldstart process is used to load the memory-locked monitor code (Kernel) and all assembly level virtual system software code from the SYS-GEN tape. This ensures that the system software is properly loaded.

Performing a coldstart does not result in the loss of any Spooler hold files. Also, any print jobs which have not been spooled will be retained as 'spooler hold file' entries. If you are coldstarting because the system hung while doing normal processing, there is a possibility of group format errors (GFEs) being generated. These must be corrected prior to doing any more work with the affected file(s). Consult Section 5.9, GROUP FORMAT ERRORS.

Coldstarting after Booting the System

To perform a coldstart, you must load the system from tape or disk. (It is not on the option list if you boot from stored memory, as in after a warmstop.)

1. Mount your current level Ultimate SYS-GEN tape, and bootstrap the system as described in Section 2.1, BOOTSTRAPPING THE SYSTEM.

2. When the Startup options are displayed on line 0, type a 'C' with no carriage return. The system will output the Disc Configuration message, the label of the tape being used, the memory size, the number of communication lines on the system, and the system serial number stored on disk (see Figure A).

At this point, the boot has already loaded the COLDSTART section (including the Kernel and some virtual code) and control has been transferred to the Coldstart program on line zero. The Coldstart software is ready to load the ABS section from tape onto the disk; this step is called the "ABS-LOAD". Line 0 is prompted for an ABS tape:

```
Mount ABS tape and enter number of files to skip, if any
```

To load the ABS section from the SYS-GEN tape, just press <CR> to start the ABS-LOAD. If a different SYS-GEN tape needs to be loaded, three files must be skipped (Boot 1/2 and COLDSTART); enter "3" and press <CR>. Or, if a different ABS tape should be loaded, mount that tape before pressing <CR>. (A tape with only an ABS file, and optionally a COLDSTART file, can be created via the ABS-DUMP system command; see Section 3.2, STANDARDS FOR SYSTEM BACKUP, for details, and the System Commands Guide.)

Once the operator presses <CR>, the ABS-LOAD will begin with the message:

'An'

indicating that the ABS-LOAD is starting with frame n. As the ABS-LOAD continues, each contiguous group of frames is displayed as it is loaded. The Spooler is started on the last line of the system. The Process workspace for line 0 is linked. The logon and Ultimate welcome message are displayed.

At the end of the ABS-LOAD step in the coldstart procedure, line 0 has automatically had its secondary workspace linked, and will be logged onto an account named "COLDSTART".

Figure B shows a sample of all ABS-LOAD messages and the COLDSTART initial message.

Coldstarting from the COLDSTART Account or Command

The COLDSTART account is simply a Q-pointer to the SYSPROG account. Ultimate supplies a password, which may be changed (from the SECURITY account) without affecting the other methods of coldstarting the system.

The Coldstart Procedure

The COLDSTART logon proc is a Q-pointer to the COLDSTART proc (system command). This proc is set up to ask the user for the current time and date. Any other operations which the user wishes to occur at coldstart time may be included in the USER-COLD-START proc in DICT SYSPROG-PL; this proc is called as a subroutine of the COLDSTART proc.

NOTE: The COLDSTART command at the TCL level executes the COLDSTART proc only, without an ABS-LOAD. (See the COLDSTART command in the System Commands Guide).

Figure C shows the sequence of messages associated with the coldstart procedure. The secondary TCL workspaces are linked. The operator is prompted to enter the current time and date; a sample operator entry is shown in boldface type. (Pressing <CR> only will skip resetting these values.) The user modes and other system software are then loaded, reported, and verified. The procedure ends with the logoff message (from the COLDSTART account) and the logon message.

All lines are locked out during this procedure until it concludes with the Logon please message. The COLDSTART proc releases these lines by executing the :ACTIVATE-LINES verb before logging off.

Users may then log on to any user account and resume work.

Error Reporting During ABS-LOAD

During the ABS-LOAD, checksum or format errors may occur. One of the following messages will then appear:

```
Checksum error on frame -- continue (Y/N)?

Format error - last frame -- continue (Y/N)?
```

This normally means that the tape has deteriorated or was misread in such a way that the format was unrecognizable. This may especially happen after entering 'Y' in response to the prompt following a parity error. Entering 'Y' in response to this prompt will initiate a search for the next ABS frame segment on tape. If such a segment is found, the ABS-LOAD will continue with the frame indicated. If no segment is found, the tape will continue moving until it is stopped or an end of file (EOF) is reached.

```
             Disc Configuration

chan       set name        #  of
FF20       ULTIMATE LSI    1  2

           Disc Configuration

chan       set name        #  of
FF20       ULTIMATE LSI    2  2

L 2000#04:54:32 29 JUL 1985 ABS ULTIMATE RELEASE 10*140J COLD LOAD FRAMI

Memory = 512KB
16 Comm. Lines
Tape attached block size: 4000


System serial # is presently TN6969

Enter system serial # or <CR> to accept
```

Figure A. Coldstart Initialization Message

```
            A1
            A600-646
            A648-849
            A920-993
            A999-1023
            Saving Monitor

            Spooler started
            Linking workspace for line 0

05 FEB 1986 08:22:14 Logon please:

<<<< Welcome to the Ultimate Computer System >>>>
<<<< Copyright July, 1981  The Ultimate Corp. >>>
<<<< 08:22:19 Release 10 Rev 140J 26 AUG 1985 >>>

This is the Cold-Start Procedure
     Enter <CR> to continue
```

Figure B. Sample ABS-LOAD Messages

```
Linking secondary TCL workspaces

[294] 1 additional task workspaces initialized

08:43:26 26 AUG 1985

Time:12:30
12:30:02 26 AUG 1985
Date:06 FEB 1986
12:30:32 06 FEB 1986

[216] Mode 'CURSUR'    loaded;  Frame = 422  Size = 187  Cksum = 5966
[216] Mode 'ENT 440'   loaded;  Frame = 442  Size = 005  Cksum = ED
[216] Mode 'IIKERNEL2' loaded;  Frame = 444  Size = 191  Cksum = 9FCD
<<<--Load terminal definitions-->>>
<<<------Apply OS Patches ------>>>
UFOS FILE EMPTY
<<<-----Verifying software------>>>
[341] Ultimate system software verified.
SP-STARTLPTR 0,0,1,S15
[1118]
 The printer control block has been initialized.
 The correct paper and LPI settings must have been
 previously set to insure proper printing.

< Connect time= 23659 mins.; CPU=465 units; lptr pages= 0 >
< Logged off at 10:01:55  on 06 FEB 1986                  >

06 FEB 1986 10:01:55 Logon please:
```

Figure C. Sample COLDSTART Procedure Messages

2.3 WARMSTARTING THE SYSTEM

Warmstarting restarts an Ultimate system without reinitializing the virtual system. Each user Process restarts at the processing point at which it was warmstopped. Warmstarts can occur at any time when the system is operational, but should only be done after bootstrapping if the system was brought down via a warmstop.

NOTE: This procedure should not be used after a Disk-copy. The system should be coldstarted.

A system warmstart can be initiated two ways:

1. When the system is bootstrapped, select the (W)armstart option.

2. At the system TCL level, enter a :WARMSTART command.

The WARMSTART process assumes that memory has been flushed (thereby preserving the state of all active Processes on the disk). It resets the hardware, reloads all controller software and the monitor/firmware, and starts execution. On Honeywell systems, both async and bisync communications settings, if any, will be preserved. The Processes are brought back in from disk and execution resumes exactly where it left off.

<u>Warmstarting after Bootstrapping</u>

After selecting the (W)armstart option at the System Startup options prompt, the system configuration is displayed, as shown in Figure D of Section 2.2, BOOTSTRAPPING THE SYSTEM.

The system then displays the following message along with the system command level (TCL) prompt (>):

```
The date and time must be re-set!!!

>
```

In order to log account and file statistics correctly, you must use the SET-TIME and SET-DATE system commands to set the current time and date as the system time and date. The time is expressed on a 24-hour clock, as:

```
hh:mm{:ss}       (example:  08:23)
```

The date may be expressed as:

```
dd mmm yy{yy}     (examples: 06 SEP 1985 or SEP 06 85)
dd/mm/yy          (example: 12/13/85)
```

Warmstarting from the System (TCL) Command

The :WARMSTART command first flushes memory and then enters the warmstart procedures described above.

Warmstopping the System

Warmstopping should be used whenever the system is taken down in an orderly manner (not via a crash or failure). This procedure first flushes memory and then halts the system.

Section 2.12, SHUTTING DOWN THE SYSTEM discusses warmstop procedures. (See also the :WARMSTOP command, listed in the System Commands Guide.)

2.4 BRINGING UP ULTIMATE ON A DEC VAX SYSTEM

The implementation of the Ultimate Operating System on a DEC VAX system is different from other installations. The Ultimate system and VMS, the native VAX operating system, are designed to co-reside within the same computer. The booting, coldstarting, and user logon procedures are, therefore, special cases of System Management and Support.

Transporting the Ultimate Operating System to the VAX is accomplished through a combination of hardware, and software. The hardware comes in the form of an "Ultimate Engine" that is connected to the Unibus. The Ultimate Engine consists of an Ultimate CPU and Ultimate Memory (up to 4 megabytes).

The software is made up of the following components:

1. Ultimate Engine Device Driver.
   This is a system program that runs under VMS. It is designed to activate the Ultimate Engine and to satisfy all the I/O requests of the Ultimate Engine. These I/O requests (for disk, tape, printer, and terminal I/O) are passed on to VMS by this device driver.

2. Ultimate Connection Requestor
   This image is run by any user wishing to gain access to the Ultimate Operating System. It informs the Device Driver that this terminal wishes access to the Ultimate Operating System. When this request for access is granted, the user is prompted with the Ultimate Logon message. The user then logs to an Ultimate account. When the user logs off, the VMS environment is restored.

3. Miscellaneous Detached Processes
   These are detached processes that are responsible for handling Process context requests for the Ultimate Device Driver.

4. Ultimate Operating System
   This is the code that is executed by the Ultimate Engine. This code is identical for all implementations of the Ultimate Operating System.

5. File Transfer Utilities
   These utilities run under the Ultimate Operating System and will transfer unpacked ASCII data between the two operating systems.

6. Ultimate Environment Utilities
   These utilities run under VMS and maintain the Ultimate Configuration files, which contain parameters defining the Ultimate Environment.

7. Initialization and Shutdown Utilities
These utilities run under VMS and carry out the initialization of the Ultimate Control Block and the Ultimate Engine. They are also responsible for the orderly shutdown of the Ultimate Operating System.

On a VAX system, the native operating system is VMS. When the VMS operating system is booted, the bootstrap process has been modified to allow loading of the Ultimate Engine Device Driver and the other Ultimate system programs that run under VMS (see above).

The command to boot VMS from a VAX console varies, depending on CPU type; refer to your VAX reference manual.

The Ultimate system programs that will run under the Ultimate Operating System must be loaded via a coldstart procedure from the VMS environment. Each time VMS is booted, the Ultimate system must be coldstarted. The command to do a coldstart from VMS is:

```
RUN ULT$COLDSTART
```

This procedure has similar messages as the bootstrap and coldstart procedures covered in Sections 2.1 and 2.2. When the bootstrap process is completed, the System Startup Options are:

(C)oldstart
(W)armstart
(F)ile Restore
(D)isk coldstart

Selecting (C)oldstart loads the COLDSTART and ABS (third and fourth files of SYS-GEN tape) into the CPU and coldstarts the system. (Don't forget to skip 3 files at the "Mount ABS tape and enter number of files to skip, if any" prompt.) See Section 2.2, COLDSTARTING THE SYSTEM, for details.

Selecting (W)armstart reloads the Monitor (Kernel) without initializing the virtual system. It restarts each user Process without loss of work that was in progress at the time of a system warmstop. This option is not available on the VAX at this time.

Selecting (F)ile Restore performs a coldstart as in (C) above, and loads the system and user files saved on a DATA tape (or the FILES section of the SYS-GEN tape; multiple tapes may be used in the restore. This is the option to restore an entire system.

Selecting (D)isk coldstart performs a coldstart as in (C) above in that all Processes are restarted, but the tape is not used. The ABS section of the disk is used as is.

Once the Ultimate system is started up, it remains in the background until a user requests access to the Ultimate system. The user must be logged onto a VMS account. The command to logon to Ultimate is:

```
RUN ULTIMATE
```

This procedure uses the Ultimate Connection Requestor software to allow the user to log onto an Ultimate account. The Ultimate Logon message is displayed on the terminal:

```
Date   Time   Logon Please:
```

The user must enter an Ultimate user account name (and password, if applicable) in order to bring up that account's Logon proc and begin work in the Ultimate environment. Once the Ultimate Operating System is invoked, it runs identically to every other Ultimate system.

When the user logs off from the Ultimate system:

```
>OFF
```

the system returns to the VMS Operating System environment (DCL).

In order to shut down the Ultimate system prior to maintenance or file save procedures, you must execute the VMS command to shut down Ultimate:

```
RUN ULT$SHUTDOWN
```

Please refer to the Ultimate VAX/VMS Installation and Operation Guide for a complete reference to the VAX implementation files, programs, and procedures.

2.5 BRINGING UP A VIRGIN SYSTEM

When a new system is installed, it is considered a "virgin system". Each manufacturer-based system has a slightly different procedure for bringing up a virgin system.

Bringing Up a New Honeywell System

To bring up a new Honeywell system, it is necessary to format the disk(s) to Ultimate specifications and then load the Ultimate system software onto the disk(s). Follow all steps below.

Bringing Up a New DEC System (other than VAX)

To bring up a new DEC system, it is necessary to load the Ultimate system software onto the disk(s). Start at Step 5 and follow the steps below.

Each system is "burned in" prior to shipment. This includes formatting of the disk(s). If you encounter any problems in bringing up your system, please call Ultimate Support for assistance.

Bringing Up a New DEC VAX System

Please use the procedures in the document "Installing Ultimate on VAX/VMS Systems", which is included with your Ultimate release tape.

Steps to Bringing up a Virgin System

1. On a system that has a tape drive, use step a; on a disk-only system, use step b.

   a) Tape System

   Mount SYS-GEN tape and bring to load point and on-line.

   Mount an appropriate disk pack if using removable disks.

   Bring the disk drive(s) to the ready position.
   Go to Step 2.

   *NOTE - The start and ready lights should be on and protect light(s) should be out

   b) Disk Only System

   Mount the SYS-GEN disk and bring the disk drive to the ready position. Go to step 2.

   *NOTE - The start and ready lights should be on and protect light(s) should be out

2. Boot the system. The Options message will appear on line zero.

   *NOTE - On DEC systems, the terminal should be at 9600 baud, No parity. On Honeywell systems, it should be 9600 baud, Even parity.

3. "Enter Option(s) or ? for Help:" Enter 'D' to load the Ultimate Diagnostics Monitor.

4. Format the disk.

   *Note - (See Section 2.10, USING THE DIAGNOSTICS MONITOR, for disk formatting procedures.) On a disk-only or any CMD system, format the FIXED disk.

5. Boot the system. The Options message will appear on line zero.

6. Enter 'F' at the Startup Options prompt. This initiates a complete File Restore.

7. At the "Enter system serial # or <CR> to accept" prompt, enter your System Serial Number (i.e., the 'TN', 'GE', or other number of your system.)

   Example: TN1234 or GE1234

   The system will re-display the number. You may re-enter the correct number if a mistake was made. Press <CR> after the correct number is entered.

8. At the "Mount ABS tape and enter number of files to skip, if any" prompt, press <CR>.

   NOTE: For VAX systems, mount the SYS-GEN tape and enter "3". Press <CR>.

9. At the "Mount DATA tape and press RETURN" prompt, press <CR>. *NOTE - DO NOT change SYS-GEN Tape/Disk. Ultimate's base system files will now be displayed on the left hand side of the terminal screen.

10. At the "seq # of this data tape 0 0 0 0" "Is this the right tape (Y/N)?" prompt, enter 'Y' and press <CR>. The File Restore will now begin.

11. At the "UPDATE/TRANSACTION TAPES (Y/N)?" prompt, enter 'N' and press <CR>.

12. At the "This is the Cold-Start Procedure, Enter <CR> to continue." prompt, press <CR>.

13. Enter the time and date when prompted

    a. "Time=" Enter HH:MM:SS <CR>
    Enter time in military format (17:00:00 for 5PM)

    b. "Date =" Enter MM/DD/YY <CR>
    Enter date as 02/03/86 for Feb 3, 1986)

    The system will display several messages. If there are any error messages such as "System does not verify" call the Ultimate System Support Group immediately.

14. At the "Logon Please" prompt, log onto the SYSPROG account to load UltiWord (the Word Processing account).

    Mount SYS-GEN tape and bring to load point.

    Select the program 'Load WP account from SYS-GEN tape' from the SYSPROG Main Menu. The program will search for and load the WP account.

15. Remove the SYS-GEN tape/disk from the drive.

    *Note - on a Honeywell disk-only system, replace SYS-GEN disk with a spare, formatted disk-pack.

16. You may now begin to load your application software.

17. It is recommended that you create one or more copies of the SYS-GEN or Boot tape. Or, if you have two or more tape drives, make one or more copies of the SYS-GEN tape. This is to ensure that you do not run into the situation where you need to coldstart the system for some reason but your only SYS-GEN tape is missing or has too many parity errors.

    Instructions to create or duplicate a SYS-GEN tape or create a Boot tape are covered in Sections 2.7, CREATING A SYS-GEN OR BOOT TAPE, and 2.8, DUPLICATING SYS-GEN TAPES.

2.6 UPGRADING FROM A PREVIOUS RELEASE

Upgrading an Ultimate system from a previous release involves three major methods. The actual procedures vary from release to release.

The three upgrade methods are:

1. File Restore from SYS-GEN tape.

2. File Restore from File Save tape.

3. Selective Upgrade.

The correct method depends on the current release your system is running on. Please refer to the instructions given in the Upgrade Procedure Document.

To obtain the Upgrade Procedure Document, from the SYSPROG Main Menu, select the Documentation Menu. From the Documentation Menu, select and print the Upgrade Procedure Document.

2.7 CREATING SYS-GEN OR BOOT TAPES

A SYS-GEN tape contains the entire system plus the user file data base. A Boot tape contains the entire system but no user files. It is recommended that SYS-GEN and Boot tapes be created to ensure that a current version of the system is always available for system regenerations, if needed.

Creating a SYS-GEN tape

A SYS-GEN tape with all user data files can be created by the SYS-GEN system command.

NOTE: This tape cannot be used for Upgrade procedures.

The WP account must be on the system in order to create a SYS-GEN tape. If it is not found, the message:

```
WP ACCOUNT NOT ON SYSTEM
CANNOT MAKE SYS-GEN TAPE
```

will be displayed. If this occurs, you can load the WP account from the SYSPROG menu and then re-execute SYS-GEN.

If the WP account is present, the SYS-GEN command will proceed to dump the entire system onto tape. The dump will be similar to the SYS-GEN tape format, but may have additional accounts. (See Section 1.5, ULTIMATE SYSTEM SYS-GEN TAPES for the SYS-GEN tape format.)

The SYS-GEN command is a proc that executes a special form of the SAVE command with the 'Z' option: SAVE(D,F,G,S,T,Z). The 'Z' option produces a tape with all zeroes in the sequencing information segment; this distinguishes it from other file save tapes. For details on the SYS-GEN command, please see the System Commands Guide.

Creating a Boot Tape

One or more copies of the Boot tape should be created after each upgrade to a new system Revision. The procedure is:

1. Mount a new blank tape, with write ring, on the tape drive and set to load point and on-line.

   For Honeywell disk-only systems: Mount a disk cartridge and put the disk drive on-line and ready.

   For systems with more than 2 tape drives: Mount the tape on tape drive 0.

2. Logon to SYSPROG. Select the "Create Boot Tape" option on the Main Menu.

   (Or, at TCL, enter **CREATE-BOOT**.)

3. Remove the tape when the program is finished.

4. Affix a tape label to the tape; mark the Revision number and tape density (800 bpi, 1600 bpi, etc.) on the label. To find out the current operating system Revision number, at TCL, enter **REV.** The number shown as "Abs.rev" is the System Revision number.

5. If you wish more copies, repeat steps 1-4.

For details about the CREATE-BOOT system command, please see the System Commands Guide.

2.8 DUPLICATING SYS-GEN TAPES

It is recommended that duplicate SYS-GEN tapes be made for backup. The complete system is required. It is important that a file save be done, and that you are prepared to do a File Restore after the SYS-GEN tape duplication.

Duplicating SYS-GEN tapes with the T-COPY Command

In order to use the T-COPY command, your system must have the following hardware configurations:

1. TWO (2) on-line tape drives (to duplicate SYS-GEN tapes) OR

2. ONE (1) on-line tape drive and ONE (1) on-line CMD (Cartridge Module Drive) disk drive (to duplicate SYS-GEN tapes/cartridges).

If your system does NOT have the required hardware configurations, see the procedure outlined below (without T-COPY). Follow these steps if you plan to use T-COPY:

1. Mount a SYS-GEN tape/cartridge, and bring to load point and on-line.

2. Logon to the SYSPROG account, and go to TCL.

3. Enter the command 'T-ATT n' and press <CR>.

   *NOTE - Where n = 'C' for CMD disk drive, or n = the tape drive number on which the SYS-GEN tape was mounted.

4. Mount a blank tape, or an Ultimate formatted cartridge, on the other tape/CMD drive.

5. Enter the command 'T-COPY m (E,U)' and press <CR>.

   *NOTE - Where m = 'C' for CMD drive, or m = tape drive number on which blank tape was mounted.

   The contents of the MASTER SYS-GEN tape/cartridge will be copied to the blank tape/cartridge.

   After the T-COPY verb is completed, the cursor will return to the TCL prompt.

6. Enter the command 'T-REW' and press <CR>.

   *NOTE - This will rewind the MASTER tape. Remove the COPY SYS-GEN tape/cartridge. Mount another blank tape/cartridge, and repeat Steps 5 & 6 for any additional copies needed.

Duplicating SYS-GEN tapes without using the T-COPY verb

Follow the steps given below if your system does not meet the hardware configuration requirements given above.

1. File-Save the system.

2. Mount the SYS-GEN tape.

3. Boot the system.

4. At the "Enter Option(s) or ? for Help:" prompt, enter 'F'. Complete the file-restore using the SYS-GEN tape/disk ONLY.

5. Logon to the SYSPROG account.

6. In the SYSPROG Main Menu, select the appropriate program to load the WP account.

7. Remove the SYS-GEN tape/disk.

8. Mount a blank tape, or Ultimate formatted cartridge.

9. At TCL in the SYSPROG account, enter 'SYS-GEN' and press <CR>. When SYS-GEN is complete, the tape will unload automatically.

10. To make additional SYS-GENs, repeat steps 8 and 9.

11. Upon completion:

    a. Mount SYS-GEN tape/cartridge and place on-line.

    b. Boot the system.

    c. At the System Startup Options prompt, enter 'F' to start a File Restore.

    d. At the "Mount DATA tape and press RETURN" prompt, mount the first tape/disk of latest File-Save and press <CR>. This starts a complete File-Restore with all tape(s)/disk(s), of the latest File-Save.

2.9 FORMATTING AND DUPLICATING DISKS

Disks need to be formatted and duplicated only for Honeywell disk-based systems. The DISK.COPY command allows rapid backup of a multi-removable-disk system by copying the on-line disk set to a back-up set of disks.

Hardware Requirements for Disk Copy

1. Two or more disk drives (5 or 10 platter removable).

2. At least one set of back-up disks.

3. The set name of the 'copy' set must be different from the name of the on-line set.

Formatting Disks

To format a disk, the recommended procedure is to boot the system, and then select the (D)iagnostics Monitor option.

The 'D' option puts the Diagnostics Monitor software in control of the system. The Diagnostics Monitor is detailed in Section 2.10, THE DIAGNOSTICS MONITOR.

Features and Recommendations about Disk Copying

Speed — DISK.COPY will copy one complete 75MB disk pack in under 4 minutes. A 288 MB pack takes approximately 15 minutes.

Data Save Only — DISK.COPY copies only the data portion of the disk, and not the empty overflow area. If you wish to copy the entire disk pack, use the 'L' option on the DISK.COPY command.

Alternate track handling — DISK.COPY handles existing bad tracks on both read and write (master and copy) packs.

Disk error handling — If DISK.COPY encounters a read or write error, up to 12 retries are attempted (if necessary) before a message is displayed. You may retry, accept, or quit the disk copy at that time.

Before Copying — Determine the sequence # of each pack by using the D-RDLBL verb on the Diagnostics Monitor. To enter this Monitor, after booting the system select the (D)iagnostics Monitor option from the System Startup Options.

Label each pack on the pack 'cover'. Do NOT put a label on the disk pack itself. If the label comes off, it can cause a head crash. To label the pack itself, use a Magic Marker on the pack hub.

Duplicating Disks (Disk Copy)

1. Stop all activity on the system and wait until the traffic light becomes only a dim flicker. (This indicates that memory has been flushed.)

2. Logto SYSPROG and execute a :WARMSTOP command to bring the system down in a controlled manner.

   NOTE: Do not enter "SET-MAX-LINES 1" before initiating DISK.COPY since another "line" (that of the Warmstart Process) will be used for the DISK.COPY, in "off-line" mode.

3. Boot the system. At the System Startup Options, enter "O,W" to enter the Off-line Monitor and then do a warmstart.

4. From line zero, logon to SYSPROG and request execution of the DISK.COPY command. (To copy the entire pack instead of data only, use the 'L' option -- DISK.COPY (L).

   From this point on, you are effectively running a memory only system. Only those frames absolutely necessary to copy disks have been locked into memory. No further virtual communication with the disk set is possible.

   NOTE: Although you may break into the Assembly Debugger, the END command will not work. The only exit from Debugger is to type "G373.1" to restart the copy routine.

5. At the prompt to "Enter the sequence # to copy or "0" for a complete backup", use "0" for a multi-disk backup. To copy a single disk, specify its sequence number.

6. At the prompt for the 'copy' set name, enter the set name of the back-up packs.

   NOTE: All drives must be ready, since the on-line (master) set will automatically be verified now.

7. At the prompt to "Mount seq# 1 master and copy packs", mount pack 1 of n of the copy set. (Pack 1 of n of the master set is already mounted.) Press <CR>.

a. If one of the drives is not ready or the correct packs cannot be found, the message "Missing master/copy packs # of n" is displayed.

b. If the 'copy' drive is 'write protected', a message is displayed.

c. For added safety, you may protect the master pack.

8. The message "Copy in Progress" is displayed. Below it, an asterisk (*) will be printed for each 16 cylinders that are copied. This gives you a visual indication that the copy is proceeding.

9. After each copy is complete, you are prompted for the next set of packs (both master and copy). You may place the master and copy packs in any drive. The labels of all packs mounted in 'ready' drives will be read to locate and set up the proper master and copy packs for the next copy sequence.

10. After the last pack has been copied, the system returns to the first message (step 5 above). If you are finished, place one of the sets back in your disk drives and perform a normal boot and warmstart to bring the system back up.

If you need additional details about the DISK.COPY command, see the System Commands Guide.

2.10 USING THE DIAGNOSTICS MONITOR

> The Diagnostics Monitor is a free standing, memory resident program designed for off-line system maintenance. The major uses are disk formatting, boot-loading, configurations display and restoring T&V's (Test and Verification procedures) to disk from tape.

The Diagnostics Monitor operates in an off-line environment. Only line 0 is activated, and the system is not able to run in the normal mode. On either DEC or Honeywell systems, the Diagnostics Monitor may be loaded from either tape or disk.

Before Loading the Diagnostics Monitor

Entry to the Ultimate Diagnostics Monitor is made by selecting the 'D' option at the System Startup options when booting the system.

Before booting the system, be sure that all users are logged off, and then do a ":WARMSTOP" from the SYSPROG account. This ensures that all data in memory is flushed to disk.

Initiating Diagnostics Monitor from Honeywell-Based Systems

Mount the SYSGEN tape and boot the system. The system will display the following message on line 0:

```
This is the Ultimate Operating System

System Startup Options:
(B)oot
(C)oldstart
(D)iagnostic Monitor
(F)ile Restore
(O)ff-line Monitor
(U)tilities Monitor
(W)armstart

Enter Option(s) or ? for help:
```

Select D. The system automatically displays the tape label.

Initiating Diagnostics Monitor from DEC-Based Systems

If booting from tape:

1. Toggle the BOOT switch on the front panel.
2. Load a SYS-GEN tape in the tape drive.
3. Press <CR> to display the Startup options.

If booting from disk:

1. Toggle the BOOT switch on the front panel.
2. Enter <CTRL/D> (CONTROL plus D) at the BOOT prompt.

Select D from the Startup options, as above.

The Diagnostics Monitors for DEC and Honeywell Systems

Line zero (0) will be the only active terminal line when running from Diagnostics Monitor.

The system is now in the Diagnostics Monitor. The Diagnostics Monitor procedures, commands, displays, etc. are different on DEC-based and Honeywell-based systems. Each type of system is documented in a following subtopic:

Section 2.10.1 Diagnostic Procedures on DEC Systems
Section 2.10.2 Diagnostics on Honeywell Systems

2.10.1 DIAGNOSTIC PROCEDURES ON DEC SYSTEMS

DEC diagnostics are menu-driven. The Formatter Menu contains the options from which to select a task to perform.

The Diagnostics Monitor Load-Completion Message

The completion of loading the Diagnostics Monitor will be indicated by a message such as the following on line zero:

```
******************************************************
             5 1/4 inch Disk Formatter

     Revision: 9.0
     Date:  23 AUG 1985
*******************************************************

**********************NOTICE***************************
Cylinder, head, and sector numbers are displayed
in DECIMAL.  All required inputs to specify cylinder,
head, and sector must be in DECIMAL.
*******************************************************

press RETURN to continue
```

Pressing <CR> causes the Monitor to search for the disk drive(s) in the system. If no drive is found, the Monitor displays a message:

```
Can't find DISC drive!!!
```

and the system halts.

If the drive(s) are found, the Monitor displays a prompt to select the drive type:

```
select drive TYPE:
     1 = CMI CM5619
     2 = CMI CM6640
     3 = CDC 94155-48
     4 = CDC 94155-86

enter no.:
```

Enter the number corresponding to your drive type. If you are not sure of the drive type for your system, call Ultimate Support for assistance.

THE FORMATTER MENU

After the drive is selected, the Monitor displays the Formatter Menu:

```
Formatter Menu

 1 = Format Disc(s)
 2 = Write Boot and Label
 3 = Add Alternate Sector
 4 = Erase Disc Label
 5 = Display Bad Sector Table
 6 = Display Alternate Track Table
 7 = Erase DISC Label
 8 = BOOT System

enter no.:
```

To exit from the Formatter Menu, you may reboot the Ultimate operating system, via the BOOT option: Enter "8" and press <CR>.

Option 1 is the most commonly used task. Each option is detailed below.

FORMAT DISC(S) (Option 1)

Normally, formatting your disk(s) will only be necessary under the following conditions:

1. It is a replacement disk drive being installed.
2. Your existing disk drive has so many random errors that reformatting your drive is the only alternative. If you have only a few errors, then see the option 3, Add Alternate Sector.
3. You are adding another disk drive.

NOTE: If you are reformatting your existing drive(s), then a file-save must be done first! A file-restore will have to be done after the format is completed.

The FORMAT routine writes headers. The headers identify sector, cylinder, and head numbers. The FORMAT routine also verifies the integrity of the disk media, assigns spare sectors and alternate tracks when necessary, reconfigures the disk, and writes the label, disk boot, Bad Sector Table, and Alternate Track Table. Following are the seven phases that make up a format operation:

```
FORMATTING DISK       VERIFYING DATA
WRITING DATA          VERIFYING DATA
VERIFYING DATA        RECONFIGURING DISK
WRITING DATA
```

Starting the Format Routine

IMPORTANT!!! If upgrading from a release prior to R110T, see 'Important Note' in the introduction of the Upgrade Procedures Document (available from the SYSPROG Documentation Menu).

To start the FORMAT routine, select '1' and press <CR> from the Formatter Menu. If your system has two disk drives, you are prompted for which drive(s) to format:

```
Format Drive 0,1 (B)oth?:
```

Enter "0" to format drive 0, or "1" for drive 1, or "B" for both drives. Press <CR>.

The following message is displayed:

```
FORMATTING drive #x    where:  x = 0 or 1
```

A warning message allows you to exit the Format routine without performing any formatting on the disk:

```
This operation will destroy ALL data on disk.
(C)ontinue or e(X)it:
```

A 'C' response initiates the formatting operation. An 'X' response exits the routine.

During the formatting operation, bad sector(s) may be encountered. Normally, the routine automatically assigns the data to an alternate sector. The operator may choose to manually assign the bad sector(s). The prompt is:

```
Assign ALTERNATE sector? (Y/N):
```

Enter one of these five valid responses:

N	no operator entries are to be made
<CTL/C>	clear the Bad Sector or temporary table
<CTL/V>	display the Bad Sector or temporary table
<CTL/D>	delete an entry from Bad Sector or temporary table.
Y	Bad Sector Table entries available

An "N" reply means there are no (more) operator entries to be made and the formatter process will continue. All other responses loop back to the same prompt until an "N" is entered.

If the <CTL/D> function is entered, the operator is prompted for the cylinder, head, and sector number to delete:

```
*<cyl>,<hd>,<sec>:
```

If a "Y" reply is given, the operator is put in an input mode. The same prompt for cylinder, head, and sector number is issued. When the operator specifies a sector, it is added to the appropriate table and the prompt repeats. The Process remains in this mode until only a <CR> is entered or an entry error is made.

Once the "Assign ALTERNATE sector" prompt has been completed, the formatting process continues through the seven phases listed above.

Formatting Displays

While each phase is in process, the cylinder and head numbers of the track that is being written is displayed in DECIMAL. Message headings are also displayed for sector, disk drive status, disk error status, and control status. The sector number will always be zero unless an error is detected. The status displays are null unless an error is detected; the error status codes are displayed in HEXADECIMAL.

Writing Headers Display

Disk headers are written a track at a time. The following two messages are displayed while the headers are being written:

```
          Writing HEADERS for drive #x  where:  x = 0 or 1

          cyl     hd     sec    rks1   rks2   rkds    rker
        nnnnn   nnnnn  nnnnn  hhhh   hhhh   hhhh    hhhh
                             |____________|  drive   error
                      control status 1 & 2  status  status
```

Verifying the Disk Display

A worst case pattern is written and verified twice to detect bad sectors on the disk. If any bad sectors are found, they are entered in the Bad Sector Table. Four complete passes of the disk are done in this phase. The following messages are displayed:

```
           Writing DATA to drive #x  where:  x = 0 or 1

         cyl     hd     sec    rks1   rks2   rkds    rker
        nnnnn   nnnnn  nnnnn  hhhh   hhhh   hhhh    hhhh
                              |___________| drive   error
                       control status 1 & 2 status  status

           Verifying DATA on drive #x  where:  x = 0 or 1

         cyl     hd     sec    rks1   rks2   rkds    rker
        nnnnn   nnnnn  nnnnn  hhhh   hhhh   hhhh    hhhh
                              |___________| drive   error
                       control status 1 & 2 status  status
```

<u>Reconfiguring the Disk Display</u>

Reconfiguring the disk is the process of moving the last sector flag from the last sector to the last non spare sector; effectively deleting the spare sectors from the disk. In order to accomplish this, the headers for every track on the disk must be rewritten. The following two messages are displayed during the reconfiguration process:

```
               RECONFIGURING drive #x  where: x = 0 or 1

         cyl     hd     sec    rks1   rks2   rkds    rker
        nnnnn   nnnnn  nnnnn  hhhh   hhhh   hhhh    hhhh
                              |___________| drive   error
                       control status 1 & 2 status  status
```

Error Messages for Reformatting a Disk

The messages for reconfiguring a disk are:

**** Header Read Error ****

During the reconfiguration process a disk error was detected while reading a track's headers. An alternate track is assigned if this should occur and reconfiguration proceeds as if nothing happened.

Alternate Header Read Error Configuration error - REFORMAT disk

While trying to read the headers of an alternate track during the reconfiguration process a disk error was detected. There is no recovery from this error; the FORMAT routine is aborted.

*** can't find LAST sector *** Configuration error - REFORMAT disk

The last sector on the track being reconfigured cannot be found. There is no recovery from this error; the FORMAT routine is aborted.

**** Header Write Error ****

A disk error was detected while trying to write the reconfigured track's headers. An attempt is made to assign an alternate track; if successful the reconfiguration process continues.

-- ALTERNATE track assigned --

An alternate track was successfully assigned to a track that had a header read or write error.

Error messages for validation of the disk are:

**** error track WITHIN Sysbase - REJECT drive ****

There is an alternate track below the Sysbase cylinder. The drive will still function but system performance can be seriously impaired.

**** MAX bad tracks EXCEEDED - REJECT drive ****

The number of alternate tracks assigned exceeds the number of bad tracks allowed for this disk. The disk drive will still function, however performance is seriously impaired.

The error messages for Spare Sectors and Alternate Tracks are:

**** can't find ERROR sector ****

While trying to assign a spare sector, the sector addressed by a Bad Sector table entry could not be found.

**** MAX bad tracks EXCEEDED - REJECT drive ****

While trying to assign an alternate track, the number of bad tracks allowed for this drive was exceeded.

WRITE BOOT AND LABEL (Option 2)

This program is used to rewrite the disk BOOT and LABEL on disk. When selected, the Bad Sector and Alternate Track tables are read from disk. The BOOT and LABEL are then rewritten on disk along with the Bad Sector and Alternate Track tables. This procedure is done for both disks if there are two disks in the system.

When these have been written, the Formatter Menu is displayed. The primary purpose for this entry is to enable updating of the disk BOOT without going through the entire FORMAT process, which destroys all the data on disk.

ADD ALTERNATE SECTOR (Option 3)

This entry provides a means of assigning a spare sector or alternate track to a disk while preserving the integrity of the data on the disk. Each type of disk has a preassigned number of cylinders that are reserved specifically for this purpose. These cylinders are used when an alternate track is required. Spare sectors are assigned on a track until they are exhausted, then an alternate track is automatically assigned if one is available. The Bad Sector and Alternate Track tables are updated on disk to reflect any changes made.

Only one sector at a time can be specified. If an entire track has to be replaced, an appropriate number of sectors must be specified to exceed the number of spare sectors on that track. The following is the prompt displayed to assign a spare sector:

```
*<cyl>,<hd>,<sec>:
```

Error Messages for Add Alternate Sector

Spare CYLINDERS exhausted CANNOT re-assign sector

> The pre-assigned number of spare cylinders for this drive has been exhausted. There is no more space for an alternate track. The disk drive should be reformatted.

MAX bad tracks EXCEEDED

> The maximum number of bad tracks that are allowed have been exceeded. The disk drive should be replaced.

Header ERROR CANNOT re-assign sector

> A header error was detected while trying to re-assign the bad sector. The disk drive should be reformatted.

ERASE DISC LABEL (Option 4)

This entry is used to erase the disk's LABEL. This routine is provided to ensure a 'fresh' reformat of a previously formatted disk.

When there are two disk drives in the system, the labels of both drives are erased. Before the label(s) are erased, the following message is displayed to ensure that this is the operation that was intended:

```
This operation will ERASE the disk LABEL
(C)ontinue or e(X)it:
```

If this routine was entered by mistake, enter "X" to abort the routine without erasing the labels. To erase the labels, enter a "C".

DISPLAY BAD SECTOR TABLE (Option 5)

The Bad Sector Table display routine displays sectors that have been flagged as being bad by the operator or FORMAT routine when the disk was last formatted. The table is located on disk; every disk in the system has its own Bad Sector Table.

If there is only one disk drive in the system, the drive select phase of the table display is skipped. In two-drive systems, the operator is prompted for the drive:

```
Drive? (0-1):
```

For drive 0, enter "0". For drive 1, enter "1". Press <CR>.

If the disk is not formatted or the table is 'null', the following message is displayed:

```
table NULL
```

If there are entries in the table, the following decimal display format is used:

```
cyl     hd    sec
 |       |     |
 |       |     +---bad sector number
 |       +---------head address
 +-----------------cylinder address
```

A PAUSE is entered after a maximum of 18 lines or the last entry is displayed. To CONTINUE press any keyboard key and the next screen will be displayed or the table display exited.

DISPLAY ALTERNATE TRACK TABLE (Option 6)

The Alternate Track Table is a record of tracks that have an excessive amount of errors and have therefore been assigned alternates. The table is located on disk; every disk has its own Alternate Track Table.

If there is only one disk drive in the system, the drive select phase of the table display is skipped. In two drive systems, the operator is prompted for the drive number:

    Drive? (0-1):

For drive 0, enter "0". For drive 1, enter "1". Press <CR>.

If the disk is not formatted or the table is 'null', the following message is displayed:

    table NULL

If there are entries in the table, the following decimal display format is used:

```
cyl     hd    alt
 |       |     |
 |       |     +---alternate cylinder in which
 |       |         track is relocated
 |       +---------head address
 +-----------------cylinder address
```

A PAUSE is entered after a maximum of 18 lines or the last entry is displayed. To CONTINUE press any keyboard key and the next screen will be displayed or the table display exited.

READ DISC(S) (Option 7)

This entry allows for checking disk(s) determine if they contain any disk errors. It reads all tracks on the disk(s) and reports any errors found.

To start the READ routine, select '7' and press <CR> from the Formatter Menu. If your system has two disk drives, you are prompted for which drive(s) to read:

    Read Drive 0,1 (B)oth?:

Enter "0" to read drive 0, or "1" for drive 1, or "B" for both drives. Press <CR>.

Displays During the Read Procedure

During this procedure, the cylinder and head numbers of the track that is being read is displayed in decimal. Message headings are also displayed for sector, disk drive status, disk error status, and control status. The sector number will always be 00000 unless an error is detected. The status displays are null unless an error is detected; the error status codes are displayed in hexadecimal.

The following status message is displayed while the disk(s) are being read. The status columns are used only when a disk error is found on a particular sector. A new message then starts on the next display line so that a historical record of errors is available on the screen.

```
           Reading HEADERS for drive #x  where:  x = 0 or 1

             cyl     hd     sec    rks1    rks2    rkds    rker
           nnnnn   nnnnn  nnnnn   hhhh    hhhh    hhhh    hhhh
                                 |___________|    drive   error
                          control status 1 & 2    status  status
```

BOOT System (Option 8)

This entry allows booting the system from tape or disk without pressing the BOOT switch on the control panel. It operates exactly as if the BOOT switch had been pressed. The following prompt is displayed:

PUT TAPE ON-LINE, PRESS <CR>

To boot from a loaded SYS-GEN tape, press <CR>. If a SYS-GEN tape is already loaded and beyond the load point, it will rewind automatically. Or, to boot from disk after a warmstop, press <CTRL/D>; that is, hold down the CONTROL key while pressing "D".

After booting, the system displays the System Startup options. (See Section 2.1, Bootstrapping the System, for more information about the Startup options.)

2.10.2 DIAGNOSTICS ON HONEYWELL SYSTEMS

The Diagnostics Monitor is command-driven. The major use of this Monitor is to format disks. Secondary purposes include displaying the system configuration and other diagnostic tasks. Each command is explained in this section.

The Diagnostics Monitor Load-Completion Message

The completion of loading the Diagnostics Monitor will be indicated by a printout of the tape label and display of the following message on line 0:

(TAPE LABEL)

```
This is the Ultimate Diagnostics Monitor
Revision ##                    (date)

:>
```

To see a list of commands, enter "HELP" and press <CR>.

Diagnostics Commands

The verbs used in Diagnostics Monitor commands are:

AD-ALT	IO
BOOT	RAT
CERTIF	RD-DSK
CONFIG	T-ATT
D-RDLB	T-DET
D-WTLB	T-FWD
DTEST	T-REW
DTX	T-SIZE
FORMAT	T-UNLOAD
HELP	XTD

To exit the Diagnostics Monitor, you may use the BOOT verb to reboot the Ultimate Operating System. During a command operation, you may exit to the Monitor prompt (:>) by pressing the <BREAK> key and typing "END".

AD-ALT

Syntax: AD-ALT channel-#

This verb is designed to assign a marginal disk track (one that is causing disk errors) to an alternate track without the necessity of a save, format, and restore. AD-ALT can be used for those errors that consistently repeat in the SYSTEM-ERRORS file (same cylinder, same track). This command reads the track and cylinder you specify, one sector at a time, making up to 10 attempts (if necessary) to read each sector.

Each step is described below:

1. If you have a Storage Module Drive (SMD), skip to 2. If you have a Cartridge Module Drive (CMD), you are prompted:

    (F)ixed or (R)emoveable?

    Enter 'F' for fixed or 'R' for removable.

2. The defective space table will be displayed on the screen. Do not enter these cylinders and tracks that are displayed.

3. Respond to the prompt for a cylinder number:

    Cylinder # (in decimal) :

by entering a 3-digit (decimal) cylinder number. If you decide not to add an alternate track, you may enter <CR> to exit the command.

4. Respond to the prompt for a track number with a 2-digit (decimal) number.

5. Respond to the cylinder # and track # confirmation prompt:

Cyl 350 track 00 sector 00 Is this what you want? (<CR>=Y/N):

with Y or <CR> to proceed, and N to re-enter the parameters.

6. The Monitor attempts to read the bad track one sector at a time and transfer data to an alternately assigned track. If an error occurs, you will be given the options to (R)etry the read again, (I)gnore the error and continue saving the rest of the track, or (Q)uit, which will not mark the track defective.

If you (R)etry and do not get an error, all data has been successfully transferred and the system is all right to use. If you (I)gnore the error, you should check your files for group format errors or loss of data. For additional help, call your dealer or Ultimate for assistance. If you (Q)uit, you will have to do a file-save, reformat the disk, and then restore the data.

The message 'Alternate added' means that data was transferred successfully to the alternate track without error.

7. The command returns to the prompt:

    Cylinder # (in decimal) :

You may enter <CR> to exit the command, or enter another marginal disk track to reassign.

BOOT Verb

Syntax: BOOT channel-#

This verb allows the operator to boot-load any disk-type device on the system without the use of a full control panel. Channel-# is the 3 or 4 hex digit bus address assignment (any leading zeros may be omitted) as it would be entered in register 'D1' on a full control panel.

For example: To boot Honeywell's T&V's from a disk configured on channel # 600, enter 'BOOT 600'.

A message will be displayed indicating the type of the device on channel # 600 and the boot will be synthesized.

Note: Honeywell's T&V's search for a 1200 baud console on the numerically highest communications channel # configured.

CERTIF Verb

Syntax: CERTIF chan# {chan# ...}

CERTIF operates the same as the FORMAT verb, except that it makes three passes instead of one on the Data Pattern Write and Data Pattern Verify formatting phases.

CONFIG Verb

Syntax: CONFIG

CONFIG will display the system configuration. This verb 'walks' the bus and reports the channel address (in hex), the device-id (in hex), the firmware revision-level of each controller and the central processor (in hex), and a description of all devices. The terminal being used as a console will have an '*' displayed next to its device-id.

D-RDLB Verb

Syntax: D-RDLB (channel-#)

The D-RDLB verb allow the operator to read the label on a disk created by the FORMAT or D-WTLB commands. You can then verify the label name and other options selected when it was created. A breakdown of these options is listed in the section describing the FORMAT verb.

D-WTLB Verb

Syntax: D-WTLB (channel-#)

The D-WTLB verb allows the operator to change a disk label without having to go through a FORMAT. The questions asked are the same as in the FORMAT procedure.

DTEST Verb

Syntax: DTEST

This is the Disk Test Program command.

DTX Verb

Syntax: DTX number

The DTX verb converts a decimal number into its equivalent hexadecimal value and displays the hex number on the screen.

FORMAT Verb

Syntax: FORMAT chan# {chan# ...}

When disks are new or in an unknown state they must be "formatted" prior to being used. This formatting process puts address information on each track and checks for defective tracks.

When disk packs are manufactured it is impossible to insure that every portion of it is perfect. To compensate for this, all disks are "mapped" during the formatting process. This "mapping" locates a defective spot (defective track), flags the defective track and assigns an alternate track.

All this information is stored in a "Defective Space Table" that serves as a cross reference of defective tracks to alternate tracks that are used in place of the defective ones.

The FORMAT verb allows you to format from one to eight disks. Up to eight channels (chan#) can be specified in the verb. The disks running on those channels will be formatted as one "disk set".

1. If more than one (1) chan# was specified go to step 4. If only one channel # was specified, the formatter will respond by displaying the device type on the specified channel.

If the disk drive is not a Cartridge Module Drive (CMD), then go to step 2. If the disk drive is a CMD, a prompt is displayed:

    (F)ixed or (R)emovable ?

Enter 'F' or 'R' to format either the fixed or the removable cartridge.

2. Respond to the prompt:

    Format for (U)ltimate or (T)VOS/GCOS ?

Enter 'U' only; a 'T' response will display a warning.

A series of questions are now posed, depending on the type of disk drive and answers to previous questions. These questions will be used to build a disk label.

3. Respond to the prompt for set name by entering a name of up to 12 characters. This will identify this set of disks. All disks in any given set must have the same set name. If you are formatting a CMD removable cartridge, go to step 6.

4. Respond to the prompt:

    Enter # of disks in set:

with the total number of disks in the set. A set may not have more than 8 disks. If '1' is entered go to step 6.

5. Respond to the prompt for the sequence # by entering the sequence number of this pack within the set.(ie. '1' of 3 or '2' of 4, etc.) In any one set of 'n' packs, the disk packs must be labeled consecutively from '1 of n' to 'n of n'.

6. Respond to the prompt:

    Save old defective space table (<CR>=Y/N):

with 'Y' or <CR> to save alternate tracks that were marked defective the last time the disk was formatted. This ensures that any bad spots on disk will always be marked defective. An 'N' response cancels the previous markings.

7. The formatter will proceed through four phases: Format, Verify, Data Pattern write and Data Pattern verify. As each phase executes, an '*' is displayed when each set of 16 cylinders is completed. If any tracks are found to contain errors, these are reported on the terminal and marked as defective.

If a multiple format was selected (more than one chan#), go to step 9.

8. When the process completes, a message is displayed indicating the total number of errors encountered:

```
This pack contains n defective tracks.
Format complete
:>
```

If more than 15 bad tracks are found or if any track within the first 8 cylinders is bad the following message is issued:

```
This pack does not meet the minimum standards
for use with an Ultimate system
- it cannot be used.
```

Do not attempt to use this disk!

9. On a multidisk format (where more than one chan# is specified), the formatting of each drive will be done consecutively and automatically. Prior to the format of each disk, a label will be displayed indicating the sequence # and the channel # of the pack being formatted.

<u>HELP</u> <u>Verb</u>

Syntax: HELP

HELP will display on the terminal all verbs in the Diagnostic Monitor. If two wrong commands are entered at the prompt (:>), this command is automatically invoked.

<u>IO</u> <u>Verb</u>

Syntax: IO

The IO verb is designed for Ultimate internal testing use. It builds instructions to read or write to a peripheral. The operator is prompted for:

```
Channel:                 Device-id:
Read or Write:
Function:                Data word:
```

<u>RAT</u> <u>Verb</u>

Syntax: RAT (channel# of disk to verify)

The Read Alternate Track (RAT) verb reads the defective space table on the disk and displays it on the terminal. If the verb returns to a prompt with no display, then no defective tracks are recorded on the disk being tested. If defective tracks are present, each will be displayed as follows:

```
CYL xxx TRACK xx SECTOR xx
```

RD-DSK Verb

Syntax: RD-DSK (channel# of disk drive to verify)

This verb performs a non-destructive disk read on the specified device. If the device is a CMD, you must specify "Fixed" or "Removable". The RD-DSK verb is actually running the second phase (Verify) of the FORMAT verb.

NOTE: This verb runs only on "Ultimate" formatted disks.

It will lend considerable assistance to the customer and the Honeywell field engineer in diagnosing a "suspected" disk problem. The RD-DSK verb will allow you to safely read each address on the disk. If it runs clean then there is no reason to reformat. If it finds errors then the suspicions of a disk problem are verified. But, even then the pattern of disk error locations might point to a disk electronics problem which may be repairable without losing data.

If an error is encountered, the defective space table on the disk is scanned. Therefore, bad tracks found in the original formatting and assigned an alternate will not be displayed. The defective space table on disk can be read with the RAT verb.

T-ATT Verb

Syntax: T-ATT channel#

T-ATT attaches the tape unit on the specified channel number.

The addressed tape unit will be used for all verbs utilizing tape. If another drive is already attached, the execution of this verb will automatically detach that drive before attempting the attach.

T-DET Verb

Syntax: T-DET

T-DET detaches the tape unit.

T-FWD Verb

Syntax: T-FWD

T-FWD moves the tape forward one (1) file. The tape is spaced forward to the next tape-mark or the end of reel. The tape is left positioned immediately following the tape-mark. The tape unit must be attached.

T-REW Verb

Syntax: T-REW

T-REW rewinds the tape. The tape is left positioned at the beginning of tape marker (BOT). The tape unit must be attatched.

T-SIZE Verb

Syntax: T-SIZE {size}

T-SIZE sets or displays the maximum tape record size. The size may be in the range 256 to 16384 and specifies the maximum size of a tape record expressed in bytes. The actual size used may be smaller, and is determined by the individual verb's software. If the optional size parameter is omitted the current maximum size is displayed. The default size is 8192.

T-UNLOAD Verb

Syntax: T-UNLOAD

T-UNLOAD rewinds and unloads the tape. The tape on the attatched tape unit is first rewound and then unloaded.

XTD Verb

Syntax: XTD hex-number

XTD converts a hexadecimal number into its decimal equivalent and displays the decimal number on the screen.

2.11 USING THE UTILITIES MONITOR

The Binary-save Utility System Monitor (UTIL) is a free standing, memory resident program designed for off-line backup and maintenance. By using UTIL, the user can save or restore the entire system in a "binary" format to or from tapes.

The Binary-save Utility System Monitor (UTIL) is a free standing, memory resident program designed for off-line backup and maintenance. UTIL has two processors: Binary Save Processor and Binary Restore Processor.

The Binary Save/Restore Processors work on TAPE drives only. (Although the user can load UTIL on a Honeywell disk-only system, Binary Save/Restore will not work on the removable packs.)

The UTIL Binary Save Processor saves the data from disk to tape, "as is", onto tape, starting from frame 1 to the start of the last contiguous block of overflow. (All frames in between, regardless of whether data frames or overflow frames, are saved/restored. Hence, the actual number of frames saved/restored by UTIL is generally greater than that by the regular save/restore.)

The Binary Restore Processor does the reverse direction: It copies the data from tape to disk, starting from frame 1.

UTIL binary save/restore runs at the maximun speed of the TAPE drive, and, in general, is faster than regular save/restore. However, the actual speed depends on the speed and density of the tape drive and the number of frames to be saved/restored. (Note again that this number is, depending on fragmentation of overflow, generally greater than the number of frames saved/restored by the regular save/restore.)

NOTES OF CAUTION:

(1) Binary save should not be used to replace the regular file-save completely. One major difference between the binary save/restore and the regular file-save/restore is that binary save does NOT check for GFE's. Data will be saved/restored "AS IS". If you have any doubt as to the integrity of files, regular save/restore is recommended. It is further recommended that if binary saves are used as daily backups, regular file-saves must be used on a weekly or monthly basis.

(2) A binary restore MUST be attempted ONLY on a system with the same hardware configuration (that is, THE SAME NUMBER OF TERMINAL PORTS, AND THE SAME TYPE AND SIZE OF DISK.)

<u>Loading UTIL</u>

UTIL is loaded by booting the system with the SYSGEN tape or cartridge (or from disk). Before booting the system, be sure that all users are logged off, and then do a ":WARMSTOP" from the SYSPROG account. This ensures that all data in memory is flushed to disk. It is strongly recommended that a "VERIFY-SYSTEM" be done here prior to the :WARMSTOP or booting the system. (For details on the VERIFY-SYSTEM command, see the System Commands Guide.)

Depending on whether the system does or does not verify, the utility-warmstart or utility-coldstart (on Honeywell-based systems) or COLDSTART (on DEC-based systems) must be used. (See below.)

<u>Initiating UTIL from Honeywell-Based Systems</u>

Mount the SYSGEN tape and initialize the system. The system will give the following message on line 0:

```
This is the Ultimate Operating System

System Startup Options:
(B)oot
(C)oldstart
(D)iagnostic Monitor
(F)ile Restore
(O)ff-line Monitor
(U)tilities Monitor
(W)armstart

Enter Option(s) or ? for help:
```

Select U. The system automatically responds by the following message:

```
(W)armstart Util from disk or (C)oldstart from tape:
```

Now you can select either 'W' or 'C' depending on whether a utility-warmstart or a utility-coldstart is desired. Both utility-warmstart and utility-coldstart load UTIL into memory. The utility-coldstart additionally loads the ABS section prior to loading UTIL. The utility-warmstart will generally be used on a daily basis when doing a binary save. (In this case, UTIL is loaded into memory from disk. Nothing on disk is changed.) However, if there have been mismatches in the system, the utility-coldstart must be used. See below.

The utility-coldstart will first load the ABS section from the SYSGEN tape to disk, and then load UTIL into memory from disk. The utility-coldstart will be used in two instances:

(1) A binary restore is desired.
(2) If for any reason the ABS section has mismatches, the utility-coldstart must be used to insure that a good ABS is loaded onto disk.

Initiating UTIL from DEC-Based Systems

If booting from tape or disk, a System Startup message and options are displayed, similar to the message displayed on Honeywell-based systems, shown above.

Select U. UTIL is loaded into memory from tape or disk.

The Utility System Load-Completion Message

The completion of loading UTIL will be indicated by the following message on line 0:

```
This is the Ultimate Utility System yyyy/mm/dd
(Revision: x                        dd/mmm/yyyy)

u>
```

(The second line is on DEC-Based systems only; 'x' is the revision number of UTIL.) The date format (yyyy/mm/dd or dd/mmm/yyyy) indicates the revision date of UTIL. Line zero (0) will be the only active terminal line when running from UTIL.

Note: When performing a binary restore to a "virgin" disk, the disk must first be formatted by using the Ultimate Diagnostics Monitor before booting the Utilities Monitor. (See Section 2.10, THE DIAGNOSTICS MONITOR.)

Binary Save and Restore

The binary save and restore are done by the SAVE and RESTORE verbs in UTIL. See below further information on the verbs.

Loading the Ultimate Operating System

When binary save/restore is done, the system can be put back to normal operation by one of the following methods:

(1) Use the BOOT verb. See below for further information on the verb.

(2) Boot the system from a SYSGEN tape and select the W option.

(3) Boot the system from disk and select the W option.

UTIL Verbs

The UTIL verbs are:

BOOT	DISC-ERRORS (DEC-Based systems only)
HELP	RESTORE
SAVE	T-ATT
T-BCK	T-DET
T-ERASE	T-FWD
T-REW	T-UNLOAD
T-WEOF	WHO

NOTES ON USAGE OF UTIL VERBS:

1) For ease of use all UTIL verb syntax is the same as the system (TCL) level. However, UTIL is a stand-alone system. Do not confuse UTIL with the normal operating system. Only those verbs documented in the following sections are available in UTIL.

2) Use of BREAK Key for Honeywell-based systems:

> The use of the break key will force UTIL into the REMOTE PANEL processor. A 'G' or Linefeed will continue processing; a 'W' will end the current process and return to UTIL command level.

Use of BREAK key for DEC-based systems:

> Pressing the BREAK key enters the Panel Program. From the Panel Program, the operator can enter the following commands:

END terminates executing task and returns control to the Utility System command mode.

ODT halts CPU, putting it in ODT. To reenter the Panel Program from ODT, enter a 'P'.

<LF> The linefeed character resumes execution of the interrupted task.

3) If the user makes two or more mistakes (e.g., misspelling of verbs) consecutively, UTIL automatically does a "HELP" (see the HELP verb below).

4) For all UTIL verbs utilizing tapes, an implicit attachment to tape drive 0 will be attempted if an explicit attach was not previously done.

5) (For DEC-based systems only). The system has two prompts. The u> prompt indicates the Utility System Command mode; the p> prompt indicates the Panel Command mode.

BOOT verb

Syntax: BOOT

The BOOT verb initializes main memory and all controllers and boots the Ultimate Operating System from the first logical disk drive in the disk set.

(For Honeywell-based systems): The Options message is displayed. Select the W option to put the system back to Ultimate Operating System normal operation.

(For DEC-based systems): The message 'System Auto-Warmstart' is displayed. Be sure to set the time and date.

DISC-ERRORS verb (DEC-based Systems only)

Syntax: DISC-ERRORS

This verb will list any disk errors encountered during the Binary Save.

HELP verb

Syntax: HELP

The HELP verb lists all verbs in UTIL on the user's terminal. If the user makes two or more mistakes (e.g., misspelling of verbs) consecutively, UTIL automatically does a "HELP".

RESTORE verb

Syntax: RESTORE {L}

The 'L' option displays the tape label on the terminal.

The RESTORE verb invokes the UTIL Binary Restore Processor. A complete binary restore from the tape to the disk set is initiated. RESTORE makes the following checks prior to writing to disk:

1) Verify that the tape to be restored was created by the UTIL Binary Save Processor - if not then abort the restore process.

2) Compare the disk capacity of the saved system to the capacity of the current disk set - if not equal the UTIL processor will warn the user of the discrepancy and allow the user to continue the restore or quit.

RESTORE will now begin a physical copy from the tape to the disk set. However, if the capacity of the disk set is too small to restore the system, RESTORE will stop and display an error message when capacity of the disk set is reached.

SAVE verb

Syntax: SAVE {options}

The options are 'A' to save all data to MAXFID, and 'L' to display tape label on screen. On Honeywell systems only: 'E' estimates tape usage only, and does not proceed with Save.

SAVE initiates the UTIL Binary Save Processor. SAVE will copy all data from the current disk set to the tape drive.

On Honeywell-based systems only, the SAVE Processor will, if specified, estimate the number of feet of tape required for the save based on the number of FRAMES to be saved and the density of the tape drive.

SAVE first writes a tape label to uniquely identify the tape as created by UTIL then copies all data, starting from FRAME 1 to the start of the last contiguous block of overflow, unless the A option is specified.

The tape block size for UTIL save is fixed at:

DEC=based systems:	3600 bytes
Honeywell systems:	8208 bytes (8192 for data and 16 for control information)

Each tape record written has a serial number as part of the control information to insure restore integrity.

T-ATT verb

Syntax: T-ATT {drive#}

where drive# is a single-digit number. T-ATT attaches the tape drive whose number is specified. Default drive attachment is 0.

NOTE: In UTIL, drives are numbered in order of increasing channel number, which may be different from the drive number used in the normal Ultimate operating system.

T-BCK verb

Syntax: T-BCK

T-BCK backspaces the tape one (1) file.

T-DET verb

Syntax: T-DET

T-DET detaches the tape drive.

T-ERASE verb

Syntax: T-ERASE

T-ERASE erases approximately 2 inches of tape from the current position.

T-FWD verb

Syntax: T-FWD

T-FWD moves the tape forward one (1) file.

T-REW verb

Syntax: T-REW

T-REW rewinds the tape to load point.

T-UNLOAD

Syntax: T-UNLOAD

T-UNLOAD rewinds and unloads the tape.

T-WEOF

Syntax: T-WEOF

T-WEOF writes an EOF mark (file mark) on tape.

WHO

Syntax: WHO

WHO identifies the system with the following message:

    Utilities System

2.12 SHUTTING DOWN THE SYSTEM

An Ultimate system may be shut down via a warmstop operation (:WARMSTOP command).

The :WARMSTOP system (TCL) command flushes memory, then halts the system. It saves the current Process workspace data to allow warmstart restarts. It should be used for all scheduled system shutdowns.

After the command has been executed, the system displays a message similar to the following:

```
     Memory Flushed
     Halt= xxxxxx
            R0     R1      R2     R3     R4     R5     R6     R7
    regs: xxxxxx xxxxxx xxxxxx xxxxxx xxxxxx xxxxxx xxxxxx xxxxxx
   stack>  xxxxxx xxxxxx
  xxxxxx
  @
```

where the 6-digit number ('xxxxxx') represents the contents of the registers at the time of the warmstop. This information is displayed only in order to assist the System Manager if there is a problem in bringing the system back up.

At this point, you may turn off the power to the Ultimate system equipment. After the scheduled maintenance or other operations are completed, the warmstop allows you to boot the system and warmstart the system. (See Section 2.3, WARMSTARTING, for details on the WARMSTART procedure.)

NOTE: Bootstrapping options after a warmstop:

After the power is turned on, you must load a SYS-GEN tape to boot the system from tape.

Or, to you may boot the system from disk. (For details on booting, see Section 2.1, BOOTSTRAPPING THE SYSTEM.)

SECTION 3

PROCEDURES TO BACKUP AND RESTORE THE SYSTEM

SECTION 3

PROCEDURES TO BACKUP AND RESTORE THE SYSTEM (continued)

3.1 SUMMARY OF SYSTEM BACKUP AND RESTORE OPTIONS

An Ultimate system can be be saved (backed up) from the disk data base to magnetic tape. A variety of commands can be used to accomplish system backup, depending on what portion of the data base is to be saved. An Ultimate system can be restored to the disk data base from backup tapes. A variety of commands can be used to accomplish system or file restores, depending on what portion is to be restored.

It is the System Manager's responsibility to establish a policy for system backups and file saves. A standard date and time should be set for each procedure. This section (Section 3) gives guidelines and procedures for a variety of options regarding backups and restores.

Overview of System Backup Concepts

The major types of system backups are (1) saving the Ultimate system software, (2) saving all data (system and user files), (3) saving only updated data, (4) saving data in a specified account, and (5) saving all parts of the system (a SYS-GEN tape).

Figure A shows the system commands and menus that are used to backup the system, as well as the specific portions of the system they each save. Topics in this section explain the procedures; the System Commands Guide details each command.

On a SYS-GEN tape, remember that there are five sections:

1/2. SYSTEM BOOT 1/SYSTEM BOOT 2 (two separate files).
3. COLDSTART section (one file).
4. ABS section (one file).
5. FILES section (multiple system files, depending on the release). User files are added to this section as the system is used over a period of time.

A back-up operation, then, refers to any procedure that saves to tape or disk some or all of the above sections that comprise an Ultimate system.

Overview of System Restore Concepts

The major types of system restores are (1) restoring (reloading) the Ultimate system software, (2) restoring all data (system and user files), (3) restoring only updated data, (4) restoring a specified account or file, and (5) restoring all parts of the system.

Figure B shows the system commands and menus used to restore the system, as well as the specific portions of the system they each restore. Topics in this section explain the procedures; the System Commands Guide details each command.

A restore operation, then, refers to any procedure that loads from tape or disk some or all of the sections that comprise an Ultimate system.

Commands & Options	BOOT Section	COLDSTART Section	ABS Section	SYSTEM Files	USER Files
	<---------------SYS-GEN Tape Contents-------->				
CREATE-BOOT+	X	X	X		
SYS-GEN	X	X	X	X	X***
ABS-DUMP		X*	X*		
:DUMP-MODULE	X				
FILE-SAVE+ or SAVE				X	X
ACCOUNT-SAVE				X**	X**
UPDATE-SAVE+				X	X
Honeywell systems: DISK.COPY	X	X	X	X	X***

+ also available on SYSPROG Main Menu

* either/both files are optionally dumped to tape.

** depends on whether specified account is a system or user account.

*** if any

Figure A. System Commands for Backup

	<---------------SYS-GEN Tape Contents-------->				
Commands & Options	BOOT Section	COLDSTART Section	ABS Section	SYSTEM Files	USER Files
Front Panel: BOOT or Initialize	X				
Startup Options:					
(B)oot	X				
(C)oldstart		X	X		
(F)ile Restore		X	X	X*	X*
(W)armstart**					
:FILELOAD				X*	X*
:WARMSTART**					
ABS-LOAD		X	X		
COLDSTART** (proc)					
ACCOUNT-RESTORE				X+	X+
SEL-RESTORE				X++	X++

* file restore loads all files on the mounted tape(s), including update/transaction tapes, if specified.

** no files are loaded with a system warmstart, or by the COLDSTART proc. (However, from the System Startup options, the COLDSTART and ABS sections <u>are</u> loaded from the Boot or SYS-GEN tape used to boot the system.)

\+ depends on whether specified account is a system or user account.

\++ depends on whether specified file is in a system or user account.

Figure B. System Commands for Restores

3.2 STANDARDS AND GUIDELINES FOR SYSTEM BACKUP

Duplication of the SYS-GEN tape and Boot tape guard against loss or damage to the Ultimate-supplied SYS-GEN tape used to generate the system.

Standard file save and update save procedures guard against loss of data due to system failures or other errors.

System Backup Tapes and Disks: SYS-GEN and Boot Tapes

Always have at least one current release level SYS-GEN and Boot tape available for emergencies.

An Ultimate system SYS-GEN tape must be used to generate an Ultimate system from scratch. Users may create backup SYS-GEN tapes by using a special file-save proc, the SYS-GEN command. The SYS-GEN tape may then be used to recover from system failures, restore after system maintenance, or transfer data to another system. The contents of a SYS-GEN tape are detailed in Section 1.5, ULTIMATE SYSTEM SYS-GEN TAPES.

An Ultimate system Boot tape contains only the system programs necessary to boot the system and coldstart or warmstart the system. These are the same programs as on the first four files of a SYS-GEN tape. However, a Boot tape contains no system or user files. Users may create backup Boot tapes by selecting the Create Boot tape option from the SYSPROG Main Menu. Or, use the CREATE-BOOT system command.

For procedures, see Section 2.7, CREATING SYS-GEN OR BOOT TAPES, and Section 2.8, DUPLICATING SYS-GEN TAPES.

For disk-based systems, a set of backup disks should be copied. See Section 2.9, FORMATTING AND DUPLICATING DISKS.

ABS Tapes

The ABS section of an Ultimate disk subsystem contains the assembly language system software stored in frames 1-1023. This section of the disk data base can be backed up and restored via two system commands: ABS-DUMP and ABS-LOAD.

To create an ABS tape, first mount a tape and bring it to load (ready) point. Then attach the tape (T-ATT command). Now, from the system (TCL) level, use the ABS-DUMP command:

    ABS-DUMP

You are prompted for the ABS tape label:

    ABS tape label:

A label string may be specified, or <CR> only to create a null label. If no tape has been attached, an error message is displayed and the system returns to the TCL level.

You are then prompted for the ABS frame limits, which are entered in the format:

    ABS limits: Ax(-x}   or Cx(-x)

The 'A' must always precede the starting frame number when creating the ABS section. The 'C' must precede the starting frame when creating the Coldstart section of a Boot tape; however, this option should never be needed by a user. One or more single (x) or groups (x-x) of frames may be saved to tape. If multiple frame(s) are saved, each parameter is separated by a comma.

Each parameter is listed as the associated frame(s) are dumped to tape. When the dump is completed, the system returns to the TCL level.

To restore from an ABS tape, mount and attach the tape. Then use the ABS-LOAD command from the system (TCL) level:

    ABS-LOAD

Or, you can restore from the ABS tape during a coldstart when the system displays the message:

    Mount ABS tape and enter number of files to skip, if any

However, normally the ABS frames immediately follow the Coldstart section of a Boot tape.

File Save Tapes

It is vital that you protect your Ultimate system data base by adequate file backups. A file save tape contains only the FILES section of the disk data base; it contains no system software. Ultimate recommends, as a minimum:

1. Daily saves of files updated during that day. This is known as an "update save". Procedures are covered in Section 3.4.

2. Retention of the update save tapes for at least one week's time before re-using the tape.

3. Weekly full file saves, retained for at least one month's time before re-using the tape.

4. Monthly full file saves, retained for at least one year.

In summary, this requires using a separate tape (or tape set for full file saves) for each day of the week, one for each week of the month and one for each month of the year. The longer cycle tape-sets should be stored off premises to provide protection in the event of physical damage such as fire.

For procedures, see Section 3.3, FILE SAVE PROCEDURES, and Section 3.4, UPDATE SAVE PROCEDURES.

For disk-based systems, a set of backup disks should be copied. See Section 2.9, FORMATTING AND DUPLICATING DISKS.

In addition, for complete protection against lost data, the Ultimate system provides a transaction logging feature. When enabled, this feature logs to tape every update made to the disk data base, in real-time (when it occurs). For more information on transaction logging, see Section 3.5, TRANSACTION LOGGING PROCEDURES.

3.3 FILE SAVE PROCEDURES

The Ultimate system provides several ways of performing a complete file save of your entire disk data base. You may select the File Save option from the SYSPROG Main Menu, or use the FILE-SAVE system command (from the SYSPROG account), or log onto a special FILE-SAVE account.

Remember that a file save tape contains only the FILES section of the disk. It does not contain the system software.

File Saving from the SYSPROG Account

File saving should only be performed by responsible system personnel. Therefore, it is recommended that users execute file saves from the SYSPROG account only.

From the SYSPROG account, file saves may be performed by:

1. Selecting the File Save option from the Main Menu.
2. From TCL, entering a FILE-SAVE command.

File Saving from the FILE-SAVE Account

You may logon to a special FILE-SAVE account. The FILE-SAVE logon proc is a Q-pointer to the SYSPROG account's FILE-SAVE proc, and the File Save procedure is automatically invoked.

The File Save Procedure

The file save operation requires a tape (or removable CMD cartridge) to be mounted and ready to receive the files. If the tape is not ready, the system displays:

```
Tape unit off-line

Ensure that the File-Save tape is mounted with a Write
(enable) Ring, and the Tape drive is ON-LINE and at LOAD
point. Hit 'RETURN' to continue.
```

If a write-protected tape is mounted, the system displays:

```
No write-ring (C)ontinue/(Q)uit?
```

Either mount a tape with a write ring and enter 'C' to continue, or enter 'Q' to exit the command. A 'Q' reply causes the system to prompt:

```
Place Write (enable) Ring on File-Save tape and
try again.  Hit 'RETURN' to continue.
```

Once a write-enabled tape is loaded and ready, the File Save operation can begin. If the tape is still off-line when the user presses <CR> to continue, the system returns to the TCL level.

Figure A shows a sample of the messages generated by the File Save procedure. When it has finished it will rewind the tape and log itself OFF. Operator intervention is required only if the data to be saved exceeds one tape reel. File Saves create a complete backup of the FILES section in your disk data base (except accounts or files set up to not be saved - "DX" files, and those requiring a special parameter - V or W - in the SAVE command).

The File Save procedure normally creates a list on the terminal of the files it finds as it saves the data base. It will output error messages if it encounters unusual or illegal conditions but it will attempt to continue to save data. If the terminal you run the save on is not a hard copy terminal, you may want to send the listing to your printer.

If the following options are not specified in the FILE-SAVE command, the system prompts for them:

    Do you want the Console listing to go to the printer?

reply 'Y' to do so. To display the listing, reply 'N'. To exit FILE-SAVE, reply 'X'.

    Enter tape block size (500-8192) -

The File Save generates statistics about the saved data as a by-product. To receive a printed report, at the prompt:

    Do you want to generate File Statistics? (Y/N/X)

reply 'Y'. To ignore the report, reply 'N'. To exit FILE-SAVE, reply 'X'.

NOTE: The FILE-SAVE command is a proc that performs a full file save, using a form of the SAVE verb: SAVE(D,F,G,R,S,T). The SAVE and FILE-SAVE commands are detailed in the System Commands Guide.

In order to save accounts and files with the 'V' File Save option, or <u>not</u> save accounts and files with the 'W' File Save option, users must execute the appropriate SAVE command (or your own user-created PROC) at the TCL level.

Note that the 'R' option resets flags that have been set by Update Save procedures.

Update Saves are file saves of only the updated files in the data base. See Section 3.4, UPDATE SAVE PROCEDURES, for an explanation of these procedures.

3.3.1 FILE SAVING FOR MULTIPLE TAPE REELS OR DISKS

For systems with multiple tapes or removable disk cartridges, a series of prompts are associated with handling the additional tapes/disks.

Multiple Reel Tape Procedures

When a tape reel reaches the end-of-tape (EOT) marker, the tape is automatically rewound, and a prompt message is displayed on the terminal:

```
Mount reel # xx:
Label = name time date
```

where 'xx' is the two-digit hexadecimal reel number. 'Name' is the tape label. If unlabeled tapes are being used, the second line is not displayed. After the tape is mounted, loaded and ready, enter 'C' to continue the File Save.

All file save tapes are labeled. The reel number and tape label are checked for agreement with the number and label of the previous reel. If an error is detected, one of these messages will be displayed:

```
Incorrect Reel #
Incorrect Label
Reel #1 was labeled
```

If the reel # message is displayed, the new reel number does not match the requested reel number (or the first tape mounted is not reel #1). Mount the correct reel and enter 'C'.

If the second message is displayed, the first tape was unlabeled and the second was labeled. Mount the correct tape and enter 'C'.

If the third message is displayed, the first tape was labeled and the second was unlabeled. Mount the correct tape and enter 'C'.

Disk Cartridge Procedures

The disk cartridge procedures and messages are the same as the multiple reel tape procedures, above.

```
Tape attached  Block size: 4000

Rewinding...
 Block size: 4000

End of file
 Block size: 4000

Rewinding...
 Block size: 4000
File-Save beginning at 14:37:18  10 FEB 1986

Do you want the Console listing to go to the Printer? (Y/N/X)

Do you want to generate File Statistics? (Y/N/X)

Enter tape label -

Seq# of this data tape:   1      1     0     0
*     1     SYSTEM
1     2     SYSTEM-ERRORS
1     3       SYSTEM-ERRORS
1     4         SYSTEM-ERRORS
1     5     BLOCK-CONVERT
...
1     70    ACC

STAT-FILE being dumped to tape
70 items dumped

Rewinding...
File Save finished at 14:14:30  10 FEB  1986

File Statistics Report

To Lineprinter?  (Y/N/X)  Y

Detail Suppress ((Y/N/X)  N

Now generating statistics report

Count of system errors in past 2 days

To Lineprinter (Y/N/X) - Y

<<Connect time = 72 min;  CPU: 1389   lptr pages = 2   >>
<<Logged off at 14:44:59     on 10 FEB 1986            >>

10 FEB 1986    14:45:00   Logon please:
```

Figure A. Sample File Save Messages

3.4 UPDATE SAVE PROCEDURES

> The Ultimate system provides several ways of performing a file save on only the disk data base files that have been changed or updated since the last file save. You may select an update save option from the SYSPROG Main Menu, or use the system commands ALL-UPDATE-SAVE or PART-UPDATE-SAVE.

Remember that an update save tape contains only files that have been flagged as "updated" since the last file save.

The update save feature is designed to provide a fast, efficient way of protecting the disk data base and minimizing data loss in case of a power failure or system hardware or software problems. Instead of saving the complete data base with every file save (as described in Section 3.3), with update save you can shortcut the backup procedure and save only the files that have changed since the last full file save.

There are two main features in the update save concept:

1. Making an "incremental" file-save. With this facility, only those file groups that have been changed since the last file save are saved. The Incremental File-save is also called the "Update Save".

2. Logging all transactions (i.e., all updates to all files) to a secondary medium, magnetic tape. Transaction logging is explained in Section 3.5, TRANSACTION LOGGING PROCEDURES.

Two system commands perform procedures associated with update saving: ALL-UPDATE-SAVE, and PART-UPDATE-SAVE. These commands are procs that execute specific forms of the SAVE command. Two options, U and R, are added to the SAVE verb. The U option specifies that the SAVE command will save only data that has been changed since the last save. This is done at the group level of all files in the system.

Associated with each group of each disk file is a flag indicating whether any item in the group has been updated. With the U option specified, the SAVE command inspects these flags, and saves those groups that have the flags set. The R option is used to reset those flags after the groups are saved on tape.

The save tape obtained with the U option is called an update save tape. To restore the system, the latest "full" file-save tape is first restored, and then the update save tapes are loaded to update the files.

The update save feature and transaction logging can be used together to provide the greatest possible security for a data base. See Section "USING UPDATE SAVE AND TRANSACTION LOGGING TOGETHER".

NOTE: If the system is not already running on Rev121A or later, it must be upgraded before the update save commands can be used.

Follow the appropriate procedure in the UPGRADE PROCEDURE document to upgrade your system. The update save commands will not work unless the appropriate upgrade procedure is followed.

If the system is not upgraded properly, an attempt to use update save commands will cause the system to display an error message:

Configuration error; F-restore required

While the update save and transaction logging procedures provide system flexibility, users are cautioned to carefully read the sections pertaining to these features before using them. Failure to follow the appropriate procedures may result in losing all updates that have occurred since the last file save.

The Update Save Options Available with the SAVE Command

Two options, U and R, are used with the SAVE command to specify update save operations. With the U option specified, the SAVE command inspects the "group-updated" flags, and saves those groups that have the flags set.

The R option is used to reset each "group-updated" flag after the associated group is saved on tape. The R option can be used either with or without the U option to reset the group-updated flags.

The U and R options must be used in conjunction with other existing options to do the saves. For example, to do a full file-save and to reset the "group-updated" bits, the command "SAVE (D,F,G,R,S,T)" must be used.

In order to minimize operator errors, several forms of the SAVE command have been stored as procs so that different "save" operations can be easily requested. These "file save" procs must be executed from the SYSPROG account; they are:

FILE-SAVE (full file save)

SAVE command format: SAVE (D,F,G,R,S,T)

ALL-UPDATE-SAVE (complete update save)

SAVE command format: SAVE (D,F,G,T,U)

PART-UPDATE-SAVE (partial update save)

SAVE command format: SAVE (D,F,G,R,T,U)

SYS-GEN (complete SYS-GEN tape)

SAVE command format: SAVE (D,F,G,S,T,Z)

These procs provide two methods of using update saves. The main difference between the two methods is that the PART-UPDATE-SAVE resets the "group-updated" flags, while ALL-UPDATE-SAVE does not. This means that a "partial" update save tape contains only the files changed since the last update save or file save. However, an "all" update save tape contains all files changed since the last full file save.

The ALL-UPDATE-SAVE Procedure

The advantage of this method is that only the full file save tape and the last ALL-UPDATE-SAVE tape are needed to restore the data base. The disadvantage is that each ALL-UPDATE-SAVE procedure could require more time and tape since it saves (or re-saves) all groups changed since the last full File Save.

The suggested procedure would be:

1. A full file-save (via the FILE-SAVE proc) weekly.
2. All update save (via ALL-UPDATE-SAVE) daily.
3. Restore data base, as needed, with most recent File Save tape and most recent All Update Save tape.

The PART-UPDATE-SAVE Procedure

The advantage of this method is that is may save time and tape on each PART-UPDATE-SAVE procedure. The disadvantage is that the full file save tape and all subsequent PART-UPDATE-SAVE tapes are needed to restore the data base.

The suggested procedure would be:

1. A full file save (via the FILE-SAVE proc) weekly.
2. Partial update save (via ALL-UPDATE-SAVE) daily.
3. Restore data base, as needed, with most recent File Save tape and all Partial Update Save tapes.

NOTE: If, for any reason, a PART-UPDATE-SAVE is aborted, the next save must be a full file save, regardless of whether the proc used contains the SAVE command with 'U' option. If the 'U' option is present in the next save, the SAVE command will terminate prematurely with the following warning message:

[994] Full file save required

Because the last PART-UPDATE-SAVE was aborted, the update save tape is incomplete and is otherwise useless. Yet some (but not all) of the "group-updated" flags are already reset by the R option. There is no way for the next update save to tell which flags were reset by the last aborted PART-UPDATE-SAVE. In order to save the integrity of the system, all files must be saved by the next save procedure. ALL-UPDATE-SAVE does not have the same problem because the R option is not used in the proc.

Although it is possible to mix the ALL-UPDATE-SAVE and PART-UPDATE-SAVE together (e.g., do an ALL-UPDATE-SAVE on Monday and a PART-UPDATE-SAVE on Tuesday), it makes the restore procedure undesirably confusing and complex.

Ultimate recommends that the user choose a system back-up procedure, either the ALL-UPDATE-SAVE or the PART-UPDATE-SAVE proc, depending on the size, applications and other requirements of the system, and consistently use only that one procedure.

Update Save with Re-allocation Parameters

If there is a re-allocation parameter in the definition item (attribute 13 of the D pointer item) of a file, and if the re-allocation parameter has different values of modulo and/or separation from those of the file, the entire file is saved on an update save even though no items were changed. For example, assume that a file called AAA was created as 3,1 with a re-allocation parameter (5,1). The entire file AAA would be saved on an update save.

This is required because the items may be hashed into different groups at the time of a file restore. In order to handle item re-hashing correctly, the entire file must be saved on an update save. When loading the update save tape at the time of file restore, the program first clears the file and then re-hashes the items on the update save tape to the appropriate groups.

With this in mind, Ultimate recommends that you not add re-allocation parameters to files until you are ready to do a full file save followed by an immediate full file restore (to re-allocate the files). It is a waste of time and tape to save the entire file on multiple update saves.

If you want to change the re-allocation parameters and continue to do the update saves, never change the re-allocation parameters back to the current values of modulo and separation of the file. Otherwise, subsequent update saves will save only those groups that have the "group-updated" flags set. (Remember, the entire file is saved by the update save only if the re-allocation parameter has a different modulo and/or separation.) Such an update save tape does not preserve the integrity of the file and will cause loss of data at the time of full file restore. An example is given below to illustrate this problem:

PROBLEM: CHANGING REALLOCATION PARAMETERS

Date & Event	Mod sep	File AAA Reallocate parameter
MAR 1: FILE-SAVE	3,1	(5,1)
MAR 2: Change Realloc. Param.	3,1	(3,1)
MAR 2: PART-UPDATE-SAVE	3,1	(3,1)
MAR 3: Full File Restore; load		
FILE-SAVE tape	5,1	
PART-UPDATE-SAVE tape	3,1	

File AAA was created as 3,1. On the FILE-SAVE tape, file AAA had a re-allocation parameter (5,1). But, the parameter was changed back to (3,1) before the PART-UPDATE-SAVE. Only those groups in file AAA that had been changed since MAR 1 were saved on the PART-UPDATE-SAVE tape. At the time of the full file restore, the file was re-allocated to 5,1; all items were re-hashed and reallocated into the appropriate groups. However, when loading the PART-UPDATE-SAVE tape, only the updated groups are present. The items in these groups would be loaded assuming the original 3,1 mod/sep, destroying any items that had been reallocated into that group. Further, some items from the update tape, placed in the appropriate group for a 3,1 file, would not be in the appropriate group for a 5,1 file. If an item is in the wrong group for the current mod/sep, the system will not be able to find or retrieve it.

A "sequencing information segment" (see below) is written to each file save or update save tape. This sequencing information is to insure that, at the time of full-restore, the file save tape and the update save tapes are loaded in the correct order.

However, to create a SYS-GEN tape, a different sequencing information segment must be written on tape. The Z option is added to the SAVE command for this purpose. The Z option is to be used for creating a full file-save tape (e.g., a SYS-GEN tape) for the purpose of transfering data to another system. The Z option inhibits updating the system sequencing information, and produces a tape with all zeros in the sequencing information segment. The Z option indicates that this file save is not in the usual sequence of file saves for the particular system, and may be for use on another system. The SYS-GEN proc uses the Z option to create SYS-GEN tapes. The command used in the SYS-GEN proc is:

```
SAVE (D,F,G,S,T,Z)
```

Note that neither the U nor the R option is present.

<u>System Sequencing Information</u>

<u>System</u> <u>sequencing</u> <u>information</u> is kept in the system and is maintained by the operating system to keep track of the number of file saves, update saves and transaction logging

sessions. The system sequencing information is updated when a file save or an update save is done, or when the transaction logger is activated or restarted. (Transaction logging is discussed in Section 3.5.)

The SAVE command reads the system sequencing information, updates it according to whether the U and/or the R option is present (see below), and writes the updated version of sequencing information to the beginning of the file save or update save tape. This version of sequencing information is also displayed by the SAVE command at the beginning of the save. The display will be similar to the following:

```
Seq# of this data tape:     1      1     0      0
*    1SYSTEM
1    2 BLOCK-CONVERT
1    3 SYSTEM-ERRORS
1    4  SYSTEM-ERRORS
1    5 WP
...  ...
```

At the end of the file save or update save, the SAVE command writes the updated version of sequencing information back to the system. The updated version of sequencing information becomes the system sequencing information only at the end of a save procedure. The system sequencing information remains unchanged until a file save or update save is completed.

If, for any reason, the save is aborted, the tape being written to is incomplete and useless. Consequently, the SAVE command does not "count" the save; it does not write the updated version of sequencing information back to the system.

In a normal save, the sequencing information is updated by the SAVE command in such a way that at the time of a full file restore, the system can check the sequencing information on tapes to make sure that tapes are restored in the proper order. If, for example, at the time of full-restore, an attempt is made to restore an update save tape made before the full file-save tape, a warning message will be displayed, and the operator will be allowed to mount a different tape. (The operator is given the option to override the sequence check, if so intended.)

The system sequencing information is initialized at the beginning of a full file restore. The file restore software reads the sequencing information on the FILE-SAVE tape and writes it to the system as the system sequencing information.

The Four System Sequence Numbers on a Data Tape

The sequencing information consists of four (4) sequence numbers. The four sequence numbers are displayed as:

```
          Seq# of this data tape:   #     #    #    #

 Example: Seq# of this data tape:   1     1    0    0
```

The first number is the Full File Save sequence number; it is the total number of full file saves completed. The second number is the R (Reset flags) sequence number; it is the number of times the 'R' option was used in a save procedure. The third number is the Update Save sequence number; it is the number of update saves done since the last full file save. The fourth number is the Transaction sequence number; it is used by transaction logging (see Section 3.5).

These four numbers are kept in the system and are updated by the operating system at appropriate times. All four numbers are written to the beginning of file save, update save and transaction tapes.

At the time of full file restore, the sequence numbers from the FILE-SAVE tape are read into the system and become the system sequencing information. (Generally, a SYS-GEN tape has a sequencing information of all zeros.)

If, after the FILE-SAVE tape is loaded, the user wants to load update save tape(s), the system reads the sequencing information at the beginning of each update save tape and checks if the tapes are loaded in the proper order and displays warning messages if necessary.

The File Save Sequence Number

Every full file save (SAVE command without the 'U' option) increments this number by one (1). This number is never reset by the system.

The Reset Flags Sequence Number

Every file save that resets the "group-updated" flags (SAVE command with the 'R' option) increments this number by one (1). This number is never reset by the system.

The Update Save Sequence Number

Every update save (SAVE command with the 'U' option) increments this number by one (1). Every full file save (SAVE command without the 'U' option) resets this number to zero (0).

The Transaction Sequence Number

The transaction sequence number is explained in Section 3.5.

Figures A and B illustrate how these numbers are updated by the operating system.

Format of Sequencing Information on Tape

The four sequence numbers are written as a "sequencing information segment" in the first tape block at the beginning of file save, update save and transaction tapes.

An account save tape will have the same format of sequencing information segment written at the beginning. A flag will be set to signify that this is an account save tape, and all four sequence numbers will be zero in the segment. The segment is included on the account save tape for compatibility reasons and is otherwise ignored by the system.

The format of the sequencing information segment on tape is as follows:

```
V^x^s^f^r^u^t^_
```

where:

V = the letter V signifies the beginning of the sequencing information segment

x = (1 digit hex). Value is 0 or 1; 0 if this is a file save, update save or transaction tape, and 1 if this is an account-save tape.

s = (4 digit hex) system revision number (currently always 0001).

f = (4 digit hex) full file-save sequence number, or 0000 for account-save tapes.

r = (4 digit hex) R-option sequence number, or 0000 for account-save tapes.

u = (4 digit hex) update save sequence number, or 0000 for account-save tapes.

t = (4 digit hex) transaction sequence number, or 0000 for file save, update save and account-save tapes.

^ = attribute mark.

_ = segment mark.

The maximum value for 'f', 'r', 'u', and 't' is (hex) 7FFF, after which values will wrap around to 0001. The 's' value is used for internal purposes and is otherwise ignored.

EXAMPLE ONE: ALL-UPDATE-SAVE

1. Assume that a system has been upgraded from an older revision to the current revision. The system has been full restored via a SYS-GEN tape. All application accounts have been restored via ACCOUNT-RESTORE from the last file save. The appropriate steps in the UPGRADE PROCEDURE have been followed to complete the system upgrade.

2. The System Manager decides to use the ALL-UPDATE-SAVE method for daily system back-up, with weekly FILE-SAVES on Friday.

		SYSTEM SEQ# BEFORE SAVE f r u t	SEQ# ON SAVE TAPE f r u t	SYSTEM SEQ# AFTER SAVE f r u t
Fri a.m.:	system upgrade	0 0 0 0	N/A	N/A
Fri p.m.:	full FILE-SAVE	0 0 0 0	1 1 0 0	1 1 0 0
Mon p.m.:	ALL-UPDATE-SAVE	1 1 0 0	1 1 1 0	1 1 1 0
Tue p.m.:	ALL-UPDATE-SAVE	1 1 1 0	1 1 2 0	1 1 2 0
Wed p.m.:	ALL-UPDATE-SAVE aborted	1 1 2 0	1 1 3 0	1 1 2 0
Thu p.m.:	ALL-UPDATE-SAVE	1 1 2 0	1 1 3 0	1 1 3 0
Fri p.m.:	full FILE-SAVE	1 1 3 0	2 2 0 0	2 2 0 0
Mon p.m.:	ALL-UPDATE-SAVE	2 2 0 0	2 2 1 0	2 2 1 0
Tue p.m.:	ALL-UPDATE-SAVE	2 2 1 0	2 2 2 0	2 2 2 0

NOTE 1: The four numbers shown in each column are the 'f' (full FILE SAVE), 'r' (Reset option), 'u' (UPDATE SAVE), and 't' (transaction) sequence numbers.

NOTE 2: Since the sequencing information on the SYS-GEN tape is "0 0 0 0", the "system" sequencing information is initially "0 0 0 0" (see first day).

NOTE 3: Wednesday's aborted ALL-UPDATE-SAVE caused the system sequencing information not to be update. The update save tape made on that day was incomplete and could not be used.

NOTE 4: On both Fridays, full FILE-SAVES were completed. The 'f' and 'r' sequence numbers on the save tapes were incremented accordingly. (The FILE-SAVE proc uses a SAVE command with 'R' option.)

NOTE 5: The Update Save ('U') sequence number was reset to zero by the full file save on Friday (after a week of update saves).

If, for any reason, the System Manager needs to restore the system on the next day (Wednesday), he has to use the latest full file save tape (seq#=2 2 0 0) along with the latest update save tape (seq#=2 2 2 0).

Figure A. Sample ALL-UPDATE-SAVE Method

EXAMPLE TWO: PART-UPDATE-SAVE

1. Assume a similar situation as Figure A, but the System Manager decides to use the PART-UPDATE-SAVE method:

		SYSTEM SEQ# BEFORE SAVE f r u t	SEQ# ON SAVE TAPE f r u t	SYSTEM SEQ# AFTER SAVE f r u t
Fri a.m.:	system upgrade	0 0 0 0	N/A	N/A
Fri p.m.:	full FILE-SAVE	0 0 0 0	1 1 0 0	1 1 0 0
Mon p.m.:	PART-UPDATE-SAVE	1 1 0 0	1 2 1 0	1 2 1 0
Tue p.m.:	PART-UPDATE-SAVE	1 2 1 0	1 3 2 0	1 3 2 0
Wed p.m.:	PART-UPDATE-SAVE aborted	1 3 2 0	1 4 3 0	1 3 2 0
Thu p.m.:	full FILE-SAVE	1 3 2 0	2 4 0 0	2 4 0 0
Fri p.m.:	full FILE-SAVE	2 4 0 0	3 5 0 0	3 5 0 0
Mon p.m.:	PART-UPDATE-SAVE	3 5 0 0	3 6 1 0	3 6 1 0
Tue p.m.:	PART-UPDATE-SAVE	3 6 1 0	3 7 2 0	3 7 2 0

NOTE 1: On Wednesday, we had an aborted PART-UPDATE-SAVE. The system sequencing information was not updated on that day. The update save tape made on that day was incomplete (and useless).

NOTE 2: A full file-save was done on the next day (Thursday) even though the System Manager first attempted a PART-UPDATE-SAVE. Notice that the sequencing information was changed based on the actual full file-save.

NOTE 3: The Reset sequence number was incremented on every day (except Wednesday) due to the fact that the PART-UPDATE-SAVE proc has the R option. The system sequencing information was not updated on Wednesday because the PART-UPDATE-SAVE was aborted.

NOTE 4: The update save sequence number was reset to zero twice, on Thursday and Friday, by the full file-save.

If, for any reason, the System Manager needs to restore the system on the next day (Wednesday), he has to use the latest full file-save tape (seq#=3 5 0 0) along with the latest two PART-UPDATE-SAVE tapes (seq#=3 6 1 0 and seq#=3 7 2 0). The update save tapes must be loaded in the same sequence as they were made.

Figure B. Sample PART-UPDATE-SAVE Method

Restoring with Update Save Tapes

To full file-restore the system with update save tapes, the latest full file-save tape must first be restored. The following is a step-by-step procedure to restore the system with update save tapes.

1. Mount the SYS-GEN tape and initialize (i.e., boot) the system.

2. At the System Startup options, select 'F'. The system will load the "COLD LOAD" section and then will display the message "Mount ABS tape and press RETURN".

3. Press <CR> to load the ABS section. When the ABS section is loaded, the system will display the message "Mount DATA tape and press RETURN".

4. Now remove the SYS-GEN tape. Mount the latest full file-save tape and press <CR>. The system will read the tape label and the sequencing information segment on tape and will display them in the following format:

   ```
   L 0FA0 18:09:10  08 SEP 1983 DATA File-Save
   seq# of this data tape:      2     2     0     0
   Is this the right tape (Y/N)? |     |     |     |
                                 |     |     | Trans#
                                 |     |  Update
                                 |   Reset    seq#
                            File-save   seq#
                                seq#
   ```

   The numbers in the above example correspond to the f, r, u and t sequence numbers of the tape to be restored. Make sure the correct file save tape is mounted. Answer 'Y' to initiate the file restore.

5. After the full file-save tape is loaded, you will be asked if there are update save tapes to load:

   Update/transaction tapes (Y/N)?

   For ALL-UPDATE-SAVE: If you have been doing ALL-UPDATE-SAVE's, simply mount the latest update save tape and answer 'Y' to the above question.

   For PART-UPDATE-SAVE: If you have been doing PART-UPDATE-SAVE's, you have to load EVERY update save tape since the last full file-save in the same sequence as they were made. Mount the first update save tape made after the last full file-save and answer 'Y' to the above question. (After this tape is loaded, mount the NEXT update save tape and repeat this step.)

After a 'Y' answer has been given, the system reads the tape label and the sequencing information on tape and displays them in the following format:

```
          L 0FA0 16:00:10  12 SEP 1983 DATA Update Save
Seq# of this data tape:        2     2     2     0
Seq# of last data tape:        2     2     0     0
Is this the right tape (Y/N)? |     |     |     |
                              |     |     |  Transaction seq#
                              |     |  Update save seq#
                              |  Reset seq#
                          File-save seq#
```

The numbers in the above example correspond to the f, r, u and t sequence numbers of the last-restored and about-to-be-restored tapes. Again, the operator is given an opportunity to verify that the correct update save tape is mounted.

If an attempt is made to restore any tape in the wrong order, a message similar to the following will be displayed:

```
          L 0FA0 17:59:11  07 SEP 1983 DATA Update Save
Seq# of this data tape:        1     1     3     0
Seq# of last data tape:        2     2     0     0
Tape out of sequence; override (Y/N)?
```

It can be seen from the sequence numbers that an old update save tape has been loaded. Answering N will cause the system to ask "Update/transaction tape (Y/N)?" again. However, you may answer Y to override the sequence check if so intended.

6. After an update save tape has been loaded, the system will ask again:

   Update/transaction tape (Y/N)?

   Load the NEXT update save tape, if any, and answer 'Y'. Or, answer 'N' to complete the file restore.

3.5 TRANSACTION LOGGING PROCEDURES

The Transaction Logger is a utility that records disk file updates on magnetic tape as the updates are made. In the event of a system failure, the tape can be used in conjunction with a file save tape to restore all files to their state at the moment of failure.

There are two main features in the update save concept:

1. Making an "incremental" file-save. This type of file-save backs up only those file groups that have been changed since the last file-save. Incremental file-saves are also called "Update Saves".

2. Logging all transactions (that is, all updates to all files) to a secondary recoverable media (magnetic tape).

The system command called "LOG" performs procedures associated with transaction logging.

The update save feature and transaction logging can be used together to provide the greatest possible security for a data base. See Section 3.5.6, "USING UPDATE SAVE AND TRANSACTION LOGGING TOGETHER".

NOTE: If the system is not already running on Rev121A or later, it must be upgraded before the transaction logging command can be used.

Follow the appropriate procedure in the UPGRADE PROCEDURE document to upgrade your system. The transaction logging command will not work unless the appropriate upgrade procedure is followed.

If the system is not upgraded properly, an attempt to use the LOG command will cause the system to display an error message:

Configuration error; F-restore required

While the update save and transaction logging procedures provide system flexibility, users are cautioned to carefully read the sections pertaining to these features before using them. Failure to follow the appropriate procedures may result in losing all updates that have occurred since the last file save.

The advantage of using transaction logging is that up-to-the-minute system updates can be captured on tape. The disadvantage is that system performance could suffer in applications that have a high volume of system updates. Please refer to Section 3.5.4, "CONSIDERATIONS BEFORE USING TRANSACTION LOGGER".

What is a Transaction

As far as the transaction logger is concerned, a "transaction" is any one of the following updates: item update, item deletion, file creation, file deletion and file clearing for any disk file on the system.

The transaction logger records all transactions on tape. It runs as an independent Process, and is transparent to the user. No re-coding of application programs is required.

NOTE: Writing the same copy of an item back to the file is considered an "item update". For example, filing an item by the FI command of the Editor is considered an "item update" even if no change was made to the item.

What is Not a Transaction

There are a few types of disk updates which the transaction logger will not record: account-restores, async and bisync receive messages, and Spooler print jobs and hold files.

Account-restores are not recorded because of the amount of overhead required. (Essentially, the account has to be duplicated in the transaction disk queue, which means that twice as many frames are needed for account restore. See below for explanation on the transaction disk queue.)

Ultimate recommends that the user restore application accounts immediately before a full file-save, and activate the transaction logger after the file-save.

Async and bisync receive messages are not recorded because of the timing requirements of the async and bisync protocols. The timing overhead introduced by the transaction logger may cause too much delay for the protocols.

Spooler print jobs and hold files are not recorded because they are separate entities (they are not files or items, and cannot be saved by a file-save, T-DUMP or transaction logger).

WARNINGS

WARNING 1: DO NOT UPDATE ITEMS IN "DX" FILES

By definition, "DX" accounts and files are not to be saved in File Save procedures. The transaction logger, however, cannot distinguish distinguish a regular account or file from a "DX" account or file. All updates are treated as transactions. Updating any file in a "DX" account or updating a "DX" file will cause problems at full file restore time. (See Section "ERROR CONDITIONS WITH TRANSACTION TAPES".)

WARNING 2: DO NOT UPDATE A "DX" POINTER ITEM

Making an update to a "DX" pointer item while the transaction logger is turned on will cause the update to be logged as a transaction. However, the account or file is never saved in a File Save procedure. This will cause problems at full file restore time. (See Section "ERROR CONDITIONS WITH TRANSACTION TAPES".)

WARNING 3: ACCOUNT-RESTORES

Note again that account-restores are not recorded by the transaction logger.

3.5.1 TRANSACTION LOGGING AND THE LOG COMMAND

> The transaction logger is invoked by executing the 'LOG' command from TCL on the SYSPROG account. This causes the terminal to be dedicated to transaction logging. Until the transaction logger is exited, the terminal remains dedicated to transaction logging and cannot be used for other purposes.

Ultimate recommends that the transaction-logging terminal be any terminal line other than line zero (0). This is because line 0 is considered a "system console" and is used to boot the system (warmstart and coldstart) and for other system functions.

Only one terminal can be used for transaction logging. If logging has already been started on another terminal, an error message is displayed and control returns back to TCL.

The dedicated terminal is used to display menus, status information, and error messages, and to get operator input.

The Transaction Logger Main Menu

After the transaction logger is invoked, the main menu (see Figure A) is displayed on the terminal screen.

```
     Logger status:  Inactive

        Transaction Logger

     1. Activate logger; start tape
     2. Deactivate logger; exit menu
     3. Suspend tape
     4. Restart tape
     5. Change tape attachment parameters

        Enter option or <CR> to display status:
```

Figure A. Transaction Logger Main Menu

Press <CR> to display the status of the logger (shown above). The logger will be in one of three possible states:

Inactive
Active and started
Active and suspended

The Three Possible States of the Transaction Logger

Once the transaction logger is invoked on a terminal, the logger may be in one of the three possible states:

1. Inactive - The logger is invoked on a terminal but is otherwise doing nothing.

2. Active and started - the logger is Active and is logging transactions. Also, the tape operation is started and transactions are being written to the tape.

3. Active and suspended - the logger is active and is logging transactions. However, the tape operation is suspended. Suspending tape allows the tape drive to be used temporarily for other purposes by another port (e.g., T-DUMP or T-LOAD files).

   During this time, transactions are recorded on disk. When tape operation is restarted for the transaction logger, these transactions are written out to tape.

   Tape should not be suspended for any great length of time, since:

   (1) if a failure occurs, the transactions queued up on disk but not written to tape cannot be recovered;

   (2) overflow frames are used to store the transactions and if already low on disk space, the system may slow down and/or run out of disk space.

How the Transaction Logger Works

When an operator presses a key at the terminal in response to the 'Enter option:' prompt, the transaction logger menu handler validates the input (a number from 1 to 5), performs the requested operation if possible, prints a response, and re-displays the current logger status and the main menu.

Whenever the logger is active, logging takes place in the background while the menu handler is waiting for input. However, since the transaction logging and the menu handler share the same workspace (of the dedicated terminal), the transaction logging can not take place when the menu handler is busy taking options from the operator and/or displaying status and menu. Hence, use the terminal sparingly.

A Transaction Logging Session

In order to allow the user to do FILE-SAVE's, T-DUMP's, or any other tape operations, the transaction logger can be suspended or deactivated to detach the tape drive from the logger.

A "transaction logging session" starts when the logger is activated or restarted and ends when the the logger is deactivated or the tape operation is suspended.

During a transaction logging session, more than one reel of tape may be needed to log all transactions. The system will display a message "Mount reel #n" to ask the operator to mount a new reel of tape. All reels of tape made in the same session are collectively called a "transaction session tape set".

Each new transaction logging session starts a new "transaction session tape set". A new reel of tape should be mounted before a new session starts. In other words, before the transaction logger is to be activated or restarted, the operator should mount a new reel of tape and the tape must be at the BOT (Beginning Of Tape) mark. The transaction logger will write a "sequencing information segment" to the beginning of the tape to signify the beginning of a new transaction logging session.

When the user deactivates the logger or suspends the tape operation, the transaction logger writes a tape End Of File (EOF) mark to tape to signify the end of the transaction logging session.

Choosing Transaction Logger Menu Options

The main menu options are shown below:

```
        Transaction Logger

    1. Activate logger; start tape
    2. Deactivate logger; exit menu
    3. Suspend tape
    4. Restart tape
    5. Change tape attachment parameters

        Enter option or <CR> to display status:
```

Option 1: This option activates the logger and starts the tape operation. A transaction tape must be already mounted on the tape drive and loaded at the BOT mark. This option starts a new transaction logging session. The logger writes a "sequencing information segment" to the beginning of the tape to signify the beginning of a new session.

If the tape drive is off line, an error message is displayed and the transaction logger remains Inactive.

Option 1 can be selected only if the logger is inactive. Otherwise, the logger displays a warning message and remains in the original state.

Option 2: If this option is selected when the logger is in the Active and tape Started state, transactions queued up on disk, if any, are flushed onto tape; a tape EOF mark is written on tape to signify the end of the transaction logging session; frames used by the logger are released to the system overflow table. The logger is exited and control is returned to TCL.

If Option 2 is selected when the logger is in the Inactive state, control is returned to TCL immediately.

If Option 2 is selected when the logger is in the Active and tape Suspended state, a warning message is displayed:

> Any transactions now queued on disc will be lost.
> Continue (Y/N)?

Answering N will cause the logger to return to the Tape Suspended state. The answer Y will cause the logger to exit without flushing the queue. But frames used by the logger are returned to the system overflow table.

The transaction logger can be activated, exited, and then re-activated any number of times. Each activation starts a new transaction logging session, and each deactivation ends the current transaction logging session. During the period when the logger is inactive, system updates are not logged as transactions.

Before the logger is activated, a new reel of tape should be mounted. (In other words, each transaction logging session should start with a new reel of tape, loaded at the BOT mark.)

Any number of reels of tape may be needed for a transaction logging session, depending on the number of transactions, the sizes of transactions and the length of transaction logging session. When the end of the reel is reached, the system will display the message:

> Mount reel #n
> Label: TRANSACTION LOG
> (C)ontinue/(Q)uit

The operator must then mount a new reel of tape, as soon as possible. The transaction logger, at this time, is active but is waiting for the tape operation to continue. Transactions are recorded on disk and overflow frames are used if needed. If a system failure occurs before a second reel of tape can be mounted, transactions queued up on disk are lost.

Option 3: This option puts the transaction logger into the Active and Suspended state and ends the current transaction logging session. Transactions in the queue are flushed onto tape and the tape is terminated by a tape End Of File (EOF) mark. The tape operation is suspended. The tape drive is detached from the transaction logger, the current transaction tape should be removed from the tape drive and the tape drive can be used by another line for other purposes (e.g., T-DUMP, T-LOAD or FILE-SAVE). During the time the tape is suspended, transactions are recorded on disk. When the tape drive is ready again for transaction logging, the operator should mount a new reel of transaction tape and select Option 4.

Option 3 can be selected only if the logger is already in the Active and tape Started state. Otherwise, the logger displays a warning message and remains in the original state.

Option 4: This option is used to restart the tape operation. This option also starts a new transaction logging session. A new reel of tape should be mounted before this option is selected. The tape operation is restarted by first writing a sequencing information segment to the beginning of the tape and then dumping all transactions queued up on disk onto tape.

Option 4 can be selected only if the logger is already in the Active and tape Suspended state. Otherwise, the logger displays a warning message and remains in the original state.

Option 5: This option is used to change the T-ATT parameters (block size and tape drive number.) This option can be selected only if the logger is in the Inactive or the Active but tape Suspended state. When this option is selected, the operator will be prompted for T-ATT options. The operator should then enter the options (block size and/or tape drive number). If a carriage return alone is entered, no parameter is changed.

The logger automatically uses the standard default options (Tape drive 0 and block size 4000) if the operator does not specify any T-ATT options.

The Relationship Between the Logger States

The following diagram depicts the relationship among the three possible states.

```
                       --------------
    go to TCL<---------|  INACTIVE  |
     ^        Option 2 |            |
     |                 --------------
     |                       |
     |         ---------------
     |         | Option 1
     |         v
     |  ----------------           ------------------
     |  |              |Option 3  |                |
     |  | ACTIVE AND   |----------->| ACTIVE AND     |
     ---| TAPE STARTED |<-----------| TAPE SUSPENDED |
Option 2|              |  Option 4|                |
        ----------------           ------------------
```

Note that there are two situations where a new reel of tape must be mounted for the transaction logger:

1. When a new transaction logging session starts;

2. When the end of reel of a tape is reached, and the system asks for "Mount reel #n".

These two situations are very different:

1. When a new session starts, the reel of tape mounted is reel #1 of the transaction session tape set. This tape has a "sequencing information segment" at the beginning of the tape. There are two reasons to start a new session: The logger was either (a) inactive, or (b) active but tape suspended. Now the operator wants to either (a) activate it by using Option 1, or (b) restart it by using Option 4.

2. When the end of reel of a tape is reached, the next reel of tape is not the first reel of the session and does not have a sequencing information segment. Tape Reel #1, Reel #2, etc, all belong to the same transaction session tape set.

When the system has to be full-restored, transactions must be loaded in the same sequence as they were logged. Hence, the operator must:

1. Keep all reels of tape made in the same session in the same sequence as they were made. (That is, reel #1, reel #2, reel #3, etc.) These tapes constitute a "transaction session tape set" because they were made in the same session.

2. Keep all transaction session tape sets in the same sequence as they were made (Set 1, Set 2, etc.).

(For more information see Section 3.5.3, "KEEPING TRACK OF FILE-SAVE AND TRANSACTION TAPES".)

3.5.2 TRANSACTION LOGGING: DISK QUEUES AND STATUS DISPLAY

> Transactions are first recorded in the "transaction disk queue" and are written to tape whenever the amount of data fills a tape block. After data is written to tape, it is removed from the transaction disk queue.

It is possible that the first part of a transaction has been written to tape but the remainder is still on the transaction disk queue, as in the case where a transaction is slightly longer than a tape block.

If transactions are logged faster than the speed of the tape drive or if the tape operation is suspended, the transactions are saved in the transaction disk queue until they can be written to tape.

The transaction disk queue consists of two parts: a permanent queue and a variable-size overflow queue. When the LOG command is performed on a terminal line, the workspace of the terminal line is initialized to become the permanent queue. The permanent queue has 128 frames.

Transactions are first saved in the permanent queue. When the permanent queue is filled up, more frames are obtained from the system overflow table and appended to the overflow queue. Transactions are then saved in the overflow queue. Frames in the overflow queue are returned to the system overflow table after data in those frames is written to tape.

The size of the permanent queue remains unchanged but the size of the overflow queue may vary from time to time.

<u>Transaction Logger Status Display</u>

A typical logger status message, which is always printed before the main menu, is illustrated in Figure A.

The first line of the status display shows the status of the logger. The logger can be in one of the three possible states: Inactive, Tape Started and Tape Suspended. see Section 3.5.1 for information about the three possible states of the transaction logger.

The Seq# display shows the sequencing information of the current transaction tape. See the heading "The sequencing information for transaction tapes", below.

Also displayed is the number of frames currently in use for the permanent queue and the overflow queue.

Each transaction is given a transaction number and is marked with the time and date of entry into the disk queue. (The transaction number is explained below.) The "Latest transaction# in disk queue" displayed in the logger status shows the transaction number, and the time and date of the latest transaction currently saved in the disk queue. The "Oldest transaction# in disk queue" shows the transaction number, and the time and date of the oldest complete transaction currently saved in the disk queue. Question marks signify that there is no oldest complete transaction in the queue. (In the above example, only a remainder of transaction #1 is found in the queue. Part of transaction #1 is already written on tape.)

The transaction logger keeps track of the number of transactions in a transaction logging session. This number is incremented by 1 every time a new transaction is logged and is assigned to the transaction as the "transaction number". This number is reset to zero when a transaction logging session ends.

```
Logger status: Tape Started; Seq#:      1     1     0     1

Logger assigned to port 2
Disk queue permanent frames used:   1
Disk queue overflow frames used:    0
Latest transaction# in disk queue:  1 11:28:51  26 AUG 1983
Oldest transaction# in disk queue:  ?        ?            ?
```

Figure A. Transaction Logger Status Display

Sequencing Information for Transaction Tapes

The fourth number in the system sequencing information (the Transaction Sequence Number) keeps track of the number of transaction logging sessions since the last file save or update save. This sequence number is incremented by one (1) every time the transaction logger is activated or restarted. (Remember that option 1 on the Transaction Logger menu "activates", while option 4 "restarts" transaction logging.)

Do not confuse "Transaction Sequence Number" with "transaction number". The former is the total number of sessions and the latter is the total number of transactions within a session.

A sequencing information segment is written to the beginning of each transaction tape. At the time of full-restore, the system checks the sequencing information to make sure that tapes are restored in the proper order. If, for example, when restoring a system, an attempt is made to restore a transaction tape made before the full file-save tape, an error message will be printed, and the operator will be allowed to mount a different tape (or to override the sequence check if so intended.)

The System Manager is responsible for keeping the file-save and transaction tapes in the right sequence, so that, after a system failure, the system can be fully restored with minimum loss of data.

Keeping track of file save and transaction tapes

Use the following guidelines for file-save and transaction tape maintenance:

(1) The latest full file-save tape should always be kept ready. This full file-save tape is the "reference" file-save tape for all subsequent transaction tapes. If the system has to be restored, this tape must be used.

(2) Transaction tapes are associated with their "reference" file-save tape. After a file-save is completed, the file-save tape is the reference tape of all subsequent transaction tapes. (Transaction tapes made prior to the file-save are associated with the previous file-save tape.) Transaction tapes should be kept in sequence as they are made. At the time of full-restore, transaction tapes associated with the latest file-save tape should be loaded in the same sequence as they were made.

For example, if you have the following tapes made:

1. File Save tape, created at 17:00 MAR 1.
2. Transaction tape (reel 1 of transaction session 1) started when Logger activated at 19:00 MAR 1.
3. Transaction tape (reel 2 of transaction session 1) started after end of reel reached at 10:00 MAR 2, and new reel mounted.
4. At 13:00 MAR 2, the transaction logging session is suspended so that T-DUMPs may be done.
5. Transaction tape (reel 1 of transaction session 2) started when Logger re-started at 14:00 MAR 2

6. Then .... (system crashed) ....

When restoring the system, the tapes should be loaded in the following sequence:

1. Load the File-save tape made at 17:00 MAR 1
2. Load reel 1 of session 1 started at 19:00 MAR 1
3. Load reel 2 of session 1 started at 10:00 MAR 2
4. Load reel 1 of session 2 started at 14:00 MAR 2

The procedure to restore the system with transaction tapes is given in Section 3.5.5, "RESTORING WITH TRANSACTION TAPES".

3.5.3 TRANSACTION TAPES: SYSTEM SEQUENCE NUMBERS

The system sequencing information consists of four (4) sequence numbers. All four numbers are written to the beginning of file-save, update save and transaction tapes.

The File Restore software checks the sequencing information to see if the tapes are loaded in the proper order. An error message is displayed if the sequence is out of order. (However, the operator is given an option to override the sequence check.)

The first three sequence numbers (Full file-save, Reset option, and Update save sequence numbers) are explained in Section 3.4, UPDATE SAVE PROCEDURES, and are not covered in this section. The fourth sequence number is explained below.

Transaction Sequence Number

The transaction sequence number is the total number of transaction logging sessions since the last file save or update save. Whenever the transaction logger is activated or restarted, the transaction sequence number is incremented by one. The transaction sequence number is reset to zero by the SAVE command if the save is either a full file-save or an update save.

Format of Sequencing Information on Tape

The transaction logger writes a sequencing information segment on a transaction tape at the beginning of a transaction logging session. The segment is similar to that written on file save or update save tapes. But, a second set of the four sequence numbers may also be appended to the segment. The second set of sequence numbers, if present, is called the continuation sequencing information.

```
     V^x^s^f^r^u^t^*^f^r^u^t^_
                  |        |
                  ----------
                       |
          continuation sequencing information
```

where:

V = signifies the beginning of sequencing information

x = (1 digit hex number) "0" if a file-save, update-save, or transaction tape; or "1" if an account-save tape.

s = (4 digit hex number) system revision number (currently always 0001).

f = (4 digit hex number) full file-save sequence number, or 0000 for account-save tapes.

r = (4 digit hex number) R-option sequence number, or 0000 for account-save tapes.

u = (4 digit hex number) update-save sequence number, or 0000 for account-save tapes.

t = (4 digit hex number) transaction sequence number, or 0000 for file-save and account-save tapes.

* = asterisk; if present, is used to separate the two sets of sequence numbers.

^ = attribute mark.

_ = segment mark.

At the end of a full file-save, the file-save sequence number (f) is incremented by one (1) and the transaction sequence number (t) is reset to zero. This signifies that a full file-save has completed and this file-save tape is the "reference" file-save tape of all subsequent transaction tapes.

When the transaction logger is deactivated or suspended, the logger terminates the current transaction tape by writing an End of File (EOF) mark to the tape. This signifies the end of a transaction logging session.

From the Transaction Logger menu, the Logger may be:

1. Re-activated after deactivation by using Option 1.

2. Restarted after suspension by using Option 4.

Both actions start a new transaction logging session. However, there is a difference between the two:

1. After the Logger is deactivated (i.e., exited) by Option 2 of the Transaction Logger menu, updates are no longer recorded as transactions. When it is re-activated, some updates to the files may have taken place but are not logged. The new (re-activated) transaction logging session is not a "logical continuation" of the last session because there is a "gap" between the two.

2. After the Logger is suspended by Option 3 of the Transaction Logger menu, the Logger is still active and transactions are still recorded on the disk queue. When the tape is restarted, transactions recorded on disk during the period of suspension are dumped onto tape. The new (restarted) transaction logging session is a logical continuation of the last session because there is no "gap" between the two.

The <u>continuation</u> sequencing information is used to mark the transaction tapes in such a way that the above two situations can be distinguished:

When the logger is re-activated or restarted, the logger increments the transaction sequence number by one (1), and then writes the sequencing information segment to the beginning of the new transaction tape. The segment always contains the sequencing information of the <u>current</u> transaction logging session.

NOTE: In the case of re-activation, <u>no</u> continuation sequencing information is written in the sequencing information segment (i.e., the asterisk and the last four numbers are not included in the sequencing information segment).

In the case of restart, continuation sequencing information is included in the sequencing information segment. The continuation sequencing information is the sequencing information of the <u>last</u> transaction logging session.

Figure A illustrates an example of how the sequencing information is updated for transaction logging.

EXAMPLE OF SEQUENCING INFORMATION IN TRANSACTION TAPES

1. Assume that a System Manager has just done the upgrade of his operating system from an older revision to the current revision. A full file-save was done on a Friday afternoon.

2. The System Manager activated the Transaction Logger on Monday morning.

		SEQ# ON SAVE TAPE f r u t	SEQ# ON TRANS. TAPE f r u t	*	f r u t
Fri p.m.:	full FILE-SAVE	1 1 0 0			
Mon a.m.:	Activate Logger		1 1 0 1		
Tue 1pm:	Suspend Logger to do T-DUMP				
Tue 2pm:	Restart Logger		1 1 0 2	*	1 1 0 1
Wed 6pm:	Suspend Logger to do File Save				
Wed 6pm:	full FILE-SAVE	2 2 0 0			
Wed 9pm:	Restart Logger after save		2 2 0 1	*	1 1 0 2
Thu 1pm:	Suspend Logger to do T-DUMP				
Thu 2pm:	Restart Logger		2 2 0 2	*	2 2 0 1
Fri a.m.:	Deactivate Logger				
Fri p.m.:	Reactivate logger		2 2 0 3		

NOTE 1: The four numbers shown in each column are the f, r, u and t sequence numbers (respectively, full file-save, R-option, Update save and Transaction sequence numbers). Note that the sequencing information segment on a transaction tape may contain two sets of numbers, separated by an asterisk. The first set of number is the sequencing information of the current transaction logging session. The second set, if present, is the continuation sequencing information.

NOTE 2: To simplify the above table, end of reel condition and "Mount reel #n" messages are not included in the table. It must be understood that each session may create more than one reel of tape.

NOTE 3: The transaction sequence number was reset to zero by the FILE-SAVE on Wednesday evening.

...

Figure A. Example of Sequencing Information on Trans. Tapes

Notes on Sequencing Information (continued)

NOTE 4: The first two transaction tape sets (seq#=1 1 0 1 and 1 1 0 2) are associated with the first (Friday evening) FILE-SAVE tape (seq#=1 1 0 0). The last three transaction tape sets (seq#=2 2 0 1, 2 2 0 2 and 2 2 0 3) are associated with the Wednesday evening FILE-SAVE tape (seq#=2 2 0 0).

NOTE 5: All transaction tape sets, except the last one (2 2 0 3), are logical continuations of the previous tapes.

NOTE 6: On the second Friday, the logger was deactivated between 10am and 1pm. When the logger was re-activated at 1pm, the transaction tape set (seq#=2 2 0 3) did not have the continuation sequencing information. This signifies that some transactions were not recorded by the transaction logger during this "deactivation period". If, for example, a user created a file during the deactivation period, this transaction was not logged and data may be lost. For more information on data loss, see also Section "ERROR CONDITIONS WITH TRANSACTION TAPES".

NOTE 7: Transaction tape set 2 2 0 1 is associated with FILE-SAVE tape 2 2 0 0. At the same time, this tape is also a logical continuation of 1 1 0 2. This means that the user could full-restore the system with any one of the following two restore sequences:

normal restore sequence		"long" restore sequence	
Load FILE-SAVE tape	2 2 0 0	Load FILE-SAVE tape	1 1 0 0
Load Trans. tape	2 2 0 1	Load Trans. tape	1 1 0 1
Load Trans. tape	2 2 0 2	Load Trans. tape	1 1 0 2
Load Trans. tape	2 2 0 3	Load Trans. tape	2 2 0 1
		Load Trans. tape	2 2 0 2
		Load Trans. tape	2 2 0 3

Figure A (cont.) Transaction Tape Sequencing Information

3.5.4 CONSIDERATIONS BEFORE USING THE TRANSACTION LOGGER

The System Manager needs to consider the goals and requirements of transaction logging and verify that a current operating system revision is installed before using the Transaction Logger software.

File Save and Transaction Logging

Transaction logging provides the capability to keep all changes to the system up to the last minute. However, this facility is not a real substitute for the "full" file-save. We suggest that the user do the "full" file-save on a regular basis, possibly once a week during the week-end.

Required System Upgrade

If the system is NOT already running on REV121A (or later), the user must upgrade the system properly before using the transaction logger.

FOLLOW THE APPROPRIATE PROCEDURE IN THE UPGRADE DOCUMENT TO UPGRADE YOUR SYSTEM. The transaction logger will not work properly if the appropriate upgrade procedure is not followed.

Use of File-save tapes

It is important that the latest full file-save tape be available before invoking the transaction logger. The file-save tape is the "reference" against which transactions are recorded on the transaction tapes.

Tape Requirement

Any number of reels of tape may be required during a transaction logging session. The operator must check the tape drive frequently enough to insure that a new reel of tape is mounted as needed.

The tape block size and the density of tape (800 bpi, 1600 bpi, etc.) affects the amount of data that can be written to tape. Estimates on the sizes of different types of transactions are given in a table below. The number of transactions over a period of time depend largely on the size of the system, the number of active users on the system, and the type of applications. It is suggested that the user run the transaction logger for some length of "experimental" period before actual usage of this facility.

Overhead to the system

The transaction logger may impose considerable overhead to the system. The System Manager must consider that, with the logger active, a system update now generates approximately twice as many disk write operations. This is because the file, as well as the transaction disk queue, needs to be updated. The amount of overhead on a particular system, again, depends on the size of the system, the number of active users on the system, and the type of applications, and must be determined by experiments.

Remember that the Update Save may be a good alternative to transaction logging. In most cases, an update save is faster and requires fewer tapes than a full file-save. The user may consider having a system back-up procedure with more frequent update saves (several times a day), along with a full file-save every week. The protection provided by such a back-up procedure may be adequate for most applications and it would not impose much overhead to the system.

Deactivation vs. Suspension

As has been discussed earlier, the Logger can be either deactivated or suspended. Deactivation causes possible loss of data. On the other hand, suspension of the transaction tape requires that disk space be sufficient for the transaction disk queue.

When the tape drive is needed for other purposes for a long period of time, for example, FILE-SAVE, the System Manager must determine, beforehand, whether the logger should be deactivated or suspended. The amount of disk space needed for the transaction disk queue during a suspension period depends on the length of the period, the number of transactions, and the size of each transaction. The following table gives some rough estimates of the size of different transaction types:

1. item update: 50 bytes + length of item id + size of item

2. item delete: 50 bytes + length of item id

3. create file: for each section (DICT or DATA), 70 bytes + length of file name. (For example, "CREATE-FILE ABC 1,1 1,1" generates two transactions.)

4. delete file: for each section, two transactions are generated. Clear-file returns "overflow" frames of all groups to the Available Space pool. Delete-file returns "primary" frames of all groups to the Available Space pool. In addition, one transaction is generated to delete the "D" pointer item from the Master Dictionary. The first two transactions are about 50 bytes. The last one is about 50 bytes + length of item id.

   (For example, DELETE-FILE ABC generates five transactions: clear-file DATA ABC, delete-file DATA ABC, clear-file DICT ABC, delete-file DICT ABC, delete-item ABC in MD.)

5. clear file: 70 bytes.

3.5.5 RESTORING WITH TRANSACTION TAPES

> To full-restore the system with transaction tapes, the latest full file-save tape must first be restored. Then the system can be brought up-to-date with the transaction tapes.

The following is a step-by-step procedure to restore the system with transaction tapes.

1. Mount the SYS-GEN tape and initialize (i.e., boot) the system.

2. At the System Startup options, select 'F'. The system will load the "COLD LOAD" section and then will display the message "Mount ABS tape enter number of files to skip, if any".

3. Press <CR> to load the ABS section. When the ABS section is loaded, the system will display the message "Mount DATA tape and press RETURN".

4. Now remove the SYS-GEN tape, mount the latest full file-save tape and press <CR>. The system will read the tape label and the sequencing information segment on tape and display them in the following format:

```
L 01F4 18:01:10  06 SEP 1983 DATA File-Save
Seq# of this data tape:       2     2     0     0
Is this the right tape (Y/N)? |     |     |     |
                              |     |     |  Trans.seq#
                              |     |   Update save
                              |   Reset    seq#
                        File-save   seq#
                         seq.#
```

   The numbers in the above example correspond to the f, r, u and t sequence numbers of the tape to be restored. You are given the opportunity to make sure that the correct file save tape is mounted. Answering Y will initiate the full-restore program.

5. After the full file-save tape is loaded, you will be asked if there are transaction tapes to load by the following message:

   Update/transaction tapes (Y/N)?

   You have to load every transaction tape since the last full file-save in the same sequence as they were made. Mount reel #1 of the first transaction session tape set made after the last full file-save and answer Y to the above question.

   After a "Y" reply, the system reads the tape label and the sequencing information on tape and displays them in the following format:

```
L 01F4 21:05:10  06 SEP 1983 TRANSACTION LOG
Seq# of this data tape:       2     2     0     2
   continued from tape:       2     2     0     1
Seq# of last data tape:       2     2     0     1
Is this the right tape (Y/N)? |     |     |     |
                              |     |     | Trans.#
                              |     |   Update seq#
                              |   Reset seq#
                  File-save seq#
```

The numbers in the above example correspond to the f, r, u, and t sequence numbers. The line "Seq# of this data tape" and the line "continued from" show, respectively, the sequencing information and the "continuation" sequencing information of the transaction tape about to be restored. The line "Seq# of last data tape" shows the sequencing information of the transaction tape just restored. The above example shows that the tape to be restored is a logical continuation of the last tape just restore. Again, the operator is given an opportunity to verify that the correct transaction tape is mounted.

If a transaction tape is not a logical continuation of the last transaction tape, then the following message will be displayed:

```
L 01F4 13:01:10  09 SEP 1983 TRANSACTION LOG
Seq# of this data tape:       2     2     0     3
Seq# of last data tape:       2     2     0     2
Is this the right tape (Y/N)?
```

Note that the continuation sequencing information is missing in the above example. Although there is nothing "wrong" as far as the full-restore software is concerned, you must be aware that some data may be lost because the logger was deactivated for a period of time.

If an attempt is made to restore any tape in the wrong order, a message similar to the following will be displayed:

```
L 01F4 4:09:11  05 SEP 1983 TRANSACTION LOG
Seq# of this data tape:       1     1     0     2
   continued from tape:       1     1     0     1
Seq# of last data tape:       2     2     0     1
Tape out of sequence; override (Y/N)?
```

It can be seen from the sequence numbers that an old transaction tape has been loaded. Answering N will cause the system to ask "Update/transaction tape (Y/N)?" again. However, you may answer Y to override the sequence check if so intended.

After the end of reel #1 is reached, the system will automatically unload the tape and display the following message:

```
Mount reel #2
Continue or quit (C/Q)?
```

Mount the next reel in the same transaction session set and enter C to continue.

6. After all reels of the transaction session tape set have been loaded, the system will ask again

   ```
   Update/transaction tape (Y/N)?
   ```

   Load Reel #1 of the next transaction session tape set, if any, and answer Y. If there is no more tape set, answer N to complete the file restore.

<u>Error Conditions with Transaction Tapes</u>

This section explains some error conditions that may arise when using the transaction logger.

<u>System Failure During Transaction Logging</u>

If the transaction logger is deactivated gracefully by selecting Option 2 (Deactivate logger; exit menu) in the Transaction Logger menu, or if the tape is suspended by selecting Option 3 (Suspend tape) the transaction tape is <u>properly</u> <u>terminated</u> by a tape End Of File (EOF) mark.

But, when a system failure occurs with a transaction tape already started, the operator must be aware that:

1. the transaction tape is <u>not</u> terminated properly (no EOF mark is written on tape), and

2. the last few transactions may be lost.

The operator must handle the last transaction tape, which is not properly terminated, with care. See below.

<u>System with Battery Backup Unit</u>

If there is a power failure and the system is equipped with a battery back-up unit, the system may auto-warmstart after the power is restored to normal. However, the tape drive loses the vacuum (or tension) necessary to hold the tape. In this case, the operator must remove the current reel of transaction tape and mount a new reel of tape on the drive. Due to uncertainty of the tape drive operation at the time of power failure, a few transactions may be lost. Also, the reel of tape at the time of power failure is <u>not</u> terminated properly. (No EOF mark was written on tape.) The new reel of tape mounted after the system auto-warmstart belongs to the same transaction session tape set.

Restore with Transaction Tapes Not Properly Terminated

At the time of full-restore, if the operator tries to load a transaction tape not properly terminated due to a system failure, the operator may get the message

    Parity error (A)ccept/(R)etry/(Q)uit

at the spot where the system failure occurred. In this case, the operator should enter Q to quit the current reel of tape. The system will, again, ask "Update/transaction tapes (Y/N)?". The operator should then remove the tape, mount the next reel of transaction tape (if any), and answer Y.

Transactions in "DX" Accounts or "DX" Files

When doing a full file-save, if the file definition item (the D pointer) of an account or a file is changed to "DX", the account or file is not saved in the file-save tape.

However, since the transaction logger treats all updates as transactions, any update to any file in the "DX" account or any change to the "DX" file is logged as a transaction on the transaction tape.

At the time of full-restore, these accounts or files will not show up because the "reference" file-save tape does not have such files. Error conditions will occur when loading the transaction tapes.

An error message similar to the following will be displayed when the "illegal" transaction is encountered:

    Missing file; base 12345
    Continue/quit (C/Q)?

The base number is the base of the "DX" file. (The base, modulo and separation of the file that a transaction "belongs to" is recorded on the transaction tape. The file name is not used by the transaction logger to avoid the problems of multiple Q pointers to the same file and the same file name for different files in different accounts.) In this case, the file that the transaction belonged to is not restored because it was a "DX" file.

Answering C to the above error message will cause the restore processor to ignore this transaction and to continue loading the next transactions. If Q is answered, the restore processor will ignore the rest of this transaction session tape set and ask again "Update/transaction tapes (Y/N)?". The user should then unload this transaction tape and, if there are more transaction session tape sets, load reel #1 of the next set and answer Y.

Updating "DX" Pointers

When the Transaction Logger is activated, "DX" file definition items ("DX" pointers) should never be updated. (Remember that writing the same copy of an item back to the system will cause a transaction to be logged.)

If a transaction is logged because an update is made to a "DX" file definition item, the file definition item will be restored back to the system at the time of full-restore. However, since the "DX" file was not saved on the "reference" file-save tape, the file definition item itself is restored from the transaction tape but the item is a bad D pointer and "points" to nowhere. Referencing the bad D pointer may destroy other existing files and may cause GFE's (Group Format Errors) in the system.

NOTE: The above warnings for "DX" pointers apply also to "DV" and "DW" pointers when they are used the same way "DX" pointers are used (to inhibit file-save).

Data Loss Due to Deactivation of Transaction Logger

If the Transaction Logger has been deactivated for a period of time, data may be lost when full-restoring the system with the transaction tapes.

There are several possibilities that might cause data loss:

1. An item update (change, deletion or insertion) has taken place during the deactivation period;

2. A file or an account has been created during this period.

In the secohd case, even if the logger was re-activated and item updates to that file or account were logged subsequently, error conditions will occur when loading the transaction tapes.

At the time of full-restore, that file or account will not be re-created because the transaction tape does not contain the file-creation transaction. An error message similar to the following will be displayed when the subsequent item-update transactions to that file or account are encountered:

```
Missing file; base 12345
Continue/quit (C/Q)?
```

The base number is the base of the file created during the deactivation period.

Answering C to the above error message will cause the restore processor to ignore this transaction and to continue loading the next transactions. If Q is answered, the restore processor will ignore the rest of this transaction tape and ask again "Update/transaction tapes (Y/N)?". The user should then unload this transaction tape and, if there are more transaction tapes, load the next one and answer Y.

Data Loss Due to Account-Restore

Remember that ACCOUNT-RESTOREs are not logged by the transaction logger. An ACCOUNT-RESTORE after a FILE-SAVE may cause data loss because the account will not show up at the time of full-restore. When transactions generated by the account are encountered by the restore processor, the "Missing file; base 12345. Continue or quit" message will be displayed.

3.5.6 USING UPDATE SAVES AND TRANSACTION LOGGING TOGETHER

It is possible to use update saves along with the transaction logger. By using the transaction logger, system updates can be kept up to the minute, and, by using update saves regularly, fewer tapes are needed for transaction logging and less time is needed for system full-restore.

However, it is the System Manager's responsibility to keep track of the tapes, so that, at the time of full-restore, the correct tapes can be loaded in the correct sequence.

Procedure for Using Update Saves and Transaction Logger

The procedure to use update saves along with transaction logging is:

1. The transaction logger is activated immediately after a full file-save. The user may keep the transaction logger activated henceforth.

2. Update saves (either ALL-UPDATE-SAVE or PART-UPDATE-SAVE) are done on a regular basis, possibly daily.

3. Full file-saves are done on a regular, but less frequent, basis, possibly weekly.

Now, transaction tapes are associated with the latest "save" tape, either update save or full file-save. That is, if the latest "save" is a full file-save, transaction tapes made after the save are associated with this file-save tape. On the other hand, if an update save is done, transaction tapes made after this update save are associated with this update save tape.

Figures A and B illustrate this procedure.

Restoring with Update Save Tapes and Transaction Tapes

To restore the system, the procedure must be followed in sequence:

1. The latest full file-save tape is loaded.

2. Then, if there are update save tapes, they are loaded depending on whether ALL-UPDATE-SAVE's or PART-UPDATE-SAVE's were done.

   If the user has been doing ALL-UPDATE-SAVE's, only the latest update save tape is needed to be loaded.

   If the user has been doing PART-UPDATE-SAVE's, all update save tapes since the latest full file-save must be loaded in the same sequence as they were made.

3. Then, if there are transaction tapes, the tapes associated with the last "save" are loaded in the same sequence as they were made.

Figures A and B also illustrate how tapes are loaded at full-restore.

RECOMMENDATIONS

As can be seen in the above restore procedure, after a save is done, transaction tapes made before the save are no longer needed. However, we recommend that the user keep at least two "generations" of file-save tapes, and their associated update save and transaction tapes. In cases where one of those tapes is lost or has too many parity errors, data can still be restored by using the other "generation" of tapes.

EXAMPLE ONE: ALL-UPDATE-SAVE AND TRANSACTION LOGGING

1. Assume that a System Manager is upgrading his system from an older revision to this new revision. He full-restores the SYS-GEN tape, Then he ACCOUNT-RESTOREs all his application accounts from the last file save and he follows the appropriate steps in the UPGRADE PROCEDURE to complete the system upgrade.

2. The System Manager decides to use the ALL-UPDATE-SAVE method for regular system back-up. He also activates the transaction logger for up-to-the-minute system back-up.

```
                             SEQ# ON
                             SAVE TAPE SEQ# ON TRANS TAPE
                              f r u t   f r u t * f r u t
                             --------- ------------------
MAY 1 system upgrade
MAY 1 5pm full FILE-SAVE      1 1 0 0
MAY 1 9pm FILE-SAVE done
MAY 1 9pm activate Logger               1 1 0 1
MAY 2 5pm suspend trans. tape
MAY 2 5pm ALL-UPDATE-SAVE     1 1 1 0
MAY 2 6pm ALL-UPDATE-SAVE done
MAY 2 6pm restart trans. tape           1 1 1 1 * 1 1 0 1
MAY 3 5pm suspend trans. tape
MAY 3 5pm ALL-UPDATE-SAVE     1 1 2 0
MAY 3 6pm ALL-UPDATE-SAVE done
MAY 3 6pm restart trans. tape           1 1 2 1 * 1 1 1 1
MAY 4 5pm suspend trans. tape
MAY 4 5pm full FILE-SAVE      2 2 0 0
MAY 4 9pm full FILE-SAVE done
MAY 4 9pm restart trans. tape           2 2 0 1 * 1 1 2 1
MAY 5 5pm suspend trans. tape
MAY 5 5pm ALL-UPDATE-SAVE     2 2 1 0
MAY 5 6pm ALL-UPDATE-SAVE done
MAY 5 6pm restart trans. tape           2 2 1 1 * 2 2 0 1
MAY 6 10am suspend tape to do T-DUMP
MAY 6 11am restart tape                 2 2 1 2 * 2 2 1 1
```

NOTE 1: The four numbers shown in each column correspond to the f, r, u and t sequence numbers (respectively, full file-save, R-option, Update save and Transaction sequence numbers).

NOTE 2: The update save sequence number was reset to zero every time the SAVE command was used (either full FILE-SAVE's or update saves).

...

Figure A. ALL-UPDATE-SAVE with Transaction Logging

Notes for ALL-UPDATE-SAVE and Transaction Logging (cont.)

NOTE 3: After a save is done, ALL-UPDATE-SAVE tapes and transaction tapes made previously are not needed and can be re-used. However, Ultimate recommends that the user keep at least two "generations" of system back-up tapes.

For example, after the full file-save save (seq#=2 2 0 0) was completed at MAY 4, 9:00pm, the previous file-save, update save and transaction tapes (seq#=1 1 0 0, 1 1 0 1, 1 1 1 0, 1 1 1 1, 1 1 2 0 and 1 1 2 1) were not needed and could be re-used. But, Ultimate's recommendation is to keep generation one (all of the above tapes) until the file-save tape of generation three (seq#=3 3 0 0) is completed.

NOTE 4: If, for any reason, the user wants to restore the system on MAY 7, the tapes have to be loaded in the correct sequence, as shown below:

```
correct restore sequence
--------------------------
Full file-save   2 2 0 0
ALL-UPDATE-SAVE  2 2 1 0
Transaction tape 2 2 1 1
Transaction tape 2 2 1 2
```

Figure A (cont.). Example of ALL-UPDATE-SAVE and Trans. Logs

EXAMPLE TWO: PART-UPDATE-SAVE AND TRANSACTION LOGGING

1. Assume that we have a similar situation as in Figure A, but the System Manager uses the PART-UPDATE-SAVE method:

	SEQ# ON SAVE TAPE f r u t	SEQ# ON TRANS TAPE f r u t * f r u t
MAY 1 system upgrade		
MAY 1 5pm full FILE-SAVE	1 1 0 0	
MAY 1 9pm FILE-SAVE done		
MAY 1 9pm activate Logger		1 1 0 1
MAY 2 5pm suspend trans. tape		
MAY 2 5pm PART-UPDATE-SAVE	1 2 1 0	
MAY 2 6pm PART-UPDATE-SAVE done		
MAY 2 6pm restart trans. tape		1 2 1 1 * 1 1 0 1
MAY 3 5pm suspend trans. tape		
MAY 3 5pm PART-UPDATE-SAVE	1 3 2 0	
MAY 3 6pm PART-UPDATE-SAVE done		
MAY 3 6pm restart trans. tape		1 3 2 1 * 1 2 1 1
MAY 4 5pm suspend trans. tape		
MAY 4 5pm PART-UPDATE-SAVE	1 4 3 0	
MAY 4 6pm PART-UPDATE-SAVE done		
MAY 4 6pm restart trans. tape		1 4 3 1 * 1 3 2 1
MAY 5 10am suspend tape to do T-DUMP		
MAY 5 11am restart tape		1 4 3 2 * 1 4 3 1

NOTE 1: Recall that all PART-UPDATE-SAVE tapes after the last full file-save are needed to full-restore the system. If the user must re-use (or discard) some tapes, he must re-use only those <u>transaction</u> tapes made <u>before</u> the latest PART-UPDATE-SAVE.

For example, after the PART-UPDATE-SAVE tape (seq#=1 3 2 0) was done, transaction tapes (1 1 0 1 and 1 2 1 1) could be re-used.

However, Ultimate recommends that the user keep at least two "generations" of system back-up tapes.

NOTE 2: If, for any reason, the user wants to restore the system on MAY 6, he has to restore the tapes in the correct sequence: Several possible sequences can be used. Two examples are given below:

restore sequence example 1		restore sequence example 2	
Full file-save	1 1 0 0	Full file-save	1 1 0 0
PART-UPDATE-SAVE	1 2 1 0	PART-UPDATE-SAVE	1 2 1 0
PART-UPDATE-SAVE	1 3 2 0	PART-UPDATE-SAVE	1 3 2 0
PART-UPDATE-SAVE	1 4 3 0	Transaction tape	1 3 2 1
Transaction tape	1 4 3 1	Transaction tape	1 4 3 1
Transaction tape	1 4 3 2	Transaction tape	1 4 3 2

Figure B. Example of PART-UPDATE-SAVE and Transaction Logs

3.6 FILE REALLOCATION: AT FILE SAVE TIME

The file reallocation feature allows users to direct the system to reorganize the data base for maximum efficiency. The Ultimate File Reallocation software is menu driven, and calculates the best modulo for a file based on its current size as indicated in the STAT-FILE. The STAT-FILE is re-created each time a system File-Save is performed. File reallocation can be requested from the REALLOCATE command or from the SYSPROG Main Menu.

Individual accounts or files can be assigned reallocation parameters (new modulo and separation) to take effect at the next file restore via the UPDATE-ACCOUNT or UPDATE-FILE commands.

The Reallocation software is invoked by logging onto the SYSPROG account, and selecting the File Reallocation Menu from the SYSPROG Main Menu. Or, at TCL, enter the system command, REALLOCATE.

Review of File Allocation and Use of Modulo and Separation

A user data file is made up of items. These items are stored in groups (also referred to as "buckets") instead of one long sequential file. Figure A shows a sample file and the groups into which it is divided. When a file is created, the user determines how big each group, or "bucket" will be. This value is the "separation" of the file. The user also determines how many groups the file will contain. This value is the "modulo" of the file. The larger the file, the larger the modulo (and optionally, the separation) should be.

For retrieval speed, each item is stored in a "group" of the file so that the system does not have to search through the entire file sequentially. Instead, the item is "hashed" to find out which group it falls into. Then the system only has to search through that group to find the item.

The "hashing" process is performed by a "hashing algorithm", which is simply a formula that calculates which of the groups to store the item in. Hashing means adding up the value of all characters that comprise the item-id; this returns some arbitrary number that has no significance by itself. However this number is then divided by the modulo. The remainder of the division added to the base FID of the file becomes the group the file will be stored in. Since the remainder can be no more than (modulo - 1), this ensures that all files will be assigned to one of the defined "buckets", or groups.

For example, assume an item-id hashes to "9". If, as in Figure A, the modulo is 3, the remainder is "0". The file would be stored in the first bucket. If the item-id hashes to "10", the remainder is "1" (the second bucket). If the item-id hashes to "11", it goes into the third bucket. If the item-id hashes to "12", it goes into the first bucket.

Why Reallocate? Faster User Data File Retrieval

Primary space is the initial space allocated to a file, consisting of (modulo x separation) frames starting at frame (base). As a file grows, additional filespace has to be linked onto the primary space. This space is obtained from the Available Space pool, also known as "overflow". File access from multiple linked frames adds to retrieval time. The system keeps adding or expanding items into this secondary filespace until the file is "reallocated" by a new modulo during a file-restore or account-restore into more groups. The items are then "rehashed" and stored into the new buckets, thereby shortening access time (it is hoped).

File Reallocation Notes

File reallocation is not automatic. The REALLOCATE command (or a similar procedure) must be used before saving a file in order to calculate a suggested new modulo for files. The actual reallocation is performed at the next file-restore.

Because of the format of the STAT-FILE, file names with embedded asterisks (*) will be ignored.

All reallocation parameters are stored in the REALLOC-FILE.

The File Reallocation Menu Options

Figure A shows the File Reallocation Menu. There are five (5) reallocation options available to the user, as follows:

1. REALLOCATE ALL FILES AUTOMATICALLY

   This will run options 2 and 5 to completion with no user intervention. This is the only method recommended for all users unless they are experts with a complete understanding of the Ultimate system and file structure.

2. CALCULATE SUGGESTED REALLOCATION PARAMETER FOR ALL FILES

   Using the data in the STAT-FILE, the Automatic Reallocation Processor will calculate the best possible Modulo for all files.

   A new modulo will be suggested only if the calculated modulo is greater than the current one. A file will never be reduced in filespace size.

   The calculation is as follows:

   a. Calculate number of items with growth allowance (GROWTH=1.05): NEW.ITEMS = INT((ITEMS * GROWTH) + .5)

b. Calculate new file size

1. AVERAGE ITEM SIZE >= 500:
   NEW.MODULO = NEW.ITEMS

2. AVERAGE ITEM SIZE < 500:
   NEW.MODULO=INT(NEW.ITEMS/INT(500/AVG.ITEM.SIZE)+.5)

c. Set the new modulo equal to the nearest prime number that is greater than or equal to the new modulo that was just calculated.

Files are not updated at this time. To update files with Reallocation parameters, use option #5.

3. LIST REALLOCATION PARAMETERS TO PRINTER

A printed report will be produced consisting of Reel#, sequence#, file level (1=account, 2=dictionary and 3=data section), account name, dictionary name, file name, current modulo, current separation, new modulo and new separation.

4. REALLOCATION ENTRY MAINTENANCE

After the Reallocation Processor calculates a suggested new modulo (i.e. after option 1 or 2) you may use your own modulo for reallocation if you do not agree with the suggested modulo. To do this you must know the Reel# and Seq# of the file as it appears on the File Statistics Report (to produce a file statistics report, enter LIST-FILE-STATS at TCL in SYSPROG).

You will first be prompted for the Reel Number. If you do not wish to make any changes simply enter END at this prompt. After entering the Reel# and Seq# the following data will be displayed:

Account name, file name, file level, current modulo, current separation, current reallocation (if a reallocation parameter was calculated previously), suggested modulo (to be used as a reallocation parameter) and suggested separation.

You will then be prompted for the new modulo.

If you decide that you no longer want to change the suggested modulo of this file you can terminate the reallocation procedure by entering END at this time and you will be returned to the reel number prompt.

Files are not updated at this time. To update files with Reallocation parameter use option #5.

5. UPDATE FILES WITH NEW REALLOCATION PARAMETERS

Using the suggested 'New Modulo' from a previous calculation, all files that have been designated for reallocation will be updated with the new modulo as a reallocation parameter.

A FILE-SAVE and FILE-RESTORE must be done at this time to perform the actual reallocation.

```
               The ULTIMATE File Reallocation Menu
                           Time   Date

1.  Reallocate all files automatically
2.  Calculate Suggested Reallocation Parameter for All Files
3.  List Reallocation Parameters to Printer
4.  Reallocation Entry Maintenance
5.  Update Files with New Reallocation Parameters
88. Logoff
99. Go to TCL

     Selection
```

Figure A. File Reallocation Menu

```
FILE MYDATA's D POINTER ITEM:

001  D                    File Type Identifier (D pointer)
002  5000                 Base (Frame at which file begins)
003  3                    Modulo (Number of groups in file)
004  2                    Separation (Number of frames in group)
...
-----------------------------------------------------
| 5000 | 5001 | 5002 | 5003 | 5004 | 5005 | 5006
-----------------------------------------------------
---Group 0---- ---Group 1--- --Group 2----
```

In file MYDATA, a total of 6 frames (FIDS) are allocated to its primary filespace. Each item is "hashed" into one of these three groups. If an item is too long to fit into the primary space, additional frames are allocated and linked, as needed. This is known as "secondary filespace".

Since one data frame contains 500 bytes of information, the total primary space for this file is 3000 bytes.

Figure B. File Allocation Parameters

3.7 FILE RESTORE PROCEDURES

After a File-save has been performed, in order to load the saved tape(s) it is necessary to perform a File-restore. File restoring loads all system or user files saved on the tape(s). Usually, a coldstart is done as part of a File-restore; coldstarting includes loading the ABS frames (FID1-1023). Files can be reloaded without coldstarting by using the :FILELOAD command.

The File-restore procedure assumes that one or more DATA tapes have been file saved. You may also File-restore (in conjunction with booting the system only) from a a SYS-GEN tape that contains user files. If you are using an Ultimate-supplied SYS-GEN tape, you should use the normal coldstart procedure. (See Section 2.2, COLDSTARTING THE SYSTEM.)

NOTE: If the File Statistics listing from the File Save shows group format errors (GFEs) in a file, there is a strong possibility that some items have not been saved properly. To correct this condition, see Section 5.9, HANDLING GROUP FORMAT ERRORS.

A File Restore procedure can be initiated in two ways:

1. When the system is bootstrapped, select the (F)ile Restore option.

2. At the system TCL level, enter a :FILELOAD command.

File Restores from the System (TCL) Level

Before attempting to do a File-restore from TCL, load the DATA tape you wish to use and bring it on-line. Do not use a SYS-GEN tape unless four T-FWD commands are used to skip over the boots (1 and 2), coldstart, and ABS sections first. Otherwise, tape format errors will be reported and the tape cannot be used.

Log onto the SYSPROG account and at the TCL level, enter:

```
>:FILELOAD
```

The current tape label and sequence number are read from the tape unit, as shown in Figure A. At the prompt:

```
Is this the right tape?(Y/N):
```

reply 'Y' to continue, or 'N' to load the correct tape.

File Restores after Booting the System from a SYS-GEN Tape

After booting the system, select the 'F' option at the System Startup Options prompt.

The system reports the currently mounted tape label and sequence and prompts for acceptance (see Figure A for sample message).

To mount another tape, answer 'N' to the prompt and remount the correct tape. A 'Y' response continues the file restore.

Sequence of Events in File-restores

The first event in a complete file-restore is the initialization of available overflow space to the complete range on the system from the Process workspaces (WSSTART) forward to the last available frame on the disk (MAXFID).

The SYSTEM dictionary is then created and cleared. Then the first account Master Dictionary (MD) is created and a pointer to it is placed in the SYSTEM dictionary. Then the first file dictionary is created and a pointer to it is placed in the account's Master Dictionary. Next is the data file, which will proceed in one of two ways:

1. The slow method. The file is created, a pointer is added to the dictionary, and then the data is loaded. This method is necessary if reallocation is being done.

2. The fast method. The file is loaded group by group as it is created. After it is completely loaded, a pointer is placed in the dictionary. This is the normal method.

Next, the file dictionary is loaded. The next file's dictionary and data sections are created and loaded, and so forth until all of the accounts are finished, when the SYSTEM dictionary is loaded. This completes the file-restore.

Account-restores proceed in the same sequence, except that the SYSTEM Dictionary is already present, and only the pointer to the account Master Dictionary is added to it. See Section 3.10, ACCOUNT RESTORING for details.

For multiple tape reel/removable disk operation, see Section 3.3.1, FILE SAVING WITH MULTIPLE TAPE REELS OR DISKS.

NOTE: File-restores will run slower on files that have been re-allocated.

Terminal Listing Accompanying File-restore

Figure A is an example of a file-restore listing. Each line corresponds to a file pointer. Each line is indented in accordance with the level of the file in which the pointer is placed. The file name is first followed by the base, modulo, and separation of the file as it is being restored. An (S) following the line indicates that the pointer has the same base as some other pointer already listed and that that file has already been created.

Restoring with Update Save or Transaction Tapes

The last message prompts for update or transaction tapes (if any). You may mount these tapes after responding 'Y' or continue by responding 'N'.

Section 3.4, UPDATE SAVES, and Section 3.5, TRANSACTION LOGGING, explain in detail these concepts and the procedures to use to create and restore the update and transaction tapes. Please consult these sections before attempting to use the Update Save or Transaction Logger features.

The Coldstart Procedure After a File Restore

The file restore procedure then saves the Monitor, links workspace, and then automatically logs on to the COLDSTART account (see Figure B).

The COLDSTART account is a Q-pointer to the SYSPROG account. Its LOGON PROC is set up to ask the user for the current time and date.

Figure C shows the sequence of messages associated with the coldstart procedure. The secondary TCL workspaces are linked. The operator is prompted to enter the current time and date; the operator entry is shown in boldface type. The user modes and other system software are loaded, reported, and verified. The procedure ends with the logoff message (from the COLDSTART account) and the system logon message.

All lines are locked out during this procedure until it concludes with the Logon please message. The COLDSTART proc releases these lines by executing the :ACTIVATE-LINES command before logging off.

Users may then log on to any user account and resume work.

Error Recovery During File Loads

If parity errors or other errors mar the files section of a FILE-SAVE tape, some data may be lost. The file-restore will continue, but operator assistance may be needed.

Parity Error Recovery Procedure

If a parity error is detected on a file restore, the prompt:

```
Parity error!
(A)ccept/(R)etry/(Q)uit?
```

will be printed. Entering 'A' will cause the data block to be accepted as it is from tape without data correction. The specific item and file affected cannot be determined except as can be judged by the tape position, and the current set of files which have not been completed. Enter 'R' to retry; enter 'Q' to cancel the restore.

Recovery From Destroyed Pointers

If tape information identifying a file is destroyed, it may be impossible for the restore to create that file and subsequent files in the right order. the message:

```
Err in "D" segment
@fff.ddd
Level (1-3)?'
```

will be printed. fff.ddd gives the frame and hex displacement of the software location at which the error was detected. The operator is expected to advise the Restore software whether to:

1. Search for and continue with the next account on tape (level 1),

2. Search for the next dictionary file on tape (level 2), or

3. Search for the next data file on tape (level 3).

The response requires the operator's judgment as to the positioning of files on the tape and the total situation.

```
L 2000#04:54:32 29 JUL 1985 DATA FILE SAVE
Seq# of this data tape:     0      0     0     0
Is this the right tape (Y/N)?

SYSTEM 9345,11,1            SYSTEM dictionary pointer
 SYSTEM-ERRORS 9391,1,1     SYSTEM ptr. to SYSTEM-ERRORS account
  SYSTEM-ERRORS 9392,1,1    DICT SYSTEM-ERRORS file in
                               SYSTEM-ERRORS account
   SYSTEM-ERRORS 9393,24,1  DATA SYSTEM-ERRORS file
BLOCK-CONVERT 9422,14,1     BLOCK-CONVERT file (single-level)
SECURITY 9447,29,1          SYSTEM pointer to SECURITY account
...
...
...
SYSTEM 9345,11,1 (S)        SYSTEM dictionary pointer
                               (duplicate entry on tape)
SYSPROG 9853,17,1           SYSTEM pointer to SYSPROG account
...
...
...
CONVERSION 10578,1,1        Shared dictionary
 DOC 10374,1,1              Multiple DATA pointers
 PGM 10608,3,1                 "
 TABLE 10663,1,1               "
 PROC 10665,7,1                "
...
...
...
...                         Other accounts
...
ACC 15108,1,1               SYSTEM pointer to ACC account
 ACC 15109,11,1             DICT ACC file in ACC account
  ACC 15120,81,1            DATA ACC file

Update/transaction tapes (Y/N)?
```

Figure A. Sample Listing of Files Restored

```
Saving Monitor

Spooler started
Linking workspace for line 0

05 FEB 1986 08:22:14 Logon please:

<<<< Welcome to the Ultimate Computer System >>>>
<<<< Copyright July, 1981  The Ultimate Corp. >>>
<<<< 08:22:19 Release 10 Rev 140J 26 AUG 1985 >>>

This is the Cold-Start Procedure
     Enter <CR> to continue
```

Figure B. Sample File Restore Messages

```
Linking secondary TCL workspaces

[294] 1 additional task workspaces initialized

08:43:26 26 AUG 1985

Time:12:30
12:30:02 26 AUG 1985
Date:06 FEB 1986
12:30:32 06 FEB 1986

[216] Mode 'CURSUR'    loaded;  Frame = 422  Size = 187  Cksum = 5966
[216] Mode 'ENT440'    loaded;  Frame = 442  Size = 005  Cksum = ED
[216] Mode 'IIKERNEL2' loaded;  Frame = 444  Size = 191  Cksum = 9FCD
<<<--Load terminal definitions-->>>
<<<------Apply OS Patches ------>>>
UFOS FILE EMPTY
<<<-----Verifying software------>>>
[341] Ultimate system software verified.
SP-STARTLPTR 0,0,1,S15
[1118]
 The printer control block has been initialized.
 The correct paper and LPI settings must have been
 previously set to insure proper printing.


< Connect time= 23659 mins.; CPU=465 units; lptr pages= 0 >
< Logged off at 10:01:55  on 06 FEB 1986                  >

06 FEB 1986 10:01:55 Logon please:
```

Figure C. COLDSTART Procedure Messages

3.8 THE FILE STATISTICS REPORT

The File Statistics Report is a valuable tool for file management. This report is automatically generated by running a FILE-SAVE, or may be generated at any time by using the LIST-FILE-STATS command. The STAT-FILE file is created by using the S option of the SAVE verb.

The FILE-SAVE process creates one item in the STAT-FILE for each D pointer item saved on the file-save tape. A listing of the STAT-FILE is generated at the end of every file-save. This listing can be output via the LIST-FILE-STATS command.

The File Statistics report adds data security by providing a list of file Base, Modulo and Separation parameters, and by recording the order of files on a FILE-SAVE tape.

The report is listed by account, with a line of information generated for each file in the account that includes:

* total and average item size
* total and average number of items per group
* utilization of file-space
* actual data stored, and "pad" space used in the file

A total line is generated for each account showing the total:

* items
* bytes (characters)
* frames (includes linked)
* group format errors

Creation of the STAT-FILE dictionary and data areas is part of the SYS-GEN procedure. STAT-FILE is contained on the System Programmer (SYSPROG) Account. As it is normally updated from this account, there is no need for STAT-FILE on any other account.

Alternately, the file may be created via the following:

```
CREATE-FILE STAT-FILE 3,1 31,1  <CR>
```

When a FILE-SAVE is started, the STAT-FILE data area is cleared and the current file statistics information is written into the data area.

The STAT-FILE data area will also be empty after a file-restore is done, because Attribute 1 of the file definition is a DY. This is desirable as the statistics are no longer applicable.

The item-id in the STAT-FILE has the general format:

```
t:n
```

where "t" is the tape reel number where the file was dumped;

this will be 0 if the SAVE was run without dumping data to the tape (no 'T' option); and "n" is the file-number. This file-number is used in the selective restoration of files via the "SEL-RESTORE" command.

The NAME field of the items in the STAT-FILE contains data in the format:

accountname*dictname*dataname

where one, two or all three of the fields may be present, depending on whether the file is an account, a dictionary, or a data-file.

```
PAGE    1  ** F I L E   S T A T I S T I C S   R E P O R T ** 10:41:02 08 JUL 198

R#   ID NAME ... BASE MOD SEP  SIZE ITEM FRAM AV/ITM ITM/GP FRM/GP %UT. .PAD C

11166 ACC        412835  1 1      32    1    1     32    1.0    1.0   6     468
11167 ACC        412836 11 1    2133   71   11     30    6.4    1.0  38    3367
11168 ACC*ACC    412847 81 1   10744   50   92    214    0.6    1.1  23   35256
                               -----  ---  ---                       ---  ------
***TOTALS FOR USER: ACC        12909  122  104                        24   39091


1 117 BASIC       57099  29 1   7820  277   31     28    9.5    1.0  50    7680
1 122 BP          58102   1 1    332    8   15     41    8.0    1.0  66     168
1 123 BP*BP       58103   5 1  13676   11   30  1,243    2.2    6.0  91    1324
1 118 NBC         57128   1 1    192    1    1     64    3.0    1.0  38     308
1 119 NBC*NBC     57129   7 1 487775   41  980 11,896    5.8  140.0  99    2225
1 120 TSYM        58070   1 1     33    1    1     33    1.0    1.0   6     467
1 121 TSYM*TSYM  58071  31 1      0    0   31           0.0    1.0   0   15500
                              ------  --- ----                       --  -----
***TOTALS FOR USER: BASIC     509828  341 1089                       93   27672


1   2 BLOCK-CONV 22703 19 1   8956  352   25     25   18.5    1.3  71    3544

1 329 BP         153931  1 1   3005   58  144     51   58.0    7.0  85     495
1 320 BP*BP      153932  3 1 132462   52  267   2547   17.3   89.0  99    1038
1 313 CAROL      142410 29 1  27532  695   70     39   23.9    2.4  78    7468
1 325 COMMSTEMP   85207  1 1     37    1    1     37    1.0    1.0   7     463
...
                          --------- ---- ----                        --- -----
***TOTALS FOR USER: CAROL 1,113,331 1920 2825                          78 60169

257 ITEMS LISTED.
```

Figure A. Example of a STAT-FILE Report

3.9 WORKSPACE ALLOCATION AND AVAILABLE SPACE AFTER RESTORE

After a file-restore, initial primary and secondary workspaces are allocated to Processes, one per terminal port plus "phantom" Processes without ports (such as the print Spooler). The BASE of the SYSTEM dictionary starts after the Spooler's allocated workspace. Overflow, or Available Space, consists of all unused frames between here (SYSBASE) and the highest data frame (MAXFID). After a restore, Overflow tends to be one or two chunks of contiguous frames. Update restores and transaction restores may later fragment the Overflow table.

Overflow grows and shrinks as frames are used for file data and secondary TCL workspaces. The Overflow table has three parts: a contiguous-frame section, a linked-frame section, and a secondary-TCL-workspace section.

The contiguous-frame section can keep track of up to 31 blocks of contiguous available frames. After that, available frames are kept track of using a variable-size linked set of frames. The secondary TCL workspace section maintains blocks of frames for use as Process workspaces.

Each terminal line may have multiple, stacked TCL levels; that is, it may use more than one Process workspace. Each secondary TCL level of workspaces is allocated from the secondary TCL workspace section, as needed. Each level requires a set of 416 contiguous frames. When the secondary TCL level is no longer needed, the system releases the space back to the pool.

For example, the BASIC statement "EXECUTE" actually creates a secondary TCL level and executes its system command in the secondary level. During this execution, the BASIC program in the previous TCL level is temporarily suspended. When the EXECUTE statement is finished, the secondary workspaces are released and control returns to the previous TCL level (the BASIC program).

If files were reallocated during the File Restore, the Overflow pool would reflect the loss of frames now allocated to primary filespace.

The system command POVF is used to "print the Overflow" space status. It shows both regular and secondary levels. Figure A shows a sample of this command used immediately after a File Restore. Note that the secondary Overflow space begins at frame 5184 with 416 free frames. Although these 416 frames are not counted in the TOTAL NUMBER OF CONTIGUOUS FRAMES, they will be released back to the regular Overflow space when (and if) needed. (For more information, see the POVF command in the System Commands Guide.)

Figure B shows a POVF command after system usage has fragmented the Overflow pool. However, there is no logical difference between linked and contiguous frames. The fragmented condition only becomes a problem if some system software is unable to perform a function, such as CREATE-FILE, due to the lack of contiguous frames.

For troubleshooting information, Sections 5.3 and 5.4 deal with MULTIPLE TCL LEVELS, and how to handle problems that may occur with the multiple TCL level feature, such as using POVF and the System Debugger. Section 5.5 covers handling workspace allocation, the LINK-WS command and suggestions for correcting workspace problems.

```
>POVF <CR>
                                    5184 (   416)
 23448-131167 : 107720

 Total number of contiguous frames= 107720
```

Figure A. Results of POVF After a File Restore

```
>POVF <CR>
   23459 (400)
    8112-  8117 (    6)    9000-  9000 (    1)
   23789- 23801 (   13)   25000- 25678 (  679)
   25681- 25692 (   12)   27123- 27323 (  201)
   34502- 35123 (  522)   35800- 35801 (    2)
   37091- 37091 (    1)   37093- 37093 (    1)
   37099- 37100 (    2)   38100- 38100 (    1)
   43100- 44234 ( 1135)   45680- 45681 (    2)
   46343- 46443 (  101)   46445- 46445 (    1)
   46448- 46448 (    1)   46451- 46451 (    1)
   46454- 46454 (    1)   46458- 46474 (   17)
   47011- 47444 (  434)   47460- 47492 (   33)
   47661- 47750 (   90)   48012- 48017 (    6)
   48018- 48018 (    1)   48020- 48101 (   82)
   48233- 48268 (   36)   48299- 48299 (    1)
   51111- 53234 ( 2124)   53400- 53601 (  202)
   60000- 97799 (37800)
  Total number of contiguous frames= 43509
```

Figure B. Results of POVF During System Operation

3.10 ACCOUNT SAVE AND RESTORE PROCEDURES

Single accounts can be saved to or restored from tape. The ACCOUNT-SAVE command allows a user to write one account to a save tape. The ACCOUNT-RESTORE command adds a single account from tape to an already existing ULTIMATE system.

When an individual account is saved, the system allows you to specify whether the password associated with that account should be written to tape. Removing passwords prevents revealing them to unauthorized personnel, thus protecting system security.

When an individual account is restored, the system allows you to specify the password, if saved, and/or to specify a new password(s), if desired. Specifying new passwords at restore time guarantees that there will be no window in time during which an account may have no password.

Account Save Procedures

The ACCOUNT-SAVE command functions similarly to the FILE-SAVE command. The files section contains no System Dictionary pointer or items, and only one account is saved. If STAT-FILE items are generated, they will pertain only to the saved account.

Follow this procedure to save an individual account:

1. Log onto the SYSPROG account.
2. Mount a tape with a write ring (allows tape to be written to). Or, make sure the tape or disk cartridge is not "protected" or "safe".
3. Attach the tape (T-ATT command).
4. At the system (TCL) level, enter:

   ACCOUNT-SAVE <CR>

5. At the prompt, specify a tape label or <CR> for a label of simply 'DATA':

   File-save tape label = tape-label <CR>

6. At the prompt, specify the following:

   Account name = account-to-be saved <CR>
   Password(s) (Y/N)?

   An account name in the SYSTEM dictionary must be specified. To omit saving the password(s), answer "N". To save passwords, answer "Y". You are prompted:

   Password(s) or <CR> (use existing password(s)):

Enter the password(s) to save to tape, or press <CR> to save the password(s) currently associated with the account. Multiple passowrds must be separated by spaces.

For systems with multiple tape reels or disk packs for a single account, see Section 3.3.1, FILE SAVING WITH MULTIPLE TAPE REELS AND DISK PACKS.

Account Restore Procedures

An ACCOUNT-RESTORE operation can be performed from any file-save or SYS-GEN tape, as well as an ACCOUNT-SAVE tape. In any case, the account will be restored and a pointer to the account will be created in the SYSTEM dictionary.

If restoring from a SYS-GEN tape, you must move the tape forward four (4) files from load point to bypass the four initial sections of the SYS-GEN tape (BOOT 1, BOOT 2, COLDSTART, ABS). (Execute four (4) T-FWDs.)

NOTE: The account may not be restored with a name that already exists on the system.

ACCOUNT-RESTOREs may be started from any tape of a multi-tape file-save! To save time in searching a tape, the STAT-FILE listing may be consulted to determine which reel the account's data starts on, and that reel may be mounted.

Follow this procedure to restore an individual account:

1. Log on to the SYSPROG account.
2. Mount a tape without a write ring (prevents writing onto tape) and attach it (T-ATT command).
3. At the system (TCL) level, enter the command:

   ACCOUNT-RESTORE new-account-name <CR>

4. At the prompt, enter the account name under which the account was saved:

   Account name on tape:

   The account name must be the name under which the account was saved (D-pointer name); it cannot be an account synonym (Q-pointer name).

5. At the prompt, specify whether password(s) should be assigned to the account when it is restored:

   Password(s) (Y/N):

   If no password(s) is desired, answer "N". Otherwise, answer "Y".

6. If passwords are desired, you are prompted to either retain the same password(s) or specify new one(s):

    Password(s) or <CR> (use password(s) from tape)

    If new password(s) are entered, they are not echoed at the terminal, and you are prompted to enter the password(s) again:

    Re-enter password(s) to confirm:

    Simply type the same password(s) as before. Multiple passwords must be separated by spaces.

After the password(s) have been entered, if applicable, the tape is searched for the account, and the restore proceeds automatically.

For systems with multiple tape reels or disk packs for a single account, see Section 3.3.1, FILE SAVING WITH MULTIPLE TAPE REELS AND DISK PACKS.

3.11 SELECTIVE RESTORE PROCEDURES

The Selective-Restore capability allows individual files or items to be loaded onto an Ultimate system from a file-save, account-save, or SYS-GEN tape.

If restoring a file from a SYS-GEN tape, you must move the tape forward four (4) files from load point to bypass the four initial sections of the SYS-GEN tape (BOOT 1, BOOT 2, COLDSTART, ABS). (Execute four (4) T-FWDs.)

To restore a single file from tape, follow these procedures:

1. Log on to the account with the file to be restored.

2. Mount and attach the tape.

NOTE: Selective-restores may be started from any tape of a multi-tape file-save. To save time in searching a tape, the STAT-FILE listing may be consulted to determine which reel the file's data starts on, and that reel may be mounted. A selective-restore may be started at any place on any reel of the file-save tape.

3. At the system (TCL) level, enter:

```
>SEL-RESTORE {DICT} file-name {item-list} {(options)} <CR>
```

where "file-name" is the file in which items will be placed. This file must be defined on the account from which the restore is run. The optional item-list enumerates those items eligible for restore. '*' may be used to indicate all items on the tape.

The data may be restored from either a specific file-name on the tape, or a file-number; the file-number may be obtained from a listing of the STAT-FILE when the tape was created.

Applicable options are:

O Overlay items already on the file.

A The tape is already positioned in the desired account. In this case the "Account name on tape" prompt will not appear.

N The file is to be identified on tape by its file number, in which case the prompt will be "File#?". The required file # is the one which accompanies the file on the statistics file print-out for the appropriate file save.

I The item-ids of the restored items will not be printed.

C This option has effect when the 'N' option is used. It causes every item before the next end of file to be a candidate for restore. This ensures that data can be restored even if a D pointer is damaged on the tape.

S Skips forward spacing of the tape. This is used when at the beginning of the second or later reels of a file-save.

4. If neither the 'N' nor 'A' option is used, the operator will be prompted:

```
Account name on tape?account-name
File name? {DICT} file-name
```

where 'account-name' is the name of the account under which the file was saved on tape, and 'file-name' is the name of the file as it appears on the tape. Pressing <CR> to 'File name?' causes the account Master Dictionary (MD) to be restored.

The file-name may be in one of these formats: file-name; DICT file-name; or file-name,data-name.

If the N option is used, the prompt will be:

```
File #?
```

and the file-number must then be entered.

As the tape is searched, the file-names on it are printed, along with the file-numbers; names are indented one space for account-names, two spaces for dictionaries, and three for data-file-names.

Upon completion, this message is displayed:

```
[992] nnn item(s) have been restored.
```

Helpful Hints for Selective Restoring

If a STAT-FILE listing for the tape is available, ensure that the account-name and file-name are on the tape as you think they should be. In the case of multiple D-pointers in the SYSTEM dictionary to an account, or multiple D-pointers in the M/DICT to the file, the account-name or file-name on the tape will be the first one the Save software encounters, and may be different from the one commonly used by you. All other names will appear in the STAT-FILE listing with no data (null SIZE field), and cannot be specified in the SEL-RESTORE.

If in doubt about the contents of the tape, the files can be listed by creating a SEL-RESTORE command designed to provide information about the tape contents:

```
>SEL-RESTORE file-name *

Account name on tape? XXXXX
File name? YYYYY
```

where XXXXX and YYYYY are fake names that will cause the SEL-RESTORE to search the tape for non-existent data; files will be printed out as encountered, along with the file-numbers. Files with an (S) are synonyms, and should be ignored.

In restoring both the dictionary and data section of a file, restore the dictionary first (DICT filename). Remember that the dictionary items FOLLOW the data items, so for large files, there may be a considerable pause after the time that the system has found the file (it stops the printout), and the actual restore of the items.

At any point, the tape may be backed up (T-BCK (n)), or forward-spaced (T-FWD (n)) to position it, and a SEL-RESTORE with the A or N options may be started; this may be faster than restarting the tape from the beginning when restoring both the dictionary and the data sections of a file, or when restoring multiple files.

Remember also that account dictionaries (M/DICT items) FOLLOW ALL OTHER FILES for the account on the tape.

To restore the Q-pointers in the SYSTEM dictionary, use the N option with FILE# = 1. Remember that this will be the last file on the tape! On a multi-reel file-save, mount reel #1 first, and start the SEL-RESTORE as usual; when the file-name "SYSTEM" has printed out, use the BREAK key to interrupt the restore. Then mount the LAST reel of the set, and type "G<CR>" to continue the process. This saves searching the entire first reel and any intermediate reels of tape.

SECTION 4

PROCEDURES FOR SECURITY, USER ACCOUNTS, TERMINALS AND PERIPHERAL EQUIPMENT

SECTION 4

PROCEDURES FOR SECURITY, USER ACCOUNTS, TERMINALS AND PERIPHERAL EQUIPMENT (continued)

4.1 SETTING UP SECURITY STANDARDS FOR YOUR SYSTEM

In a multi-user computer system environment, especially when a network is involved, system protection and security are crucial to a successful system. The Ultimate system provides a number of system security features that each System Manager can use to prevent misuse of the data base and the programs.

Before setting up any specific users and equipment (peripherals such as terminals and printers), the System Manager should determine the amount of control and security to be exercised in the day-to-day operation of the system.

An Ultimate system contains a number of features that may be set up to maintain as closely monitored or open a system as desired. For example, two system accounts are included on the SYSGEN tape with no special password:

SYSPROG	System Management/Maintenance account
SECURITY	System Security/Administrator account

It is highly recommended that personnel responsible for the system establish a policy that determines which users will have access to these system accounts, and which users will have access to other user-defined accounts and files.

At the minimum, the SYSPROG and SECURITY accounts should be updated by the System Manager to assign them distinct passwords. This will restrict their access to personnel who have been given the "password".

System Managers of new systems should be aware of the SECURITY features discussed in this section, and take advantage of them.

NOTE: The COLDSTART procedure has a pre-defined password, which may be changed by the System Manager.

Background on the Security Provisions

Beginning with Release 10 Rev 130, higher level security functions are controlled from the SECURITY account. All functions may still also be controlled by SYSPROG, however, if desired. The SECURITY account was designed to limit the number of users needing access to SYSPROG, and to provide a separate account to be accessed only by those concerned with higher levels of system security.

The SECURITY account is defined with a SYS2 privilege level. In addition, it is never denied access to a file, regardless of update and retrieval locks. The SECURITY account may always logon to line zero (0). The account is supplied without a password, but one should be assigned as soon as the system has been installed.

To set up security for your system, logon to the SECURITY account and set up the initial system security specifications (Option 1 on menu).

All account passwords are encrypted, making it more difficult for someone to learn an account's password. An account may have multiple passwords. Logon attempts can be monitored and reported, and the line temporarily disabled when too many unsuccessful logons are attempted. This feature prevents someone from trying many password combinations to gain entrance to an Ultimate system. An individual line may be included or not in this disablement. This allows, for instance, for protecting only lines to which modems are attached.

If the logon monitoring is enabled, each unsuccessful logon or LOGTO attempt is recorded in the LOGONH file of the SECURITY account. This count is a "daily record" since it is reset daily and whenever the line is disabled. On the SECURITY menu, option 8, "Display Terminal Logon Failures", lists the day's logon failures.

The System Manager can define the lines which are allowed to logon to each system account. If someone knows the account name and password but doesn't have access to a line that allows entry to the account, the logon attempt will be unsuccessful. For example, an account concerned with payroll processing can be made available only on a terminal in the payroll manager's office.

When a user account or file is created, certain security parameters are defined. The critical security parameters can only be updated from the SECURITY account (or SYSPROG, if the security features are enabled in SYSPROG; that is, SECURITY and SYSPROG are "equal"). Users canot bypass file security and create their own file definition items. Each file has file access and update codes that contain the account names and UltiNet system names (if applicable) that are allowed to access or updated the file.

<u>Background about Passwords</u>

Passwords are encrypted. The only way to create or change a password is through the CREATE-ACCOUNT/UPDATE-ACCOUNT command (see Option 3 on menu). If an unencrypted password is manually inserted in an account D-pointer via the Editor, you will be unable to logon to that account.

The encrypted form of a password is unique to the account on which it is created. That is, password "ABC" would be encrypted differently if used as the password for account "RED" than as the password for account "BLUE".

Passwords may contain any ASCII (7-bit) characters except spaces.

An account may have multiple passwords, so that if a user and his password are added or removed, other users of that account are not affected. This is also intended to facilitate "master" password arrangements. This feature is useful, however, only on accounts which cannot run the UPDATE-ACCOUNT program since that would allow them to see all the passwords. (That is, the account should not be given access to the SYSTEM dictionary.)

When restoring accounts from tape, via the File Restore option at System Startup, or the :FILELOAD or ACCOUNT-RESTORE commands, the system automatically encrypts any passwords that are on the tape in unencrypted format. (Passwords already encrypted are not changed.)

Logon Monitoring

When a user enters an account name at the "Logon Please" prompt or with LOGTO, the Logon program retrieves the specified account's logon information to determine whether the user's line is allowed to logon to that account. If the line isn't accepted, a message is displayed:

    This line may not log on to the specified account.

When logon monitoring is enabled, if the unsuccessful logon count for a line exceeds the allowed value, the Process is put to sleep, the BREAK key is disabled and the line's data terminal ready (DTR) signal is turned off (thus disconnecting any telephone line which may be involved.)

The disablement affects only the unsuccessful line. After the specified period of sleep time, the logon attempt count is reset and the line reactivated. The line can be enabled from the SECURITY-STATUS command within option 2, "Terminal Security Specifications", on the SECURITY menu.

4.2 USING THE SECURITY ACCOUNT'S MAIN MENU

The SECURITY account's menu contains options to set up system and terminal security, create, update, rename, and delete accounts, save and restore accounts, display instances of failed logon attempts, and print the security system documentation.

To use the SECURITY account, you must logon to that account. After logon, a main menu is displayed (see Figure A).

To return to the system (TCL) level, enter 'TCL'. From TCL, to redisplay the menu, enter 'MENU'. To logoff the SECURITY account, enter 'OFF'. Each menu option is summarized below; some options are detailed in the following topics.

System Security Specifications (Option 1)

This option invokes a system command in the SECURITY master dictionary called SECURITY-STATUS. This program can be used to enable or disable assembly code operations, enable or disable logon monitoring, and enable lines disabled because of logon violations. (See Section 4.3, SETTING UP SYSTEM SECURITY and SECURITY-STATUS.)

Terminal Security Specifications (Option 2)

This option invokes a system command in the SECURITY master dictionary called TERMINAL. This program can be used to enable or disable logon monitoring for an individual line, and to vary the monitoring parameters. It can also be used to specify characteristics of individual ports such as the baud rate, parity, terminal type, etc. These characteristics can be stored as current parameters for the line or as Logon parameters, presently stored in DICT ACC. (See Section 4.4, SETTING UP TERMINAL SECURITY.)

Create/Update an Account (Option 3)

This option invokes a system command called CREATE-ACCOUNT (or its synonym, UPDATE-ACCOUNT). This program can be used to modify account parameters, such as passwords and valid logon lines, for any account on the system. It also creates and updates synonym (Q pointer item) accounts. From the SECURITY (and SYSPROG, if equal) accounts, any account can be created or modified. From a user account, only that account may be displayed and modified. (See Section 4.7, CREATING OR UPDATING AN ACCOUNT'S PARAMETERS.)

Rename an Account (Option 4)

This option invokes the RENAME-ACCOUNT command. It may be selected only from the SECURITY account (or SYSPROG, if equal). You must specify both the old and new account names. Any passwords are re-encrypted, and the logon proc or program in the account's master dictionary is also renamed. The retrieval and update locks are not changed. If they need to be updated, use the UPDATE-ACCOUNT option or command. Another command, RENAME-FILE, works the same way, but on a file instead of an account. (See the System Commands Guide for details on using these commands.)

Account Save to Tape (Option 5)

This option invokes the ACCOUNT-SAVE command. It allows the SECURITY account user to specify whether the account's password should be saved to tape, and to optionally assign a new password(s) for the ACCOUNT-RESTORE procedure. User accounts may only ACCOUNT-SAVE their own account. (See Section 3.10, ACCOUNT SAVES AND RESTORES, for procedures.)

Account Restore from Tape (Option 6)

This option invokes the ACCOUNT-RESTORE command. It allows the SECURITY account user to specify a password that is required to logon to the account. (See Section 3.10, ACCOUNT SAVES AND RESTORES, for procedures.)

Delete an Account (Option 7)

This option invokes the DELETE-ACCOUNT command. This option and the system command itself may only be executed from the SECURITY account (or SYSPROG, if equal). Files to be deleted may be either listed on the screen or printed on the printer. After the files are listed, you may verify that the account is to be deleted (or cancel the deletion). (See Section 4.10, DELETING A USER ACCOUNT, for procedures.)

Display Terminal Logon Failures (Option 8)

This option is valid only if logon monitoring and control was enabled from the SECURITY-STATUS command. All logon failures are reported in the LOGONH file in the SECURITY account. This is the only option that displays LOGONH file data (see Figure B). The line number is given, along with the account name and password (if any) entered, plus the time and date. If no password was entered, a "@" is displayed next to the user-id.

Security System Documentation (Option 9)

This option outputs the security documentation for the current release of your system. The document can be output to the printer or displayed on the terminal.

```
*******************************************
*      The ULTIMATE Accounts Manager      *
* Version #.#              Date           *
*******************************************

 1.  System Security Specifications
 2.  Terminal Security Specifications
 3.  Create/Update an Account
 4.  Rename an Account
 5.  Account Save to Tape
 6.  Account Restore from Tape
 7.  Delete an Account
 8.  Display Terminal Logon Failures
 9.  Security System Documentation

 Enter Selection, 'TCL', or 'OFF' _
```

Figure A. SECURITY Account Main Menu

```
Terminal Logon Failures for 28 FEB 1986

LOGON#... USER-ID...  PASSWORD....   DATE...  TIME

001       BOO @                      02/28/86 02:56
002       BOO HOO @                  02/28/86 02:58
          SYSTEM      ANY            02/28/86 03:00
004       DONNA       NEW            02/28/86 04:30

Hit Return to Return to Main Menu
```

Figure B. Sample Logon Failure Report

4.3 SETTING UP SYSTEM SECURITY (SECURITY-STATUS)

The System Security Specification (option 1) on the SECURITY menu invokes a system command in the SECURITY master dictionary called SECURITY-STATUS. This program can be used to enable or disable assembly code operations, enable or disable logon monitoring, and enable lines disabled because of logon violations.

Figure A is a checklist for the System Manager to use to determine how to respond to the issue of system security. Figure B shows a sample display of the SECURITY-STATUS command and its menu.

To select an option, enter the number; then make the desired change on that line. After each change, you are returned to the prompt. To exit the menu without storing any changes, enter 'EX'. To store the changes and exit, enter 'FI'.

NOTE: You may make selections by entering a slash command (/FI, /EX, or /line#) at any line other than the prompt.

When assembly code modification capability is shut off, these operations can be performed only from the SECURITY account (or SYSPROG, if enabled). The shutoff includes all Assembly Debugger functions except the basic functions (END, OFF, etc.) and the DUMP, MLOAD, and :ABSLOAD commands.

Logon monitoring allows the SECURITY account to keep track of the number of logon and LOGTO attempts that failed at a given terminal. With no logon monitoring, no line is ever disabled because of excessive failed attempts to logon/LOGTO. When monitoring is enabled: (1) a line may be disabled after a specified number of (sequential and unsuccessful) logon attempts for (2) a specified amount of time; (3) a line may be disabled after a specified number of unsuccessful logon attempts per day for (4) a specified amount of time. A default value may be assigned via SECURITY-STATUS for each of these four parameters, applicable to all lines. (The TERMINAL command can specify different values for a specific terminal, which will override the system defaults.)

When a line has been disabled as a result of the logon monitoring described above, it is put to "sleep" (that is, a SLEEP command is executed) for a specified period of time. The SECURITY-STATUS command can be used to reactivate a terminal before its sleep-time has elapsed. This restores the user's ability to logon from that line.

SECURITY-STATUS is also used to control whether the SYSPROG account is to be given the ability to run the same special security related programs that can be run from SECURITY, and at the same capacity, such as testing for unlimited file access. The default status is enabled, the same as SECURITY.

1. Should assembly code modifications be enabled (supported) on this system?

2. Should the SYSPROG account have the same privileges and access as the SECURITY account?

3. Should the system record and report failed logon attempts as a security control measure?

4. Do you need to re-activate a line that has been disabled because of failed logon attempts?

Figure A. System Security Checklist

```
                 Security Feature Status Control

(1)  Assembly code modification capability (E or D): Enabled

(2)  Enable Security Features in SYSPROG   (E or D): Enabled

(3)  Logon Error Recording and Control     (E or D): Enabled

(4)  No. of Sequential Logon Errors Before Disablement (#):10

(5)  Duration of Disablement (HH:MM:SS): 00:02:00

(6)  No. of Logon Errors Per Day Before Disablement (#):20

(7)  Duration of Disablement (HH:MM:SS): 00:05:00

(8)  Reactivate Line Disabled Due to Logon Failures (#):


                    Enter Option (#, EX, FI):
```

Figure B. SECURITY-STATUS Sample Display

4.4 SETTING UP TERMINAL SECURITY (TERMINAL)

The Terminal Security option (option 3) on the SECURITY menu invokes a system command in the SECURITY master dictionary called TERMINAL. This program can be used to enable or disable logon monitoring for an individual line, and to vary the monitoring parameters. It can also be used to specify characteristics of individual ports such as the baud rate, parity, terminal type, etc. These characteristics can be stored as current parameters for the line or as Logon parameters, presently stored in DICT ACC.

Figure A is a checklist for the System Manager to use to determine how to respond to the issue of terminal security. Figure B shows a sample display of the TERMINAL command and its menu.

Note that from SECURITY (or SYSPROG, when equal) any line may be specified. From other accounts, only the line logged on to that account may be specified, and only parameters 1-16 can be altered.

On the menu, lines 1-7 display the communications parameters that are currently in effect for a selected line. These parameters may be changed (same as SET-BAUD command).

On the menu, lines 8-15 display the default system-wide communications parameters that are put into effect every time the logon message is displayed. Line 15 is the same as the TERM command, and the complete options string (page width, page depth, line skip, line feed delay, form feed delay, back space character, printer page width, printer page depth, and terminal type) may be specified.

Line 16 gives the location that will be displayed with a LISTU or LISTUSERS command.

Lines 17-21 will override the parameters set in the SECURITY-STATUS command. These parameters can only be altered from SECURITY or SYSPROG (if equal to SECURITY).

1. Does an individual terminal line need to be set up differently from the system default logon parameters?

2. Does the system-wide logon security control need to be overridden for an individual line?

Figure A. Terminal Security Questions

```
                 Ultimate Terminal Processor

Line Number: 002

Current Parameters                Logon Parameters
   1. Baud Rate:  9600                8. Baud Rate: 9600
   2. Char. Length (5,6,7,8): 8       9. Char. Length (5,6,7,8): 8
   3. Stop Bits (1,2): 2             10. Stop Bits (1,2): 2
   4. Parity (EVEN,ODD,NONE): EVEN 11. Parity (EVEN,ODD,NNE) EVEN
   5. Echo (ON,OFF): ON              12. Echo (ON,OFF): ON
   6. Xon/Xoff (ON,OFF): ON          13. Xon/Xoff (ON,OFF): ON
   7. Type Ahead (ON,OFF): ON        14. Type Ahead (ON,OFF): ON
                                     15. Terminal Type: C
                                     16. Loc: Front Office

Logon Security
  17. Logon Attempt Restrictions (ON,OFF): ON
  18. Allowable Attempts per Session: 3
  19. Disable Time for Session Violation (HH:MM:SS): 00:30:00
  20. Allowable Attempts per Day: 5
  21. Disable Time for Day Violation (HH:MM:SS): 02:00:00

          Enter Option (#, EX, FI) :
```

Figure A. TERMINAL Sample Display

4.5 CONSIDERATIONS FOR SETTING UP NEW USER ACCOUNTS

A new user account is created via the CREATE-ACCOUNT command or the CREATE/UPDATE ACCOUNT option on the SECURITY menu. Before actually creating the account, the System Manager should take into consideration the user account's purpose, privilege and security requirements, and terminal lines at which the account needs to be accessible.

A security system protects the user data base from access or update by outside intruders. It also helps prevent a user with access to certain accounts from accessing other, unauthorized accounts. The Ultimate security system is controlled from the SECURITY account (or SYSPROG, if equal).

Control of user account and file access is based on the account names and security codes stored in both the account's user identification item (in the SYSTEM dictionary) and the individual file definition items within the account.

Below is a list of questions the System Manager should answer about each new account <u>before</u> actually creating the account.

1. What is the purpose of this user account? Will it require access to the Word Processing (WP) account, which uses the UltiWord software? Will it be using BASIC, the Editor, Recall commands, etc.?

2. What is the expected size of this account: the number of items? The size of each item?

3. Does the account need one or more passwords to protect access to its menus, commands, and data?

4. What privileges does this account require in regard to using tapes or disks, updating master dictionaries, using commands that affect another line, etc.?

5. Should any line be able to access this account? Should any be specifically denied access?

6. Does the account need to protect file access and/or update of its account dictionary and data files?

7. Should an accounting history of time logged on and line printer pages generated be kept?

8. Should a user automatically be restarted at the account's logon proc after exiting the Debugger?

9. Should the BREAK key be inhibited at logon?

10. Should this account be file-saved? Under all or special conditions?

11. Will this account be used to write any assembly language programs? If so, a TSYM file must be created so that the variables can be stored.

12. Will this account be used for data base retrieval (Recall commands such as SELECT, SORT, LIST, etc.)? If any select lists are saved, a POINTER-FILE dictionary must be created so that the account can access the lists saved by SAVE-LIST.

13. At logon, should this account be at the system (TCL) level or should a special logon proc be defined in the account's master dictionary? If so, the desired menu should be set up in a proc, unless the account is set up to access the Word Processing (WP) logon proc. Any special messages to be generated at logon are also stored in the logon proc.

4.6 CREATING NEW USER ACCOUNTS AND SYNONYM ACCOUNTS

The CREATE-ACCOUNT command creates new user accounts. This command may be entered at the system (TCL) level from the SECURITY or SYSPROG accounts, or from the SECURITY menu.

The CREATE-ACCOUNT command displays a menu for creating a new account and its criteria (see Figure A). This command can be used to create regular (D item) accounts and synonym (Q item) accounts.

If you enter an account name that does not currently exist in the SYSTEM dictionary, the screen displays a prompt for the type of account you wish to create:

```
Account Name (<CR> to exit): NEWACCOUNT
Defining (A)ccount or (S)ynonym:
```

A synonym account merely means that the new user account name will "equate to" another account name already on the system. Logging onto a synonym account is interpreted "as if" you logged on to the existing account, but the logon proc or program executed (if any) will be the one for the synonym account. Synonym accounts have fewer criteria, or "parameters", than regular accounts, since they have no data of their own. You may, however, assign a special password, assign a privilege level, and logon line(s). The account also has the account history option, restart option, and BREAK key inhibit option.

A regular account is a user account that will contain a master dictionary and data files. Items in the master dictionary (including data file pointers) will be stored on disk space as determined by a specified modulo and separation, which defines the amount of primary filespace allocated to the account. The master dictionary can have file access and update locks to prevent other accounts from using it. The account's files can be file-saved in several ways, as needed.

<u>Retrieval Locks and Update Locks</u>

Both file definition ("D" code) and synonym file definition ("Q" code) items have L/RET (retrieval locks) and L/UPD (update locks) attributes. When these attributes have values stored, they are known as security codes. Any number of account names can be stored as values in these attributes.

Retrieval locks are checked for most system commands that access file data. BASIC checks the lock when an Open command is executed. Update locks are checked for COPY, EDIT, BASIC write commands, and the Assembler.

The exact account name must match in the Retrieval Code attribute for the user attempting to retrieve a file in another account. In order for that user to update the file, the account name must match in the Update Code attribute for the account where the file is currently located.

If a user attempts to access or update a file to which his account has not been given access (it is not on the "Lock list"), an error message is displayed:

[210] File xxx is access protected.

<u>How modulo and separation affect file/account size</u>

When a <u>file</u> is created, a contiguous block of disk space is reserved. This block contains (modulo * separation) frames and is called the file's "primary" filespace. The "separation" is the number of frames per group ("bucket") of the file. If the separation is one (1), the typical and default condition, then each group will consist of one initial frame, so up to 500 bytes will be the maximum that an item could contain in order to fit into one group. This would be reflected in the ITEM and GROUP commands.

The "modulo" is the number of groups a file contains in primary filespace. If the modulo is one (1), the default condition, then the file will have one group. This will be reflected in the ITEM and GROUP commands. (These commands are discussed in the System Commands Guide.)

As data is placed in each group, the group may overflow by linking on additional disk frames as needed. Although a file may expand to any length (within the limits of the disk space), a group should be kept as small as possible (that is, 1 frame per group) for efficient retrieval time. Also, the modulo should follow this guideline: the larger the average item-size, the smaller the number of items in a group. This may be achieved by using the "automatic optimum calculation" option for the modulo (in the CREATE- or UPDATE-FILE menu).

NOTE: When an <u>account</u> is created, the default modulo and separation are 29,1 (that is, 29 groups of 1 frame each are reserved in a contiguous block of disk space).

<u>Allocating the New Account (NEWAC) Files to a new account</u>

After the account's specifications have been decided, entered, and checked, file the account information. This creates a D pointer item in the SYSTEM dictionary.

CREATE-ACCOUNT then copies the contents of the NEWAC file (the prototype master dictionary) to the new user account's master dictionary. Finally it adds a file synonym (Q item) to the account into SYSPROG's master dictionary.

For details on entering the account parameters, see Section 4.7, CREATING OR UPDATING AN ACCOUNT'S PARAMETERS.

```
                   ULTIMATE Account Processor

            Account Name: SAMPLE (Existing)
            Modulo, Separation: 61,1

Dimensions:
     1. Reallocation (Modulo, Separation) : 73,1

Security:
     2. Password(s): DEMO CPU
     3. System Privileges (0,1,2): 2
     4. Lines to Allow Logon: 0,1,5,16,29-36,39
     5. Retrieval Locks (Y/N): Y
     6. Update Locks (Y/N): Y

Special Options:
     7. Accounting Option (Y,N): Y
     8. Restart Option (Y,N): N
     9. Inhibit Break Key at Logon (Y/N): Y
    10. File-Save Options (V,W,X,Y): W

            Enter Option (#,EX,FI):
```

Figure A. CREATE-ACCOUNT Sample Display

4.7 CREATING OR UPDATING AN ACCOUNT'S PARAMETERS

The Create/Update account option (option 3) on the SECURITY menu invokes the CREATE-ACCOUNT system command or if the account already exists, its synonym UPDATE-ACCOUNT. This program can be used to modify account parameters, such as passwords and valid logon lines, for any account on the system. It also creates and updates synonym (Q pointer item) accounts.

From the SECURITY and SYSPROG (if equal) accounts, any other account can be created or modified. However, the security parameters may be modified only from the SECURITY account. From a user account, only that account may be displayed and modified.

NOTE: Read Section 4.5, CONSIDERATIONS FOR SETTING UP NEW USER ACCOUNTS, before entering responses to the items on the CREATE-ACCOUNT menu.

Figure A shows a sample of the CREATE-ACCOUNT display. Data entry follows the same rules as for SECURITY-STATUS. To select an option, enter the number; then make the desired change on that line. After each change, you are returned to the prompt. To exit the menu without storing any changes, enter 'EX'. To store the changes and exit, enter 'FI'.

NOTE: You may make selections by entering a slash command (/FI, /EX, or /line#) at any numbered line.

The first prompt is for the account name. If the entered name does not exist in the SYSTEM dictionary, the program adds "(New)" and prompts for "(A)ccount or (S)ynonym". Selecting A creates a new account by adding a D pointer item to the SYSTEM dictionary. Selecting S creates a synonym by adding a Q pointer item to the SYSTEM dictionary. The default values for account definition items are displayed, and the values to be used for this account may be specified, as desired.

If the entered name already exists in the SYSTEM dictionary, the program adds "(Existing)". If the account is a regular (D item) account, the current modulo and separation of the account's master dictionary are displayed. If a synonym (Q item) account, the current linkage (the account the Q pointer points to) is displayed. The linked account's account definition values are displayed, and the values to be used for this account may be specified, as desired.

The modulo and separation values are specified in new accounts only; the defaults are 29,1. This means that the account Master Dictionary (MD) is allocated 29 groups of 1 frame each as primary space. If the displayed modulo and separation is not adequate for the account, you may specify different (if a new account) or new (if an existing account) values on line 1, as "reallocation" parameters. Both values, separated by a comma, must be entered. (This means, reallocate the MD to the new modulo and separation at the next File (or Account) Restore procedure.)

Passwords are entered or changed on line 2. Password(s) may be viewed and changed only from the CREATE/UPDATE ACCOUNT program. A space is used to separate multiple passwords assigned to the same account. Only one password is needed to logon.

User accounts are assigned the system privilege level 0 (SYS0) by default, on line 3. The privilege level may be increased only from the SECURITY account (or SYSPROG if equal). The level may be lowered from any account. NOTE: SYS2 privileges are required to execute any command that affects another line, such as SET-BAUD, BREAK-KEY-ON, and :RESTARTLINE n, as well as the File Save and File Restore procedures. If assembly code operations have not been disabled globally (see SECURITY-STATUS above), then SYS2 privileges are required to use the Assembly Debugger, the DUMP and LOAD commands, and the Assembler. SYS1 privileges are required to use the magnetic tape or disk cartridge, as well as to update an account's master dictionary from within the account.

Line 4 specifies which lines (terminal ports) in the system may logon to the account. If no line is specified, then all lines may be used to access the account. Any entry restricts access to the specified line(s). A range of lines may be specified (for example, 5-8). Individual entries are delimited by commas. If access should be given to all but a few lines, you may specify access as:

    NOT(line#),NOT(line#,....

NOTE: For Honeywell systems only: If the Off-Line Monitor is to be used for disk to disk copy operations, then the Warmstart line must be included in the SYSPROG account's logon lines. The Warmstart line number is one less than the Spooler's line number, or three more than the highest numbered physical line (asynchronous communications channel).

Beginning with the Release 10 Rev 130 (R10*130) operating system, retrieval and update locks have a different meaning and content than in previous releases. It is recommended that all old mnemonic format locks be removed prior to installing R10*130.

Retrieval and update locks are applicable only to regular (D-item) accounts, not synonym (Q-item) accounts since they are only synonyms for another regular account. All regular accounts with synonyms, however, must specify the Q account name in their lock codes.

If no locks are specified, any other account may access or update the master dictionary of the account. The lock codes are made up of the System Name (for UltiNet users) and the Account Name; only those names specified may access or update the master dictionaries of these accounts. (At the file level, the locks specify which accounts may access or update that particular file. Locks must be put on each file that should have restricted access.)

To specify locks, enter "Y". Another menu appears on the screen; an example is shown in Figure B. You may use the screen to add or delete accounts from the "authorized access or update" list; select "I" to insert or "D" to delete. At the prompt for system name, enter a period (.) for accounts in the local system, or the actual UltiNet System name. Then enter the account name.

The accounting option on line 7 (for regular accounts) or line 5 (for synonym accounts) specifies that logon usage statistics are kept in the Accounting History (ACC) file. This value may be changed only from the SECURITY account (or SYSPROG, if equal).

The restart option specifies that, when restarting from the Assembly Debugger (by typing "END"), the account will automatically execute the logon proc or program stored in the account's master dictionary. (This proc or program has the same name as the account name.) This value may be changed only from the SECURITY account (or SYSPROG, if equal).

The BREAK key inhibit option specifies that the BREAK key is to become inoperative at logon time for this account. In the inhibited state, pressing BREAK will not abort the current Process operation <u>until a BREAK-KEY-ON command occurs.</u> This value may be changed only from the SECURITY account (or SYSPROG, if equal).

The File Save options (applicable only to regular D accounts) are V, W, X, and Y. Only one option may be selected, as follows:

- V — Account is file-saved only if the SAVE command contains the 'V' option.
- W — Account is file-saved only if the SAVE command <u>does not</u> contain the 'W' option.
- X — Account is never file-saved.
- Y — Account is file-saved, but no master dictionary items are saved except D-pointers. The account will be created at File Restore time without any verbs, etc.
- (null) — Account is always file-saved.

When the values have all been entered and checked, the account is created or updated when you enter "FI" at the prompt (or /FI at any line). For regular (D item) accounts, an account definition item is created/updated with attributes from 1-14. For synonym (Q item) accounts, attributes 1-4 are used. Figure C shows the format of a D item and Q item.

NOTE: The update and retrieval lock codes are stored in the following format:

System-name \ Account-name] System-name \ Account-name]...

For example a null value would indicate no locks. A "\SYSPROG" entry would indicate only SYSPROG on the local system. A "BLUE\SYSPROG\DEANER" entry would indicate SYSPROG and DEANER accounts on the BLUE system. A "GREEN\\" entry would indicate all accounts on the GREEN system. An entry of "BLUE\DEANER]GREEN\CJM" would give access to DEANER on BLUE and CJM on GREEN.

All items from SYSPROG's NEWAC file are copied to the account's master dictionary (for new accounts only).

```
                  ULTIMATE Account Processor

          Account Name: SAMPLE (Existing)
          Modulo, Separation: 61,1

Dimensions:
     1. Reallocation (Modulo, Separation) : 73,1

Security:
     2. Password(s): DEMO CPU
     3. System Privileges (0,1,2): 2
     4. Lines to Allow Logon: 0,1,5,16,29-36,39
     5. Retrieval Locks (Y/N): Y
     6. Update Locks (Y/N): Y

Special Options:
     7. Accounting Option (Y,N): Y
     8. Restart Option (Y,N): N
     9. Inhibit Break Key at Logon (Y/N): Y
    10. File-Save Options (V,W,X,Y): W

          Enter Option (#,EX,FI):
```

Figure A. CREATE-ACCOUNT (and UPDATE-ACCOUNT) Sample Display

```
     Account Name: NEWACCOUNT (New)
     Retrieval Locks
     Function = Insert

                         Ultimate System Security

System Name     Account Name       System Name     Account Name

(Local)          NEWACCOUNT
                 DONNA
                 ULTIMATE
                 SUPPORT
BLUE             TRANSFERS
                 DROP-SHIP
                 ORDERS


     Enter (I)nsert or (D)elete:
```

Figure B. Retrieval and Update Lock Sample Display

Attr#	Name	Account Item	Synonym Item
1 (Req)	D/CODE	D,DV,DW,DX,DY (Line 10, File Save)	Q
2 (Req)	F/BASE	Base FID of account	Account name
3 (Req. if D)	F/MOD	Modulo of file	
4 (Req. if D)	F/SEP	Separation of file	
5	L/RET	Accounts able to access MD (Line 5, Retrieval Locks)	
6	L/UPD	Accounts able to update MD (Line 6, Update Locks)	
7		Password(s) (Line 2)	
8		Privilege Level (Line 3)	
9		Accounting (ACC) (Line 7), Restart (Line 8), Inhibit BREAK (Line 9)	
10		(Not used)	
11		(Reserved)	
12		(Reserved)	
13	F/REALLOC	Reallocation parameters (Line 1, Modulo and Separation)	
14		Logon Lines (Line 4)	

Figure C. Account Definition Item Formats

4.8 CREATING A POINTER-FILE DICTIONARY

An account that will be using any Recall commands should have a POINTER-FILE defined for it. A POINTER-FILE is required in order for an account to access stored select lists (which were saved by the SAVE-LIST command). The POINTER-FILE items are pointers to the stored lists.

The items in the POINTER-FILE may be examined, but, like file-pointers, are protected from modification by users. The format of these items is:

Item-id		Item-name used in SAVE-LIST command.
001	CL	CL for lists.
002	fid	Base FID of the list.
003	n	Number of frames in the list.
004	m	Number of items in the list.
005	time & date	Time and date of generation.

Each account or group of accounts may be allocated a POINTER-FILE when the account is set up by the System Manager.

NOTE: Since the account name generating a list is not stored as part of the list name, the default list name '' (null) should not be used if more than one account uses the same POINTER-FILE (via Q-pointers).

It is generally better to provide each account with its own POINTER-FILE. The name POINTER-FILE is reserved and known to the select-list software. It is therefore possible, though not recommended, that POINTER-FILE could be constructed as a Q-pointer to another file, and could be changed to point to different files for different tasks which may be executed on the account. The POINTER-FILE software may reference only one pointer file at a time, however, and all processes logged onto a particular account will reference the same POINTER-FILE.

4.9 SETTING UP A WORD PROCESSING (WP) ACCOUNT

Many user accounts are primarily designed for word processing functions. These accounts can be set up as Word Processing accounts, with a special UltiWord menu as the Logon proc, and with several special files used only for UltiWord operations. When an account becomes a word processing user account, all UltiWord features are available whenever that account is logged on.

In order to use UltiWord and the Word Processing (WP) account, the WP account must be already loaded onto the Ultimate system. The WP account may be loaded from the SYSPROG menu (or system commands that do four T-FWDs and an ACCOUNT-RESTORE of WP).

When the "Load WP account from SYS-GEN tape" selection is made from the SYSPROG menu, the system displays:

```
Please mount the latest SYS-GEN tape on the tape drive
Press <CR> when ready.
```

The system then rewinds the tape and attempts to position the tape at the account WP. If the account cannot be found, an end-of-file message is displayed.

When the WP account is found, the associated files are listed as they are loaded. For example:

```
WP
     WP 15638,31,1
      WP-PROGS 15474,7,1
       WP-PROGS 15669,17,1
       TABLES 15686,3,1
       USER 15689,31,1
      WP-DOCUMENTS 15789,29,1

Press <CR> to continue-
```

When you press <CR>, the system level (TCL) prompt is displayed (>). You may then enter any system command.

Setting up a user account for UltiWord

Once the WP account is loaded onto the system, you may set up individual user accounts as "Word Processing" accounts. This allows the user account to use the UltiWord software.

To set up a user account for UltiWord, from the TCL level you must logto the WP account:

```
>LOGTO WP
```

The password, if requested, is then entered. The WP account automatically displays the UltiWord Main Menu (see Figure A).

Select the (S)et up user account option from the Main Menu by entering "S" (upper or lower case).

The system displays the prompt:

    Enter name of account to set up or <EDIT> to quit

Enter the account name that the user will logon to for word processing tasks. Do not enter "WP" as the account name; use only a unique user account name.

The account name is verified; it must already exist in the system (see the Create Account topics in this manual).

To be used for word processing, each account must contain several word processing verbs and one proc. A check is made to see if any of these items to be copied to the account already exist on the account. If they do, the item names are shown and a message is displayed:

    These will be destroyed if Y is entered, (Y)es or <EDIT>

Press <EDIT> to return to the Main Menu without erasing the items. (NOTE: on all terminals the <CTRL/E> keys will perform the <EDIT> function.) Type "Y" to copy the needed information to enable the current UltiWord functions into the account and overlay the existing items. The Main Menu is re-displayed.

NOTES: If the needed word processing verbs or proc are not already in the account, they will be created.

If a WP-DOCUMENTS file already exists in the account, the file and all documents (items) within it will be preserved. Otherwise it may be created after logging to the account, or created automatically by logging to the account and using UltiWord (WP command) the first time.

```
***** The Ultimate Corp.*****
*****   WORD PROCESSING *****
*****  Document Manager *****
            Main Menu

YOUR USER NAME:
DOCUMENT NAME:

(C)opy documents
(D)elete user
(E)dit/Enter
(L)ist
(N)ew
(P)rint
(S)et up user account
(U)tility
(V)iew
e(X)it

     Enter option wanted
```

Figure A. WP Account (UltiWord) Main Menu

4.10 DELETING A USER ACCOUNT (DELETE-ACCOUNT)

The DELETE-ACCOUNT system command deletes an account and all its files from an Ultimate system.

In order to run DELETE-ACCOUNT, you must be logged on to SECURITY (or SYSPROG, if it has SECURITY privileges).

Deleting an account produces a listing of all files being deleted. The listing may be displayed on the terminal or printed on the printer.

If the account name is not found, or cannot be accessed, the message:

    The account name is not valid, or is access-protected.

is displayed on the terminal, and control returns to the system (TCL) level.

See Figure A for sample usage and resultant listing.

```
>DELETE-ACCOUNT       PROC name is typed

ACCOUNT NAME ?SHERRY       at TCL.

FILES TO BE DELETED IN ACCOUNT SHERRY     02 APR 78      PAGE
  1

FILE           BASE     MOD  SEP

MD             34593    37   1
GEN/LED        85344    1    1
GEN/LED        49911    231  1
BP             44319    17   5

DO YOU STILL WANT TO DELETE THE ACCOUNT ?YES
                                       (Must start with 'Y')
```

Figure A. Sample DELETE-ACCOUNT Display

4.11 CREATING NEW FILES

User files are created within user accounts. A file may be created by the CREATE-FILE command. This command sets up the file definition (D) item in the account's master dictionary. The CREATE-FILE command has a synonym, UPDATE-FILE, which may be used to change the file's parameters.

The CREATE-FILE command may be entered with the file name and all parameters following the verb in a complete system (TCL) command. Or, just the CREATE-FILE verb, optionally followed by a filename, may be entered. This method calls up a menu for data entry of each parameter separately, from prompts.

The initial menu display is shown in Figure A. This display is skipped if a filename is entered.

Once you have specified the file and portion (dictionary and/or data) to be created (or updated), another menu appears (see Figure B).

If the file already exists, the current modulo and separation are given next to "Dictionary Section" and "Data Section"; for example:

    Dictionary Section (Modulo, Separation = 1,1)

and lines 2 and 9 read:

    2. Reallocation (Modulo,Separation):

You are prompted for each field. Entering <CR> alone retains the displayed default value. An operator "helps" message appears at the bottom of the screen for some fields, such as:

(1) Enter SPACE to delete field

(2) Enter (AC) for automatic optimum calculation
Enter estimated number of items:
Enter estimated number of bytes per item:

You should be familiar with the security codes and lock concepts before entering Retrieval and Update Locks. These Lock fields have the same criteria as in creating an account. (See the Create Account topics in this section.)

When all parameters have been entered, you may change the displayed value(s) by entering the field number. To exit the menu without storing any values, enter "EX". To file the information and create the account as specified, enter "FI". The command returns to the initial display for creation of another file or <CR> to exit to the system (TCL) level.

NOTE: An account using the Assembler must have a TSYM file defined. An account using Recall commands should have a POINTER-FILE defined.

A Note about filespace

The CREATE-FILE software allocates a new file's primary filespace to contiguous frames only, not linked frames. This means that the Available Space pool must contain many contiguous frames as the total primary filespace requires (modulo times separation, or number of buckets times number of frames per bucket in the file). The system, therefore, must have at least this many contiguous frames available. The POVF command displays the current Available Space frames. (See the POVF command in the Troubleshooting section or in the System Commands Guide.)

For details on CREATE-FILE, see the System Commands Guide.

```
                  ULTIMATE File Processor

File Name (<CR> to exit):


Valid File Name Formats:

file-name
file-name,file-name
DICT file-name
DATA file-name
DATA file-name,file-name
```

Figure A. CREATE-FILE Initial Display

```
File Name = NEWFILE (New)
Dictionary Section
     1. File Options (V,W,X,Y):
     2. Modulo,Separation: 1,1
     3. Retrieval Locks (Y/N): N
     4. Update Locks (Y/N): N
     5. Conversions:
     6. Correlatives:
     7. Justification (L,R,T,U): L
     8. Length: 10

Data Section
     9. File Options (V,W,X,Y):
    10. Modulo,Separation: 1,1
    11. Retrieval Locks (Y/N): N
    12. Update Locks (Y/N): N
    13. Conversions:
    14. Correlatives:
    15. Justification (L,R,T,U): L
    16. Length: 10
```

Figure B. CREATE-FILE Menu

4.12 OVERVIEW OF COMMON PROCEDURES FOR USER ACCOUNTS

A number of system commands are available for procedures often required by a user account.

From the system (TCL) level, a user account can request a number of commands that perform commonly used procedures or display useful information. Figure A contains a listing of such commands.

For details about these commands, and any other system (TCL) command, see the System Commands Guide. Each command is alphabetically listed and fully described.

COMMAND NAME	OPERATION
BLOCK-PRINT	Sends block characters to terminal or Spooler
CHARGES	Displays current user's computer usage
CHARGE-TO	Ends current session and starts logging time to a specified account or name
CLEAR-FILE	Removes all file items from a file or dictionary, but retains the filename
COFF	Logs off system and communications line
COPY	Copies contents of files and dictionaries
CT	Displays items on terminal
DATE	Displays the time and date
DELETE-FILE	Deletes an entire file and all items
DELETE	Deletes an item in a file
LISTF	Lists D-pointer items, not synonyms (Q-pointer items)
LISTFILES	Lists names of files in an account
LOGTO	Logs off current account and onto another
MESSAGE	Sends a message to other users
MSG	Same as MESSAGE
OFF	Terminates a user session (logs off)
SP-ASSIGN	Assigns output to specified Spooler peripheral equipment (printer,tape)
SP-STATUS	Displays Spooler status
TABS	Sets tabs for input or output
T-ATT	Attaches a tape unit
T-DET	Detaches a tape unit
TERM	Displays or sets terminal characteristics
TIME	Displays the time and date
WHO	Displays the line number and account name to which any terminal is logged on

Figure A. Commonly Used System Commands

4.13 THE ACCOUNTING HISTORY FILE: PERIODIC CLEARING

> To avoid overflowing the accounting history items in the Accounting History (ACC) file for a specific user, the items should be periodically cleared.

To clear the accounting history items from the ACC file, follow the steps detailed in Figure A.

The point of overflow on a particular port (line) is determined by the activity of the user account (however, approximately 1000 Logon/Logoffs are allowed). This point can be calculated by following the procedure detailed in Figure B.

At logoff, if the accounting history item for a user account/port number combination exceeds 32265 bytes in size, the user will be logged off, but the Accounting History File will not be updated. To recover from this situation, follow the procedure detailed in Figure A.

1. Logon to the SECURITY (or SYSPROG, if enabled) account.

2. Type the following (if you need a listing only):

   >**SORT ACC WITH NAME LPTR <CR>** lists logged-on users

3. Type the following:

   >**SELECT ACC WITH NO NAME <CR>** selects only updated items

   >**DELETE ACC <CR>**

Figure A. Procedure to Clear all Accounting History Items

1. To determine the current size, type:

   >**LIST ACC 'user-name#port-number' 9999 <CR>**
   (EXAMPLE: >LIST ACC 'DW#2' 9999)

   This will produce the following output:

   ```
   PAGE 1                          Time date
   ACC........9999......
   User-name  xxxx
   ```

   If the value displayed for 9999 (i.e., xxxx) approaches 32265, then the user account is approaching the overflow point.

Figure B. Overflow Point for ACC Item

4.14 SETTING UP NEW LINES (PORTS) AND CHARACTERISTICS

When a new terminal line (port) is connected to an Ultimate system line (port), the System Manager should define the characteristics for that port and the type of terminal being used. Port parameters can be specified at logon time for each port via the TERMINAL command.

The Ultimate system supports a variety of different terminals which can be used for operator/user interaction with the system. When a new user line (port) and/or terminal is installed, the System Manager is responsible for selecting the terminal control and security appropriate to that line and terminal, and the users who will be logging onto Ultimate from it.

The main "system level" of Ultimate operations is called the Terminal Control Level (TCL). The system (TCL) level prompt is the greater-than sign:

    >

At the prompt, users request processing to be performed by using a Terminal Control Language vocabulary of "verbs". These verbs are used to construct system level (TCL) commands, sometimes called "sentences" or "statements."

An Ultimate system is supplied with over 200 standard system commands, plus any commands associated with custom or optional software installed with the system. The standard system commands are alphabetically listed and fully detailed in the Ultimate System Commands Guide.

An Ultimate system is also "menu-driven" in that a number of accounts, such as SYSPROG and SECURITY, have an associated menu of task options. These menus are designed to help users to easily request and perform commonly used procedures.

When a new terminal is installed on a line (port), the System Manager should review the requirements for that terminal line and use the appropriate commands to set up the initial conditions:

BREAK-KEY-ON or BREAK-KEY-OFF
ECHO-ON or ECHO-OFF
SET-BAUD
TYPEAHEAD-ON or TYPEAHEAD-OFF
X-ON or X-OFF
TERMINAL

The TERMINAL command is the same as selecting the Terminal Security option from the SECURITY account menu. This command has a menu that allows you to set up the echo, typeahead, X-on or X-off, baud rate values, and logon security values all at one time. The BREAK key condition at logon-time is associated with a user account rather than the physical terminal; it is set up via the CREATE-ACCOUNT command. (See Section 4.7, CREATING OR UPDATING AN ACCOUNT'S PARAMETERS.)

The terminal line conditions set up via TERMINAL are actually stored in items in DICT ACC: item-ids are 3-digit port numbers, line 1 can be used for a location description, line 2 for SET-BAUD parameters, and line 3 for TERM parameters. For example:

```
     023
001  MODEM
002  1200,8,NONE,OFF,OFF
003  V
```

To set and view the terminal page width, depth, type, and other characteristics, use the TERM command (listed alphabetically in the System Commands Guide).

4.15 DEFINING A TERMINAL

The TERMDEF file on the SYSPROG account is used to define terminal and auxiliary port characteristics for various types of terminals. This allows programs to work with any defined terminal or aux port device through a standard interface, without having to inspect input or generate output specific to any one device.

The Ultimate BASIC software supports a number of terminals and their cursor control firmware. Cursor control in BASIC is accomplished via the @ function (the @ function is explained in the BASIC Reference manual).

However, before the BASIC software is used, the System Manager or terminal user should define each terminal type via the TERM command or SET-TERM command. (See the TERM and SET-TERM commands, listed alphabetically in the System Commands Guide.)

The Ultimate SYS-GEN tape includes support for a standard set of terminal and printer types, including the following:

Code	Type
	Terminals:
A	ADDS Regent 40 (25 line CRT)
C	ADDS Viewpoint (Color)
D	DEC VT100
H	Honeywell VIP-7200
L	Liberty Freedom-200
P	IBM Personal Computer
R	ADDS Regent 25
U	Ultimate (Volker-Craig)
V	ADDS Viewpoint
W	Wyse WY-50 (in enhanced Viewpoint emulation mode) or Ultimate ULT-50
Y	Wyse WY-85 in VT220 7-bit mode
	Printers:
H	Honeywell (NEC) Printer (slave printer)
L	HP LaserJet Printer

A terminal type code may be specified in a TERM or SET-TERM command to indicate the type of terminal connected to a port; for example:

```
>TERM R
```

It may also be specified as a logon parameter for a port using the TERMINAL command.

A printer type code may be specified in a PRINTER command for output generated on a port; for example:

```
>PRINTER L
```

Most output is generated regardless of printer type, however, UltiWord being an exception.

Users may add items to the TERMDEF file to support non-standard terminals. Whenever the TERMDEF file is changed, the LOAD-TERMDEF program in SYSPROG-PL must be run to load the new definitions into an area for use by the system terminal I/O routines.

Each item in TERMDEF defines one terminal type or auxiliary port device type (typically a slave printer). The item consists of a series of specifications, plus optional comments. Each specification or comment takes one line. Blank or null lines are also treated as comments.

The various specification formats are described below. CODE is the first sequence of non-blank characters appearing in a line. PARAM is the second such sequence (i.e., PARAM is separated from CODE by one or more blanks). Within the PARAM column, "xx..." stands for a sequence of hexadecimal digits, in multiples of two. Within the CODE and Meaning columns, "n" stands for one or more decimal digits. Characters in brackets are optional.

Although specifications may appear in virtually any order, maintenance may be easier if they appear in the following standard order.

For an example, see the TERM.U item in the TERMDEF file.

CODE	PARAM	Meaning
*	(any)	Comment
T[ERM]	Alpha character	TERM type code; signals terminal definition
A[UX]	Alpha character	AUX type code; signals aux port device definition
XY	Uxxxx	Mode-id of cursor-positioning routine for terminals, or general routine for aux-port devices
-n	xx...	Character sequence to output for @(n) cursor function in BASIC or PROC
O[PTIONS]	xxxxxxxx	TERM-INFO option code (see below)
Fn	xx...	Character sequence generated by Function Key n
FnS	xx...	Character sequence generated by Function Key n shifted
CH[OME]	xx...	Character sequence generated by Cursor Home key
CL[EFT]	xx...	Character sequence generated by Cursor Left key
CU[P]	xx...	Character sequence generated by Cursor Up key
C[DOWN]	xx...	Character sequence generated by Cursor Down key
CR[IGHT]	xx...	Character sequence generated by Cursor Right key

For terminals, any unspecified @(n) function will produce a null result on output; any unspecified function or cursor key definition will return X'00' on a TERM-INFO call; and an unspecified TERM-INFO option word will default to X'00000000'. Undefined terminal or aux port devices will use these default values, along with default cursor and aux port handling routines.

The @(n) functions are defined as follows:

CRT Functions:

@(-1) = Clear screen	@(-2) = Home
@(-3) = Clear to end of screen	@(-4) = Clear to end of line
@(-5) = Start blink	@(-6) = Stop blink
@(-7) = Start low intensity	@(-8) = Stop low intensity
@(-9) = Cursor left	@(-10)= Cursor up
@(-11)= Cursor down	@(-12)= Cursor right
@(-13)= Slave port enable	@(-14)= Slave port disable
@(-15)= Slave enable transp.	@(-16)= Slave local print
@(-17)= Start underline	@(-18)= Stop underline
@(-19)= Start inverse video	@(-20)= Stop inverse video
@(-21)= Delete line	@(-22)= Insert line
@(-23)= Scroll up	@(-24)= Start boldface type
@(-25)= Stops boldface type	@(-26)= Delete one character
@(-27)= Insert one blank char.	@(-28)= Start insert char. mode
@(-29)= Stop insert char. mode	

Slave Printer Functions:

@(-101,P)	= Set VMI	@(-102,L)	= Set HMI
@(-103)	= Set alt. font	@(-104)	= Set std. font
@(-105)	= Half line-feed	@(-106)	= Neg. half line-feed
@(-107)	= Neg. line-feed	@(-108)	= Print black ink
@(-109)	= Print red ink	@(-110)	= Load cut sheet feeder
@(-111)	= Feeder1 select	@(-112)	= Feeder2 select
@(-113)	= Std thimble select	@(-114)	= P/s thimble select

TERM-INFO: Guidelines for Defining Terminal Keys

The following programming information can be used to define function and cursor control key characters, as needed. Also, see the Assembler Manual for more about Ultimate's assembly language capabilities.

TERM-INFO, defined in SYSPROG's PSYM file, is an assembly-language routine which returns a 32-bit flag word in D0 and 38 bytes of function and cursor control key definition characters. The 38 bytes are returned at R15 followed by an SM. However, if flag bit 0 (B31) is not set, indicating an undefined terminal type, no function or cursor key information is returned.

The flag bits are specified in the TERMDEF O[PTION] code (8 hex characters), and are returned in D0 on a TERM-INFO call, as follows:

Flag	Meaning	Flag	Meaning
0	Terminal is a CRT	1	Terminal has Aux Port
2	Programmable Aux Port	3	Transparent Aux Port
4	Discrete visual attrib.	5	Attrib. takes a space
6	Delete Line	7	Insert Line
8	Delete Character	9	Insert Character
10	No-Roll Status Line	11	Multi-byte Function Keys
12	Multi-byte Cursor Keys	13	<CR> Follows Function Key
14	Multi-byte Cursor Leadin	15	Multi-byte Function Leadin
16	Has reverse scroll		

Bits 17-31 are unused and available

The function and cursor key definition characters are returned by TERM-INFO in the following order:

Bytes	Definitions
0	Function Key Lead In Char (normally ESC; except STX for Viewpoint)
1-16	Function Keys 1 to 16 (2nd byte if flag bit 11 set)
17-32	Shifted Function Keys 1 to 16
33-37	Cursor Home, Left, Up, Down, Right (2nd byte if flag 12)

Routines named in XY mode-id specifications should conform to the following interface:

Input:	R15	Points to output area for generated string
	CTR10	Contains second parameter: screen row address for cursor positioning; absent parameter if less than zero; screen rows begin with zero
	CTR11	Contains first parameter--screen column address for cursor positioning; absent parameter if less than zero; screen columns begin with zero
	TH	Contains relative function number for aux port routines, equal to -(n-100) for @(-n), so T4=1 for @(-101), 2 for @(-102), etc.
Output:	R15	Points to last character of generated string (or unchanged for null value)

Below is an example of a XY cursor mode which can be put into any USER-MODES frame.

```
!ADD-CURSOR EQU *
            BLZ  CTR10,AC100   BRANCH IF NO LINE ADDRESS SPECIFIED
            MCI  X'0B',R15     SET UP 'VT' FOR ROW ADDRESS
            LOAD CTR10         GET LINE NUMBER
            DIV  24            REMAINDER (T2)= LINE ADDRESS, 0 TO 23
            MCI  H4,R15        SET UP LINE ADDRESS CODE
            SB   R15;B1        MAKE CHAR NON-CONTROL CODE
AC100       LOAD CTR11         GET COLUMN NUMBER
            BLZ  T0,XIT
            MCI  X'10',R15     SET UP 'DLE' FOR COLUMN ADDRESS
            DIV  80            REMAINDER (T2) = COL ADDR, 0 TO 79
            MOV  T2,CTR11      SAVE VALUE
            LOAD T2
            DIV  10
            MUL  6
            ADD  CTR11         VALUE=X/10*6+X (IS CORRECT BCD CODE)
            MCI  H0,R15        SET UP COLUMN ADDRESS CODE
XIT         MOV  R15,R14       PRESERVE THIS FRAME
RTN         RTN
            END
```

4.16 TERMINAL CONTROL FEATURES

The Ultimate system uses a typeahead feature to provide terminal input and output control. This feature works with the bell character, which alerts the user if the CPU is not ready to receive input. With typeahead enabled, any data entered is stored and fed to the CPU as it becomes ready.

All data input through a terminal port is read by the CPU as soon as possible, and is stored in an input buffer in memory before (or awaiting) processing. The size of the typeahead buffer for each terminal port varies by system.

Under certain extremely heavy system loads, the CPU may not be able to read stored input within its allowable "holding" time period. Normally, however, within the constraints of system load and the size of the typeahead buffer, no data is lost due to an executing program's use of system resources.

The <BREAK> key serves as a way to clear the type-ahead input buffer. When the <BREAK> key is pressed, the system clears all typeahead input, regardless of whether the <BREAK> key has been toggled "on" or "off" via the system commands BREAK-KEY-ON or BREAK-KEY-OFF. This means that if the System Assembly Debugger or BASIC Debugger has been entered via the <BREAK> key, any data previously typed will not be processed by the Debugger.

The typeahead function can be turned on or off on a per-terminal basis with the TYPEAHEAD-ON and TYPEAHEAD-OFF commands.

The Ultimate system also handles bidirectional X-on and X-off control; that is, both incoming and outgoing. If the user's terminal typeahead input buffer is almost full, the system automatically sends out an X-off character. When the buffer empties to the point where more data can be accepted, the system automatically sends an X-on character. Similarly, if a terminal or serial printer sends an X-off character to the computer, the system will not send any more data characters to the terminal or printer until the device sends an X-on character to the computer.

In addition for Honeywell system users, these commands are available for communications activities:

SET-LOGOFF sets automatic logoff when DSR drops
RESET-LOGOFF resets automatic logoff
READ-STATUS returns the communications status of port
DROP-DTR drops the Data Terminal Ready signal
RAISE-DTR raises the Data Terminal Ready signal
DROP-RTS drops the Request to Send signal
RAISE-RTS raises the Request to Send signal

All of the system commands mentioned above are alphabetically listed and fully described in the System Commands Guide.

4.17 USING MULTIPLE TCL LEVELS

Each terminal line may have stacked, multiple TCL "levels". The EXECUTE statement in BASIC is one way to access a secondary TCL level, in that it actually creates a secondary TCL level and executes the command in the secondary level. While the EXECUTE statement is executing the specified system command, the BASIC program at the previous TCL level is temporarily suspended. When the EXECUTE statement is finished, workspaces used by the secondary TCL level are released to secondary Overflow and control returns to the previous TCL level.

There is no limit on the number of secondary TCL levels created under any single terminal line. However, each secondary TCL level requires a set of workspaces of 416 contiguous frames. Care must be taken not to recursively EXECUTE the same BASIC program; otherwise the entire disk space will be used up.

NOTE: The WHERE command displays all TCL levels running terminal lines. (See Section 5 for examples of troubleshooting with the WHERE command; also see the System Commands Guide.)

The System Assembly Debugger may be used for troubleshooting if the user account has SYS2 privileges and <u>does</u> <u>not</u> have the 'R'estart option in the user identification item in the SYSTEM dictionary. (To check this, use the UPDATE-ACCOUNT command to view the user-id item.) Section 5 covers the System Debugger commands that relate to entering and exiting TCL levels (the ">", ">>", "<", "<<", and "END" commands).

To terminate a program being tested, the END command in the BASIC Debugger and the System Assembly Debugger will abort it and release all TCL levels.

4.18 PROCEDURES FOR COMMUNICATIONS BETWEEN ULTIMATE SYSTEMS

A user can transmit or receive data from another Ultimate system or other computer using the binary synchronous (BSC, or "bisync") protocol. Honeywell-based systems may use a modified asynchronous 2780-type protocol.

Requirements for Bisync Communications

The BSC protocol requires an IBM 2780-type or 3780-type communications device.

For Honeywell-based Ultimate systems with a Honeywell DCM9110 Communications-Pac and a Bell System 801-C Automatic Calling Unit, automatic dialing is possible (see Section 4.19, USING AN AUTOMATIC DIALING UNIT FOR COMMUNICATIONS, as well as B-DIAL below).

Binary Synchronous (BSC) Communications Procedures

Several system commands are associated with bisync communications. These commands are listed below; each is detailed in the System Commands Guide.

Command	Operation
B-ATT	Dedicates bisync controller to a port
B-DET	Detaches bisync controller from a port
B-LIST	Lists bisync attachment table & ports
TRANSMIT	Initiates transmit file request
RECEIVE	Initiates receive file request
DISCONNECT	Disconnects a telephone connection and takes data set out of auto-answer mode
B-DIAL	(Honeywell systems only). Initiates automatic calling to specified number.

Use these steps to transmit and/or receive data with BSC:

To Transmit Data

Any or all items in an Ultimate file (to which the user has access) may be transmitted to another computer with the Ultimate BSC communications process. Follow these steps:

1. Use the Ultimate system level (TCL) command, B-ATT, to attach a bisync communications controller to the line.

2. Use the Ultimate system level (TCL) command, TRANSMIT, to specify the filename, a list of items, and any options. The item-list may have come from a select-list.

3. Dial the other installation which is to receive the data. After voice contact with the operator on the other end is established and he puts his station into the receive mode (or after the other end, in auto-answer mode, answers the line and responds with a 10-second high pitched tone), put your data set into the data mode.

4. The data transfer takes place between the machines.

5. After the transmission is complete, the sending terminal either returns to TCL level ready for another system command, or goes into the receive mode, ready to receive data from the machine on the other end. This depends on the option selected with the TRANSMIT command.

6. To abort the transmission at any time, press the <BREAK> key and type "END". This returns to TCL.

To Receive Data

To receive data, the user account to which the file(s) are transmitted must have a defined RECEIVE-MSGS file (either a D- or Q-pointer item) in the account's master dictionary. Each received message is constructed into a separate item in the file (or items, if the B option is used on the RECEIVE command). Follow these steps:

1. Use the B-ATT command to attach the controller.

2. Use the RECEIVE command to specify the file name to be received into (this defaults to the RECEIVED-MSGS file) and any options.

3. Establish the line connection either by calling the operator at the other end, or by waiting for an incoming call and either answering it or (if the data set is in data mode) having the computer auto-answer.

4. The data transfer takes place. The computer will either file the message(s) in the specified file or route them to the printer Spooler; this depends on the options used on the RECEIVE command.

   NOTE: A filed message is assigned an item-id based on (1) the current time and date, or (2) the value of the first record, or (3) a sequence number; the type of item-id used depends on the options specified in the RECEIVE command.

5. After the transmission is complete, the receiving terminal either returns to TCL level ready for another system command, or re-enters the receive mode for another message. This depends on the option selected with the RECEIVE command.

6. To get out of the receive mode at any time, press the <BREAK> key and type "END". This returns to TCL.

Appendix C contains a Bisync Communications Error Message List, plus the 2780/3780 Protocol (Control Codes), Examples of Data Block Transmission, Transmit Data Parameters, and an ASCII to EDCDIC Conversion Table.

Additional details on 2780-type devices and the communications protocol used can be found in the IBM publications "General Information--Binary Synchronous Communications" (GA27-3004), "Component Information for the IBM Data Communications Terminal" (GA27-3063), and "Component Description: IBM 2780 Data Transmission Terminal" (GA27-3005).

4.19 USING AN AUTOMATIC DIALING UNIT FOR COMMUNICATIONS

Automatic calling is possible on Honeywell-based systems using the bisync protocol with a Honeywell DCM9110 Communications Pac and a Bell System 801-C Automatic Calling Unit.

Equipment Installation

The Honeywell DCM9110 Communications Pac (or Multi-Line Communications Processor) attaches to the MLCP (communications controller). Each DCM9110 supports two automatic calling devices. The Automatic Calling Unit (ACU) lines are logically connected through the board placements on the MLCP(s). For example, if bisync line# 0 is attached to the modem, then the first 9110's cable "0" should be attached to the auto-dialer associated with that modem.

When you execute the B-ATT command, you are specifying which bisync port you want to use; the default is 0. The first bisync board (with the highest priority on the bus) logically contains ports 0 and 1.

Figure A shows an example of an MLCP board configured for automatic dialing. Note that the BISYNC and ACU boards do not have to be next to each other, but the first BISYNC board is logically associated with the first ACU board. In this example, BISYNC lines 2 and 3 (on the second BISYNC board) do not have an associated ACU board. If another ACU board did exist, then that second ACU board would service the second BISYNC board.

Use the following options when ordering the 801-C unit from the Bell System:

A.	1	Signal ground connected to frame ground.
B.	4	Call termination thru data set after DSS on.
C.	5	ACR timer stopped after DSS is on.
D	8	Customer selected ACR timer of 25/28 seconds.

```
----------------------------------------------------------
|                     MCLP BOARD                          |
| -------------  --------------  -----------  ---------- |
| |  BISYNC   |  |  ASYNC     |  |  BISYNC |  |  ACU   | |
| |   2158    |  |   2118     |  |   2158  |  |  2120  | |
| |   0  1    |  |            |  |   2   3 |  | 0   1  | |
| -------------  --------------  -----------  ---------- |
|     |  |                                      |   |    |
______|__|______________________________________|___|_____
      |  |                                      |   |
      |  |           ----------------------------   |
      |  |           |                              |
      |  |___________|________________              |
      |              |               |              |
      |              |               |              |
 -----------         |         -----------          |
 |         |         |         |         |          |
 | MODEM 0 |         |         | MODEM 1 |          |
 |         |         |         |         |          |
 -----------         |         -----------          |
      |              |               |              |
      |              |               |              |
 ------------        |          ------------        |
 | DIALER 0 |  <------          | DIALER 1 |  <------
 ------------                   ------------
```

Figure A. Example of ACU and MCLP Board

4.20 ASYNCHRONOUS COMMUNICATIONS

Honeywell-based systems may transmit and receive data from another Ultimate or other system using a modified asynchronous 2780-type protocol. The protocol is "modified" in that parity is not used on the data characters.

Asynchronous-2780 Communications Procedures

Several system commands are associated with asynchronous-2780 (also called "async-2780") communications. These commands are listed below; each is detailed in the System Commands Guide.

Command	Operation
SET.LINE#	Selects async-2780 communications for port
RESET.LINE	Returns the port to normal usage
A.TRANSMIT	Initiates file transmit request
A.RECEIVE	Initiates file receive request
A.DISCONNECT	Breaks a telephone connection and takes a data set out of auto-answer mode

The ASYNC-2780 Driver

A utility is available that will set up all the necessary parameters to perform ASYNC 2780-type communications. From the SYSPROG Main Menu, you may select the automatic Async-2780 set-up option. Or, from the system (TCL) level, use the command:

```
ASYNC
```

This program automatically resets the line for normal operations when the 'T' option is used in an A.TRANSMIT command, and when the 'E' option is used in an A.RECEIVE command. (A RESET.LINE command is automically executed at the end of the transmission before the terminal returns to the TCL level.)

The command may be terminated by entering 'END' at any prompt.

Use these steps to transmit and/or receive data with ASYNC:

To Transmit Data

Any or all items in an Ultimate file (to which the user has access) may be transmitted to another computer with the Ultimate async-2780 communications process. Follow these steps:

1. Use the Ultimate system level (TCL) command, SET.LINE#, to load the asynchronous-2780 communications software for a specific port (line).

2. Use the Ultimate system level (TCL) command, A.TRANSMIT, to specify the filename, a list of items, and any options. The item-list may have come from a select-list.

3. Dial the other installation which is to receive the data. After voice contact with the operator on the other end is established and he puts his station into the receive mode (or after the other end, in auto-answer mode, answers the line and responds with a 10-second high pitched tone), put your data set into the data mode.

4. The data transfer takes place between the machines.

5. After the transmission is complete, the sending terminal either returns to TCL level ready for another system command, or goes into the receive mode, ready to receive data from the machine on the other end. This depends on the option selected with the A.TRANSMIT command.

6. To abort the transmission at any time, press the <BREAK> key and type "END". This returns to TCL.

7. Optionally, use the RESET.LINE command to return the port to normal terminal usage.

To Receive Data

To receive data, the user account to which the file(s) are transmitted must have a defined RECEIVE-MSGS file (either a D- or Q-pointer item) in the account's master dictionary. Each received message is constructed into a separate item in the file. Follow these steps:

1. Use the SET.LINE# command to load the asynchronous-2780 communications software for a specific port (line).

2. Use the A.RECEIVE command with options, if any.

3. Establish the line connection either by calling the operator at the other end, or by waiting for an incoming call and either answering it or (if the data set is in data mode) having the computer auto-answer.

4. The data transfer takes place. The computer will either file the message(s) in the RECEIVE-MSGS file or route them to the printer Spooler; this depends on the options used on the A.RECEIVE command.

   NOTE: A filed message is assigned an item-id based on (1) the current time and date, or (2) the value of the first record.

5. After the transmission is complete, the receiving terminal either returns to TCL level ready for another system command, or re-enters the receive mode for another message. This depends on the option selected with the A.RECEIVE command.

6. To get out of the receive mode at any time, press the <BREAK> key and type "END". This returns to TCL.

7. Optionally, use the RESET.LINE command to return the port to normal terminal usage.

Appendix D contains an Asynchronous Communications Error Message List.

Additional details on 2780-type devices and the communications protocols used can be found in the IBM publications "General Information--Binary Synchronous Communications" (GA27-3004), and "Component Description: IBM 2780 Data Transmission Terminal" (GA27-3005).

4.21 Asynchronous Communications Using XMODEM Protocol

Any Ultimate system, including PC-based systems running the Ultimate PC/OS operating system, may transmit and receive data from another Ultimate system using the MODEM/XMODEM protocol. This protocol sends 128 data characters at a time. At least one serial port is required on the PC.

The XMODEM Async software does not use the X-ON/X-OFF protocol, and is not recommended for any function other than data transfer.

Requirements Before Using XMODEM

If the non-host (local) computer is a PC, it must be an Ultimate PC110, or IBM AT, or IBM XT, with at least one serial port.

The user at the non-host computer must be able to logon to an account on the host computer in order to complete the connection between the two computers.

The Ultimate host (remote) computer must have a port already configured as follows (see the SET-BAUD command in the System Commands Guide):

1. no parity
2. x-on/x-off set to off
3. proper baud rate for file transfer
4. 8 data bits
5. echo-on/echo-off set to off
6. 2 stop bits

Format of Blocks Transmitted via XMODEM

A received transmission block has the following format:

A	B	C	D	E

Field	Description
A	SOH (Start Of Header), which is hexadecimal X'01', is the start of the transmission block.
B	An 8-bit binary block number of the block of information currently being sent.
C	An 8-bit one's complement of field B (flipped bits).
D	Data in the block currently being sent (128 characters).
E	An 8-bit checksum of the data sent in field D.

XMODEM Asynchronous Communications Procedures

Several system commands are associated with XMODEM async communications. These commands are listed below; each is detailed in the System Commands Guide.

Command	Operation
XMODEM	Initiates XMODEM async driver proc
XMODEMT	Initiates file transmit request
XMODEMR	Initiates file receive request

The XMODEM Driver

A proc is available that will set up all the necessary parameters and perform XMODEM async communications. From the SYSPROG account, at the system (TCL) level, use the command:

    XMODEM

You will then be prompted for the serial port on your system where the file transfer will be made:

    Enter port # that modem is attached to:

Enter the appropriate port number and press <CR>.

Next you will be prompted for the baud rate at which the file transfer will occur:

    Enter the baud rate (300, 1200, or 9600):

This baud rate must be the same as the baud rate of the port on the host computer, which has previously been set (see above).

The proc displays a screen containing all the command sequences for XMODEM file transfer. Note that each command, except "HELP", has a 3-key sequence: <TAB>, a number, and <CR>. You can display the Help screen at any time by entering the word "HELP" and pressing <CR>. These commands are summarized below:

Command	Description
HELP<CR>	Displays the Help sceen.
<TAB>0<CR>	Exits the program; returns to TCL.
<TAB>1<CR>	Establishes contact between the PC (or your system) and the asynchronous line or modem on the host Ultimate computer system.
<TAB>2...<CR>	The ellipsis (...) contains characters you wish to transmit to the modem, up to (but not including) the <CR>. This is useful if the PC has an auto modem.
<TAB>3<CR>	Initiates receiving an item from the host Ultimate computer. (See below.)
<TAB>4<CR>	Initiates transmission of an item to the host Ultimate computer. (See below.)

IMPORTANT: Remember that this procedure controls all aspects of the file transfer from the user's terminal; no host terminal is involved. Use these steps to transmit and/or receive data with XMODEM:

<u>To Transmit Data</u>

Any or all items in an Ultimate file on your system (to which you have access) may be transmitted to a host Ultimate computer with XMODEM. Follow these steps:

1. Be certain the host computer has been set up (via the SET-BAUD command) for XMODEM communications (see above for required settings).

2. Use the Ultimate system level (TCL) command, XMODEM, to invoke the proc.

3. At the XMODEM prompt symbol (#), enter the command sequence **<TAB>1<CR>** to establish communications with the host. You can establish the modem connection before or after the command since the proc will retry until contact is made. You are normally requested to logon to an account on the host system. Ignore any characters echoed to the terminal, and the prompt symbol (which will probably be the TCL prompt ">").

4. To initiate the transfer, enter the command sequence **<TAB>5<CR>**. The proc prompts for the name of the file which contains the item you wish to transfer. Then the proc prompts for the name of the item (item-id) to transfer.

5. You will then be directed to enter the command sequence **<TAB>1<CR>** again, to re-establish communications with the host.

6. When connected, enter the system TCL command "XMODEMR" from the host's SYSPROG account. This invokes the host's receiving proc.

7. You will be prompted for the receiving file name on the host computer; this file must already exist in the host data base. Enter the file name and item name for the received copy.

8. To begin the transmission, enter the command sequence **<TAB>6<CR>.** XMODEM should display the block number of each block of data as it is being transmitted. If this block count is not being displayed or (on a large item) if it is not incrementing, the transfer may be unsuccessful. XMODEM will retry each block up to 10 times before quitting.

9. When the transmission is complete, XMODEM reports "Transfer Successful."

<u>To Receive Data</u>

Any or all items in an Ultimate file on the host computer system (to which your user account on that system has access) may be transmitted to another Ultimate computer with XMODEM. Follow these steps:

1. Be certain the host computer has been set up (via the SET-BAUD command) for XMODEM communications (see above for required settings).

2. Use the Ultimate system level (TCL) command, XMODEM, to invoke the proc.

3. At the XMODEM prompt symbol (#), enter the command sequence **<TAB>1<CR>** to establish communications with the host. You can establish the modem connection before or after the command since the proc will retry until contact is made. Ignore any characters echoed to the terminal, and the prompt symbol (which will be the TCL prompt ">").

4. To initiate the transfer, enter the command sequence **<TAB>3<CR>.** The proc prompts for the name of the file on your system which will receive the item you wish to transfer. Then the proc prompts for the name of the item (item-id).

5. You will then be directed to enter the command sequence **<TAB>1<CR>** again, to re-establish communications with the host.

6. When connected, enter the system TCL command "XMODEMT" from the host's SYSPROG account. This invokes the host's transmit proc.

7. You will be prompted for the file in which the item is stored on the host computer. Enter the file name and item-id where the item is now stored.

8. To begin the transmission, enter the command sequence **<TAB>6<CR>.** XMODEM should display the block number of each block of data as it is being transmitted. If this block count is not being displayed or (on a large item) if it is not incrementing, the transfer may be unsuccessful. XMODEM will retry each block up to 10 times before quitting.

9. When the transmission is complete, XMODEM reports "Transfer Successful."

4.22 SPOOLER SET-UP AND MAINTENANCE

As a multi-user system, each Ultimate user's Process is completely independent of other users. The system line printer(s) and magnetic tape unit(s), however, are shared among users. Print jobs and tape handling are managed by the Spooler and Tape Handler software, respectively.

The Spooler can handle up to forty (40) printers operating simultaneously. Up to four (4) of the total number in operation may be parallel printers; the rest must be serial printers. The Spooler allocates print jobs to a job queue(s) and enables each terminal to continue processing even if the printer device is temporarily unavailable.

The Spooler Process is allocated at system boot time. The Spooler controls outputting from an Ultimate system to a hold file and/or a line printer. In order to output data, you must define to the Spooler the logical printer numbers which are attached to actual line printers.

The output is called a "print job", which is placed in a "job queue" according to the current SP-ASSIGN location. Up to 128 job queues may be used for different types of jobs, each with its own criteria (e.g., forms, time requirements, etc.). Each enqueued job is printed in its turn.

A print job can be held on disk in a "hold file" instead of being enqueued for output, if specified in the current SP-ASSIGN command. The job may be printed as needed, and then deleted, by the SP-EDIT command.

A number of "Spooler" commands are available as system (TCL) level commands. In addition, from the SYSPROG menu, you may select a Spooler menu that contains the most commonly used Spooler commands. From the Spooler menu (see Figure A), you may perform a number of tasks related to Spooler and printer set-up and maintenance. Also, at the TCL level on the SYSPROG account, you may use the SP-MENU command to display the Spooler menu.

Each option corresponds to an Ultimate system command:

1. SP-STARTLPTR
2. SP-STOPLPTR
3. SP-DELETELPTR
4. SP-ASSIGN
5. SP-LISTQ
6. SP-EDIT
7. SP-DEQ
8. SP-STATUS
9. SP-LISTLPTR

The Spooler menu options offer an alternative to entering the commands at the system level. However, the prompts and information requested will be the same as if the command itself had been entered.

Other commands relating to the Spooler and line printers are:

```
PRINTER   specifies a letter quality printer type
SET-TERM  displays the printer page width and depth
    TERM  displays the printer page width and depth
SP-CLOSE  closes all open print jobs
SP-KILL   abort job currently being printed
SP-LISTASSIGN lists assignments of all terminal lines
SP-OPEN   keeps next print job open for more data
SP-SKIP   sets the number of pages to skip between jobs
```

For details about the Spooler commands, see the System Commands Guide; each command is alphabetically listed and fully described.

```
                 The Ultimate Spooler Menu
                    Time           Date

     1.   Start a line printer
     2.   Stop a line printer
     3.   Delete a line printer
     4.   Set the output print assignments for your line
     5.   List the print jobs in the job queue
     6.   Process a Hold File in the job queue
     7.   Dequeue a job in the job queue
     8.   List the status of the Spooler & each line printer
     9.   List the assignments and status of every printer

     Please enter the option of your choice or TCL or OFF
```

Figure A. Spooler Menu

4.23 OVERVIEW OF TAPE CONTROL COMMANDS

The system Tape Handler software manages the tape drive(s) in the system so that only one user at a time is attached to any given tape drive. In a disk only system, the removable disk cartridge or floppy drive is always treated as the tape device when a tape command is used.

In order to use a tape unit, the magnetic tape or cartridge must be mounted and brought to "load" (ready) point. Then, the user must issue a command (in most cases) to attach the tape to that user's Process and position it to the block where information is to be written or read.

The Ultimate system Tape Handler supports variable tape block sizes. When reading from tape (e.g., T-LOAD, READT, etc.) the Tape Handler automatically checks the size of each block and the correct number of bytes of data is transferred into the internal tape buffer. (The block size recorded in the tape label is for reference purpose only, and is not used by the Tape Handler.)

Multiple tape drives (up to four per system) are supported, plus a removable disk cartridge or floppy. Thus, two different users (on two different terminals) can use two different tape drives at the same time. For example, one user may be doing a T-DUMP on tape drive 0 while another user is T-LOADing a file from tape drive 1.

On a single-tape drive or disk-only system, it is not necessary to do a T-ATT. Before executing any Tape command, the Tape Handler will attempt to attach the (only) tape drive if it is not already attached. Similarly, on a disk-only system, before executing any Tape command the Tape Handler will attempt to attach the removable cartridge or floppy if it is not already attached.

On a system with multiple tape drives, no implicit T-ATT will be made when another Tape command (such as T-DUMP or T-LOAD) is executed. Prior to the use of any Tape command, a user must explicitly do a T-ATT to a specified drive; otherwise the Tape command will cause an error message to be displayed:

```
No tape drive attached.
```

On-line copies can be made from tape to tape, from tape to disk cartridge, or from cartridge to tape. A T-STATUS command displays the status of all tape drives on the system.

Below are listed the Tape commands available as system (TCL) level commands:

T-ATT	attaches a tape device to user's Process
T-DET	detaches a tape device from user's Process
T-COPY	copies files from one tape to another
T-ERASE	erases a 2-inch section of tape
T-FWD	moves tape forward 'n' blocks or to end-of-file
T-BCK	moves tape backward 'n' blocks or to end-of-file
T-REW	rewinds tape to load point
T-SPACE	moves tape forward 'n' files
T-EOD	moves tape forward to double end-of-file mark
T-WEOF	writes an end-of-file mark on tape
T-CHK	checks a tape for parity errors
T-DUMP	dumps file items onto tape
S-DUMP	sorts and dumps file items onto tape
T-LOAD	loads file items from tape
T-READ	dumps tape contents to terminal/printer
T-RDLBL	reads tape label
T-STATUS	displays current status of all tape drives
T-UNLOAD	rewinds and unloads the tape

For details about the Tape commands, see the System Commands Guide; each command is alphabetically listed and fully described.

NOTES

SECTION 5

TROUBLESHOOTING AND PROBLEM RESOLUTION

5.1 HOW TO DETERMINE SYSTEM STATUS WHEN PROBLEMS OCCUR

The first task in solving problems within an Ultimate system or user Process is to diagnose the scope of the problem, and the probable cause(s). This is called "troubleshooting." Once the problem has been defined, the appropriate solution or recovery procedure can be undertaken.

When the entire system has crashed

The major symptom of a system "crash" is the situation where no terminals are operative. Users are unable enter data on any terminal, including line 0, and are unable to perform any function from any attached line (port). This condition may be caused by a software problem, a hardware malfunction, or a power failure.

In the case of system crashes, it is highly recommended that the System Manager immediately call the Ultimate System Support group for diagnostics and assistance. Do not turn off any terminals, printers, or tape drives.

To restart the system, the system must be re-booted from the Boot or Initialize button on the CPU front panel. (Please see Section 2.1, BOOTSTRAPPING THE SYSTEM for procedures.) Try to bring the system up, first via the (W)armstart option and then the (C)oldstart option.

The warmstart will, if possible, restart all user Processes at some point within a few moments of the crash. The DUMP, WHAT and WHERE commands may be of value in diagnosing the cause of the crash.

If a warmstart does not bring up the system, use the (C)oldstart option. Coldstarting re-initializes the entire system software and user Processes. All users will have to log on again, and some work that was in progress may be lost. The WHAT and WHERE commands will not reflect anything that happened before the crash. It may be helpful to do a dummy file-save to check for group format errors (GFEs).

When a terminal appears hung up

A terminal "hang up" is the situation where one terminal becomes inoperative, but other terminals in the system are still operational. One user at a terminal is unable to enter data on that terminal and cannot perform any function. This may be caused by incoming AC power fluctuations or static discharges, or hardware malfunctions.

The first step is to verify the status of the terminal; the second step is to try to restart the terminal. Use these procedures, in the order given (simple to difficult "fixes"):

NOTE: Make sure the terminal is securely connected to the computer cable.!

1. Press the <BREAK> or <CTRL/BREAK> key code. If the Debugger prompt is displayed, note the BREAK address (I ###.###). If the frame number is 667, the program is waiting for an item lock to clear. Enter "G <CR>" to restart the program (as soon as possible). If no response, go to the next step.

2. Turn the terminal off, and then on again. Then try step 1 again. If no response, go to the next step.

3. If the <BREAK> key is not operative (but is enabled), check the asynchronous port by going to another terminal; issue a "SET-BAUD port,baud" command specifying your port (line number) and the baud rate of 9600. Then try step 1 again. If no response, go to the next step.

4. From another terminal issue a ":RESTARTLINE line#" command. The Logon message should appear, with the terminal set at the system's default baud rate. Use SET-BAUD to reset, if necessary.

   To check the communications between terminal and computer, issue a "WHERE line#" command. If the Process (PIB status) is 02 (the first column after the line number is "02nn"), try this. Enter a ":TRAP line#" command, and then type any character or press any key on your keyboard. Then issue another WHERE command for your line. If the :TRAP has changed the Process to "01nn", enter an ":UNTRAP line#" command. The Logon message should appear on your terminal, as in :RESTARTLINE.

5. If none of the above methods work, call Ultimate Support for assistance.

If you restart with the Logon message displayed (or after logoff), log on again and resume work. The work that was performed since the last file update will not be saved.

NOTE: A terminal may appear inoperative if it is being used to run a program while the BREAK key is disabled, or is running in batch mode, as well as waiting for a group or item lock to clear. Sending an X-off (the <CTRL/S> key sequence) may also cause a terminal to appear hung, when actually it will return to normal simply by sending an X-on (<CTRL/Q>) or by pressing the appropriate <BREAK> key function.

When a tape drive, printer, or disk drive stops functioning

On occasion, an interrupt from one of the devices on an Ultimate system may not be processed.

If a tape drive becomes inoperative, press the On Line button twice (to take it off-line and back on-line).

If a printer becomes inoperative, follow these steps:

1. Take the printer off-line by pressing the On Line button.
2. Open and then close the printer gate (if possible).
3. Put the printer back on-line by pressing the On Line button again.
4. If the above does not work, turn the printer off. Restart the printer and resend the print job.

If a disk drive becomes inoperative, some systems may be able to restore the connection by following these steps:

1. Press the Protect button; the Protect light should go on.
2. Press the Start button; the start and ready light should go off.
3. Let the disk drive cycle down; then press the Start button again and the Start light should go on.
4. When the Ready light comes on, press the Protect button. The Protect light should go out and the Start and Ready lights should go on.

If any of the above procedures are unsuccessful, call the Ultimate Support Group for diagnosis and assistance.

When an error message appears on the terminal

If a BASIC program is being compiled or run, the error message is normally a BASIC error message. Appendixes A and B in the BASIC Reference Manual list the messages and causes.

If an assembly program is being assembled or run, the error message is normally an Assembler or System error message. Appendix A of this manual lists the System error messages; the Assembler manual lists the Assembler messages.

Command	Operation
DUMP	Display data in frame or group of frames.
LIST-SYSTEM-ERRORS	Display log of disk, memory, and other errors.
MVERIFY	Verify assembly language object code.
POVF	Display frames in Available Space (Overflow) pool, both contiguous and linked.
REV	List the release level of the current (running) Ultimate operating system.
VERIFY-SYSTEM	Verify assembly language system software.
WHAT	Display system configuration, the current status of its locks and tables, the location of all logged-on Processes, and all TCL levels for all terminal lines.
WHERE	Subset of WHAT. Display all TCL levels for all terminal lines.

NOTE: All system level (TCL) commands are alphabetically listed and fully explained in the System Commands Guide.

System Status and Verification Commands

5.2 THE ULTIMATE SYSTEM DEBUGGERS AND PROMPT SYMBOLS

When the system is executing commands and programs, normally one of three main prompt symbols may appear on the terminal, awaiting user command input. The most common prompt symbol is the greater-than (>) sign, which is the system (TCL) level prompt. In addition, there are two Debugger prompt symbols. When one of these appears on the terminal, the system has entered either the BASIC or Assembly Debugger program.

An Ultimate system has three main prompt symbols:

- > the system level (TCL) prompt
- ! the Assembly (System) Debugger prompt
- * the BASIC Debugger prompt

These prompt symbols tell a user whether the system is operating at the TCL level (normal system operating level), or has entered one of the Debuggers. The Assembly Debugger may be entered if an assembly program fails, or the user presses the <BREAK> key, <u>or</u> if a user turns off the terminal without logging off (via the OFF command). The BASIC Debugger is entered if a BASIC program fails, or if the terminal is turned off or the user presses the <BREAK> key while a BASIC program is running.

From the system (TCL) level prompt, you may enter any system command (within the limits of your account's privilege level). All Ultimate-supplied commands are alphabetically listed and explained in the System Commands Guide.

For options at the BASIC Debugger prompt, see below.

For options at the Assembly Debugger prompt, see below.

<u>When the BASIC Debugger (prompt = *) has been entered</u>

BASIC programs that encounter an unrecoverable error during runtime, or are running when a user presses <CTRL/BREAK> or turns the terminal off, generally exit to the BASIC Debugger.

To exit the Debugger, use one of these commands:

END	Exits the program and returns to TCL.
DEBUG	Exits to the System Assembly Debugger.
DE	Same as DEBUG.
OFF	Exits the program and logs user off.

NOTE: The END command releases any secondary TCL levels currently in effect for the line.

To use the Debugger for program testing and diagnostics, refer to the Debugger commands in the BASIC Reference Manual.

When the System Assembly Debugger (prompt = !) is entered

The System Assembly Debugger is generally entered if:

1. The user presses the <BREAK> key function for the terminal.

2. A user has not properly logged off the terminal (via the OFF command). The "!" prompt may appear when the terminal is turned on again.

3. During run-time, a program written in assembly code encounters an unrecoverable error.

It is also known as the "System" Debugger since most system software is written in assembly code.

The general format of an error message generated when the System Debugger is entered during run-time is:

```
error message; Reg = nn.
Abort @ fff.bbb
!
```

For example:

```
Forward Link Zero; Reg = 14
Abort @ 165.1A7
!
```

refers to the error "Forward Link Zero". The address register in the user Process' PCB (Primary Control Block) that was currently being used is "14". The system aborted at frame 165 (decimal); the byte offset was 1A7 (hex), which is 423 in decimal.

In case of an unrecoverable error, copy this entire message down, because Ultimate Support needs the information for troubleshooting.

To exit the Debugger and return to the TCL level, enter:

END

The program cannot be restarted from where it failed after you "END" the Debugger session. ENDing releases any secondary TCL levels which may be in effect for the line.

To use the Assembly Debugger for program testing and diagnostics, the user account must have a SYS2 privilege level. In order to use the ">" and ">>" commands, the account's user identification item in the SYSTEM dictionary must not have the Restart (R) option. The Assembly Debugger commands are covered in the Assembler Reference Manual.

5.3 MULTIPLE TCL LEVELS: ENTRANCE/EXIT COMMANDS

Each terminal line may have stacked, multiple TCL levels. From the System Assembly Debugger (prompt = !), users may create secondary TCL levels. The WHERE command displays all TCL levels currently in effect for the line.

The System Assembly Debugger contains several commands that allow entering and exiting from secondary TCL levels for program test purposes:

```
>command       Execute TCL command from System Debugger
>>             Access next secondary TCL level
<              Return to current TCL level
<<             Return to previous TCL level
END            Return to first TCL level
```

<u>The ">" command</u>

A TCL command may be executed from the Debugger by entering
<u>The ">" command</u>

A TCL command may be executed from the Debugger by entering
the ">" followed by the command. For example:

```
!>WHO 0
0 SYSPROG
!
```

The ">" command creates a secondary TCL level, executes the TCL command in the secondary level, and exits by releasing the secondary level workspaces back to the Available Space pool. Control is returned back to the Debugger of the current TCL level.

<u>The ">>" command</u>

A new secondary TCL level may be created without executing a command by entering only ">>". Figure A shows an example of a WHERE command executed at the first TCL level, followed by a <CTRL/BREAK> to get to the Debugger.

For example:

```
     >WHERE

 000 0210 000400      6.094     6.033    5.088
*003 0010 000460    121.001   121.074  <--------you are here
 009 4030 000520    170.047   170.07B

     >   <--------------press <CTRL/BREAK> to get to
                        System Debugger
     I 6.94
     !>> <------------- enter the >> command

     >>  <------------ you are at the new secondary TCL level
```

```
     >>WHERE  <-------- execute a WHERE command

 000 0210 000400      6.094     6.033    5.088
*003 0010 005B6A    121.001   121.074   <--------you are here
          000460      1.0BD     6.094    6.033  5.088 <----
                                        execution suspended
 009 4030 000520    170.047   170.07B

     >>  <--------------- press <CTRL/BREAK> again
     I 6.94
     !>> <--------------- enter >> command again

     >>  <----------- you are at ANOTHER secondary TCL level

     >>WHERE  <---------- execute a WHERE command

 000 0210 000400      6.094     6.033    5.088
*003 0010 005D7D    121.001   121.074   <--------you are here
          005B6A      1.0BD     6.094    6.033  5.088 <----
                                        execution suspended
          000460      1.0BD     6.094    6.033  5.088 <----
                                        execution suspended
  09 4030 000520    170.047   170.07B
```

<u>The "<<" command</u>

The "<<" command in the System Debugger brings the user to the previous TCL level, unless it is already at the first TCL level. If so, it is equivalent to the END command. The workspaces of the current TCL level are returned to the Available Space (Overflow) pool.

Continuing from the last example:

```
     >>  <------------ press <CTRL/BREAK>
     I 6.94
     !<< <------------ enter the << command
     !  <------------ you have returned to the previous level
     !G <------------ to leave the System Debugger
      <CR> <--------- press <CR> to get the prompt

     >>  <----------- you are at a secondary TCL level, but
                      which one?

     >>WHERE
 000 0210 000400      6.094     6.033    5.088
*003 0010 005B6A    121.001   121.074   <--------you are here
          000460      1.0BD     6.094    6.033  5.088 <-----
                                        execution suspended
 009 4030 000520    170.047   170.07B
```

The END command

The System Debugger END command brings the user out of the Debugger and back to the first TCL level. All secondary TCL levels are exited and all workspaces of secondary levels are released to the Available Space (Overflow) pool.

Continuing from the last example:

```
     >>  <------------ press <CTRL/BREAK>
     I 6.94
     !END <----------- enter the END command

     >   <------------ now you are at the first TCL level

     >WHERE
 000 0210 000400      6.094     6.033    5.088
*003 0010 000460    121.001   121.074   <---------you are here
 009 4030 000520    170.047   170.07B
```

The "<" command

The "<" command brings the user out of the System Debugger and back to the current TCL level. When the user is already at the first TCL level, this command is equivalent to the END command.

This command is useful when the user is in the System Debugger and wants to return to the current TCL level. (Note that the END command brings the user back to the <u>first</u> TCL level.)

Continuing with the same example:

```
     >   <--------------press <CTRL/BREAK> to get to
                        System Debugger
     I 6.94
     !>> <------------- enter the >> command

     >>  <------------ you are at the new secondary TCL level

     >>WHERE  <-------- execute a WHERE command

 000 0210 000400      6.094     6.033    5.088
*003 0010 005B6A    121.001   121.074   <---------you are here
          000460      1.0BD     6.094    6.033  5.088 <----
                                        execution suspended
 009 4030 000520    170.047   170.07B

     >>  <--------------- press <CTRL/BREAK> again
     I 6.94
     !<  <--------------- enter the < command
```

```
>>  <----------- you are back to the same TCL level

>>WHERE  <---------- execute a WHERE command

 000 0210 000400      6.094     6.033    5.088
*003 0010 005B6A    121.001   121.074   <---------you are here
          000460      1.0BD     6.094    6.033  5.088 <----
                                          execution suspended
 009 4030 000520    170.047   170.07B
```

5.4 MULTIPLE TCL LEVELS: HANDLING TRANSFER OF INFORMATION

Generally speaking, each TCL level is independent of any other TCL levels. Terminal characteristics, tape attachments and printer (SP-ASSIGN) assignments, however, are carried over between levels in both directions. Select lists created by EXECUTE statements can be used by a READNEXT statement in the same program.

During program operation, each TCL level is independent of any other TCL levels. For example, a select-list produced by the SELECT command on a secondary TCL level is lost when that level is exited and hence cannot be carried back to the previous level.

However, there are several exceptions:

1. In a BASIC program, a select-list produced by an EXECUTE statement can be used by a READNEXT statement in the same program. For example:

   ```
        EXECUTE "SELECT MD 'ED'"
   10   READNEXT ID ELSE STOP
        PRINT ID
        GOTO 10
   ```

   will print one item-id.

2. The terminal characteristics are carried over between levels in either direction. That is, when entering a new TCL level (either via the ">>" command or the EXECUTE statement), the terminal characteristics are carried over to the new level. If any of the terminal characteristics are changed in the new secondary TCL level, the changes will be carried back to the previous level.

   Terminal characteristics include the following:

   All parameters changeable by the TERM command
   Echo on/off (changed by ECHO-ON/ECHO-OFF cmds.)
   X-on/off (changed by X-ON/X-OFF commands)
   All parameters changeable by SET-BAUD command

3. The SP-ASSIGNments are carried over between levels in either direction. That is, when entering a new TCL level (either via the ">>" command or the EXECUTE statement), if the SP-ASSIGNments are changed in a secondary level the new assignments will be carried back to the previous level.

4. The tape attachment is carried over between levels in either direction. That is, when entering a new TCL level (either via the ">>" command or the EXECUTE statement), the tape attachment is carried over to the new level. If the tape attachment is changed in a secondary level, the new tape attachment will be carried back to the previous level.

TERM command	- terminal characteristics
ECHO ON ECHO OFF	- echo characteristic on terminal
X-ON X-OFF	- communications protocol characteristic
SET-BAUD	- baud rate and related parameters
SP-ASSIGN	- Spooler assignment for terminal
T-ATT n	- tape attachment
SELECT commands	- any select-list produced

NOTE: All system level (TCL) commands are alphabetically listed and fully explained in the System Commands Guide.

Figure A. Commands Carried Over Between Mulitple TCL Levels

5.5 HANDLING WORKSPACE ALLOCATION PROBLEMS

User Process workspaces (at the first TCL level) are a set of contiguous, linked frames that are initialized by the system at coldstart or system-generation time. If the system cannot allocate or find workspaces for a Process during run-time, the currently executing program will terminate.

The first TCL level of workspace is linked after a File-restore, or it may be linked from TCL by use of the LINK-WS command. At File-restore time, the Spooler links the workspace for all the other lines, and no other user can log on the system while this linkage is taking place. The message:

    Linking workspace--wait

will appear until the Spooler has finished the linkage.

The starting FID of the workspace may be computed as below:

    WSSTART = 1024 + (number of lines) * 32

Each line has three (3) workspaces of one hundred and twenty-eight (128) contiguous frames. This is added to the PCB and following 31 frames for a total of 416 frames of workspace per user Process.

The workspace may be linked on a live system using the LINK-WS command on the SYSPROG or SECURITY account. This may be done if it is suspected that the links of the additional workspace have been destroyed for some reason.

As the linkage proceeds, the line-number of the process whose workspace is currently being linked is displayed on the terminal where the LINK-WS command was entered. If the line is logged on, the message "On!" will be displayed, and <u>the workspace is not relinked.</u>

The Spooler's workspace can only be relinked via a coldstart. However, internally, most of the Spooler's workspace is not used in a manner that can cause problems if its links are not correct.

<u>BASIC Run-time crash: Message "Not Enough Workspace"</u>

If a BASIC program aborts to the BASIC Debugger with the message "Not Enough Workspace", it is likely that the workspaces must be relinked on that line. The LINK-WS command relinks primary TCL level workspaces.

Workspaces for all secondary TCL levels are obtained from the Available Space pool (Overflow). The workspaces for the primary TCL levels on all terminal lines are built at system boot time and are never released to the Available Space pool.

When a secondary TCL level is no longer needed, its workspaces are returned to a separate Secondary Level Overflow table. The secondary Overflow table frames are kept initialized and ready for later use, as needed.

If at any time regular Overflow frames are needed and there are not enough frames in the Available Space pool, then frames from the Secondary Level Overflow table are taken back to the regular Overflow table. The frames are returned, 416 at a time (the workspace size for one secondary TCL level), until the required number of frames is available.

The POVF command displays the number of frames in both the primary and secondary overflow tables (see Figure A). Secondary overflow, if any, is indicated in the right-hand side of the first display line. The starting FID of the secondary overflow table is printed, followed by the number of frames in the table in parentheses.

NOTE: If workspace has become damaged or unlinked, use the LINK-WS command to relink the primary workspace. (See the LINK-WS command, listed alphabetically in the System Commands Guide.)

```
>POVF
                     5184 (    416)
23488-131167 : 107720

TOTAL NUMBER OF CONTIGUOUS FRAMES: 107720
```

Figure A. POVF Display of Overflow Table Frames

5.6 INITIALIZING WORKSPACES FOR SECONDARY TCL LEVELS

The :TASKINIT command can be used to obtain and initialize workspaces for TCL levels beyond the primary workspaces, which are linked automatically at system boot time.

When secondary workspaces are needed by a user Process, these frames must be available in the secondary Overflow pool. If they are not, it will require extra time to obtain and initialize the workspaces from the primary Overflow table.

The :TASKINIT command may be used to obtain and initialize extra TCL workspaces for future use, and reserve them in the secondary Overflow pool. The size of workspaces for one TCL level is 416 frames.

The general format is:

    :TASKINIT {levels} {(C)}

The optional <u>levels</u> specifies the number of TCL levels to obtain and initialize workspaces for. If <u>levels</u> is omitted, the command displays the number of levels currently initialized. If there are already enough workspaces in the secondary Overflow table for that number of TCL levels, the command takes no action; otherwise, additional workspaces are obtained and initialized.

The 'C' option causes the system to check all existing secondary levels to see if they are properly linked. If any level is found unusable, it is discarded but the frames used by that level <u>are not returned</u> to the Overflow table.

To make sure that additional workspaces are available when needed by secondary TCL levels, you may wish to include the :TASKINIT command in your USER-COLD-START proc (in SYSPROG's DICT SYSPROG-PL file).

NOTE: For additional details about the :TASKINIT command, see the System Commands Guide.

```
>POVF
                         5184 (    416)
23488-131167 : 107720

TOTAL NUMBER OF CONTIGUOUS FRAMES: 107720
```

Figure A. POVF Display of Overflow Table Frames

5.7 THE EFFECT OF NEW STANDARDS ON DATA FILE UPDATES

Beginning with the Ultimate system release level R10*130, the updating of files, dictionaries, and programs conforms to new standards. The first four attributes of file definition items (D-pointers), object code pointer items (CC-pointers), and saved list pointer items (CL-pointers), cannot be created or modified by ordinary file-updating software such as BASIC or the Editor. Users should be familiar with these new standards and their effect on the data base.

With R10*130, the updating of file definition (D) items was restricted to the CREATE-ACCOUNT, CREATE-FILE, and File-restore programs. The CREATE-FILE and CREATE-ACCOUNT software (and their synonyms, UPDATE-FILE and UPDATE-ACCOUNT) allows updates to existing file definition items, but it does not allow adjustments to the base, modulo and separation values. (Reallocation parameters can, however, be specified so that the modulo or separation will be changed at the next File-restore.)

The updating of cataloged BASIC items is restricted to the BASIC compiler, and the creating of pointer items is restricted to the SAVE-LIST/EDIT-LIST program.

The Editor is allowed to modify file definition, cataloged, and pointer items with the exception of the first four attributes, which contain the item type code, base, modulo, and separation.

The format of File-Save and Account-Save tapes is the same as in releases prior to R10*130. Passwords and file access codes (retrieval and update locks), however, are handled differently. Thus, they may require some manual conversion when brought from one system to another.

Sometimes, also, a user wants to remove a file when its integrity is questionable, but doesn't want to affect the integrity of the rest of the system data base. The D item in question can be deleted from the Editor.

If a program is having problems accessing a file that should be in the data base, the user should make sure that the file access codes are in the proper format. The UPDATE-FILE and UPDATE-ACCOUNT commands, or the Editor, could be used to verify these codes.

NOTE: Sections 5.8 and 5.9 discuss error handling procedures for parity and group format errors encountered during a File-restore.

<u>File Access Problems Due to Group and Item Locks</u>

The system group lock table contains 368 entries; the item lock table allows for 408 entries. Group locks are used by system software to prevent more than one user Process at a time from modifying a group (as during a BASIC WRITE statement). Item locks are available to control access to items (via the BASIC READU, READVU, and MATREADU statements) which are subsequently modified. If a file contains no more than one item per group, there is no practical difference between locking an item and locking a group. With several items per group, however, it is possible for different user Processes to lock different items within the same group.

Updating any item in a group requires locking the group during the actual update. Also, item locks are not unique to each item; a hash value derived from the item-id is used to retrieve a lock from a pool of available item locks for a particular item. It is possible for more than one item in a group to generate the same hash value. In that case, the first item would be granted the lock and the others would be forced to wait. (The LIST-LOCKS command may be used to monitor this situation.)

Also see Section 5.10, TROUBLESHOOTING THE RESTORED SYSTEM AND DATA BASE, for a summary of system (TCL) commands that can be used to verify the system software and perform different types of analysis on the data base.

5.8 RESOLVING PROBLEMS WITH FILE RESTORES: PARITY ERRORS

> If parity errors or other errors mar the FILES section of a FILE-SAVE tape, some data may be lost. The File-restore will continue, but operator assistance may be needed.

If errors occur in a File-restore, a possible cause is that a faulty reallocation has taken place. A file can be allocated incorrectly if, for example, too small a modulo is used, thus causing data to be lost on the restore. The cause of using the faulty modulo may not be due to human error but to a parity error in reading the tape, or perhaps a hardware or software error when the tape was created.

Parity Error Recovery Procedure

If a parity error is detected on a file restore, the prompt:

```
Parity error!
(A)ccept/(R)etry/(Q)uit?
```

will be printed. Entering 'A' will cause the data block to be accepted as it is from tape without data correction. The specific item and file affected cannot be determined except as can be judged by the tape position, and the current set of files which have not been completed.

Recovery From Destroyed Pointers

If tape information identifying a file is destroyed, it may be impossible for the restore to create that file and subsequent files in the right order. The message:

```
Err "D" segment
@fff.ddd
Level (1-3)?'
```

will be printed. The "fff.ddd" number gives the frame and hex displacement of the software location at which the error was detected. The operator must advise the Restore software as to the appropriate action. Each "level" takes a different path:

1. Search for and continue the restore with the next account on tape,

2. Search for the next dictionary file on tape, or

3. Search for the next data file on tape.

The response requires the operator's judgment as to the positioning of files on the tape and the total situation.

5.9 PROCEDURES FOR GROUP FORMAT ERRORS (GFEs)

A group format error (GFE) may occur when either reading or writing an item from/to disk. The GFE indicates that the group (bucket) in which the item is located has been incorrectly or incompletely updated. Some or all of the data in that group may be destroyed or erroneous. The frames in that group must be corrected in order to restore data base integrity.

The term "group" in a file is used to specify one "bucket" of storage. A file is made up of a collection of groups; the modulo specifies the number of groups in the file.

Disk file storage is handled by a "hashing algorithm." The hashing algorithm uses the specified item-id to decide which group the item belongs in. The file retrieval or storage routine then searches that group for the specified item. The hashing algorithm may be thought of as dividing the item-id by the modulo in order to obtain the remainder. This remainder is then the 'group number', and specifies the group which is to be searched. Within each group the items are stored physically end to end. The logical end of a group is indicated by a segment mark following the segment mark at the end of the last item in the group. An empty group has a segment mark in the first data byte of the group.

Each item is made up of a key, a count field, and the data. The Ultimate system has conventionally used the term 'item-id' in place of the term 'key'. The item-id is the key which is used to look up the location of the item.

The count field exists only in a file representation of the item. Its value is a 16-bit binary number, such that the high-order bit is zero, represented in the file in ASCII hexadecimal notation, and as such takes up four bytes of storage. It immediately precedes the item-id in the file. If the item in question is the first item in the group, the count field starts in the first data byte in the frame. If the item is not the first item in the group, then the count field starts at the first byte after the termination mark of the last item. The count field is used as a pointer to the end of the item. The end of the item must be an attribute mark followed by a segment mark. If the count field does not point to this pattern, there is a group format error, and the group format error handler will be entered.

Group Format Errors: Temporary or Permanent?

A group format error may be either temporary (soft) or permanent (hard). Temporary group format errors will be encountered if another Process is writing an item into the group at the same time that you are trying to read an item in the group. The read without update routines, notably RECALL, RUNOFF, and PROC, do not check the group or item locks which are set by the update routine.

The system may catch the group in a temporary state in which the end of an item has moved, for example, but the count field has not yet been corrected. This condition appears as a group format error to the read routines. In cases like this, the read software will first re-execute the read after setting the appropriate locks (after the update routine has released them). Then, if no errors are detected, the locks will be released and the read will be considered error-free.

Otherwise, a permanent group format error has been encountered, and a message is displayed on your terminal.

Permanent Group Format Errors

A permanent group format error is sensed if the count field does not point to a segment mark. This may occur if the count or the data at the end of the item is wrong.

The count field is definitely wrong if any or all of the four digits which make up the count field are not ASCII hexadecimal digits, which are X'30' - X'39' (corresponding to 0-9), or X'41' - X'46' (corresponding to A-F).

The end of item data may be wrong if the count field contains the wrong ASCII hexadecimal digits, or if the end of item data itself is incorrect.

The end of item data may be incorrect in several ways. If the item is contained within a single frame, then the same errors that apply to the count field will apply to the end of item data.

If a single item is stored across (continued past) a frame boundary, certain other situations may occur. If a user was in the process of updating an item, to the extent that the first frame containing the item was written to disk, but that the last frame was not written when the program was interrupted by something like a cold start, then a group format error will occur. If the overflow handler becomes confused, the frames attached to a group may be acquired by another data file or by a print (Spooler) file.

The DUMP command may be used to look at the contents (including link fields) of the affected frames. The difference between a data file and a print file frame should be obvious on inspection: Print files do not normally contain attribute or value marks and data files do not normally contain carriage-return, line-feed sequences. If the damaged frame is the result of an incomplete update, then the difficulty is localized. Repair of this group will usually attend to the matter. (The System Assembly Debugger or the FIX-FILE-ERRORS command should be used.)

If the damage appears to be due to co-ownership of the frame, the problem may be greater. In this case it is best to leave the frame with the frame to which it has a back-link, presuming that the data is consistent in that chain. Then, using the Assembly Debugger, cut the forward link in the spurious chain and terminate the group.

A file's "primary space" is the initial storage area prescribed by the modulo and separation values; for example, a file with mod/sep values of 2,1 would have 2 frames of primary filespace. In the case where print file data has been written into the primary space, it is probably best to recreate the file and selectively restore (SEL-RESTORE command) it. The old file pointer should be deleted using the Editor. Do not use the DELETE-FILE command on the old file, because this may cause further group format errors.

Recovery from Group Format Errors

Figure A shows a proc that may be used to display GFEs logged in the STAT-FILE.

The recovery strategy is to identify the file affected and do a SEL-RESTORE on the file. It is best to do this as soon after the group format error is noticed as possible. Figure A shows a PROC that can be used to display GFEs on disk, grouped by user account. If you are trying to recover items that may not be saved on tape, then the FIX-FILE-ERRORS command might be of some help.

In this context, note that the organization of file-save tapes written by the Ultimate system puts an end-of-file mark at the end of each account, and a tape label at the beginning of each account. This means that the reel upon which the needed file starts may be mounted, rather than starting at the beginning of the tape. If the beginning of the required reel occurs in the middle of the desired account, do a SEL-RESTORE command with the 'A' option.

For additional information about using the Assembly Debugger as well as technical details about frame formats and link fields, please see the Ultimate Assembler Manual.

```
     DISPLAY-GFES
PQ
C
T C
O
OGFE REPORT IN PROGRESS
O
C DISPLAYS INFORMATION REQUIRED TO WORK ON GFES
C
HSORT STAT-FILE WITH ERRORS > "0" BY USER BY NAME
H X NAME BMS
H TOTAL GFE TOTAL PT.ERR TOTAL OBJ.ERR
H GRAND-TOTAL "TOTAL NUMBER OF ERRORS IS: 'U'"
H HEADING "GROUP FORMAT ERRORS DISPLAY 'TL'" ID-SUPP
P
O
IF E = 401 OTHERE ARE NO GFE'S....
O
O PRESS <CR> TO CONTINUE +
IN
```

Figure A. PROC Stored as Item 'DISPLAY-GFES'

5.10 TROUBLESHOOTING THE RESTORED SYSTEM AND DATA BASE

If a restored Ultimate system seems to be faulty, a number of system commands are available to verify the system software and perform different types of analysis on the data base.

When problems are encountered in File-restores, often the symptoms are either parity errors, group format errors, or both (see Sections 5.8 and 5.9 for error handling procedures).

In addition, Figure A summarizes a number of system (TCL) level commands that may be used to troubleshoot system and data base problems. If additional assistance is needed, call the Ultimate Support group.

Command	Operation
CHECK-SUM	Verify the checksum of file items to determine if data in a file has been changed.
DUMP	Display the format of data in a frame or group of frames.
FIX-FILE-ERRORS	Determine the nature of group format errors and correct them, if possible.
GROUP	Display the base FID of each group in a specified file, plus every item-id in the group and its character count.
HASH-TEST	Produce a file hashing histogram on the basis of a specified test modulo.
ISTAT	Produce a file hashing histogram for specified file items. This shows the structure of the file and its items, as well as other statistics.
ITEM	Display the base FID of the group into which a specified item-id hashes, plus every item-id in that group and its character count.
MVERIFY	Verify assembly language object code.
POVF	Display the Available Space (Overflow) pool of contiguous and linked frames.
VERIFY-SYSTEM	Verify assembly object code in system software.

NOTE: All system level (TCL) commands are alphabetically listed and fully explained in the System Commands Guide.

Figure A. System and Data Base Troubleshooting Commands

5.11 SUMMARY OF SYSTEM UTILITY COMMANDS

A number of Ultimate system utility programs are available to assist in program debugging, resolve system problems, or to perform special user functions such as arithmetic or number conversion. These programs are executed from the TCL level as system commands.

In order to use a system utility command, the system must be at the TCL level, with the following prompt:

>

displayed on the terminal.

The system utility commands can be grouped by the type of service they provide. Figure A summarizes some utility commands used primarily for system management, debugging and troubleshooting. Figure B summarizes some utility commands used primarily by user accounts in day-to-day operations.

All system commands are alphabetically listed and fully detailed in the System Commands Guide.

Command	Operation
LOCK-FRAME	Lock a frame in memory.
UNLOCK-FRAME	Unlock a previously locked frame.
LIST-LOCKS	Display any BASIC group and/or item locks in effect.
CLEAR-BASIC-LOCKS	Clear the BASIC Lock Table. These 48 locks are manipulated with LOCK and UNLOCK statements; they are not group or item locks.
SET-TIME	Set system time.
SET-DATE	Set system date.
SET-LPTR	Set line length and depth for line printer.
PRINTRONIX	Set the number of lines in a form for Printronix printers.

... (continued)

Figure A. Some System Utility (TCL) Commands

Command	Operation
:INIT-SYSTEM	System manager only. Clears all locks, including group, item, BASIC, and other system locks.
:RESTARTLINE	System Manager only. Restarts specified line with the Logon prompt.
:TRAP :UNTRAP	System debugging tool. Set trap bit. Unset trap bit.
:ZLINKED	Zero out the chained link of Available Space (Overflow) frames.
:SET-MAX-LINES	Limit number of lines that can be logged onto system.
LIST-SYSTEM-ERRORS	List disk errors.
LOOP-ON	Re-execute specified TCL command in a loop. Exit only via System Debugger.
REV	Display the current release of the system
SET-BAUD	Set the baud rate and other characteristics for a line.
SET-PRIO	Display priority on specified line.
ECHO-ON ECHO-OFF	Allow or suppress display of keyboard input onto screen.
BREAK-KEY-ON BREAK-KEY-OFF	Allow or suppress the BREAK key function.
LINK-WS	Link workspace for one or more user Processes.
DUMP	Display the specified frame or group.
POVF	Display the Overflow pool of frames.
WHO	Display the current line and account.
WHAT	Display the system configuration.
WHERE	Display the terminal lines being used.

Figure A (cont.) Some System Utility (TCL) Commands

Command	Operation
CHARGES	Print computer usage since logon.
CHARGE-TO	Assign logon charges to another account.
ADDD	Add decimal numbers.
ADDX	Add hexadecimal numbers.
DIVD	Divide decimal numbers.
DIVX	Divide hexadecimal numbers.
DTX	Convert decimal number to hexadecimal.
MULD	Multiply decimal numbers.
MULX	Multiply hexadecimal numbers.
SUBD	Subtract decimal numbers.
SUBX	Subtract hexadecimal numbers.
XTD	Convert hexadecimal number to decimal.
RTD	Convert number in specified radix to decimal.
DTR	Convert decimal number to equivalent value in specified radix.
DELETE	Delete a file item.
SET-FILE	Set up a synonym file definition item.
EXCHANGE	Reverse (exchange) two item-ids.
SEARCH	Search an item for a specified string.

Figure B. Some User Utility (TCL) Commands

5.12 SYSTEM ERROR REPORTING AND INTERPRETATION

Various errors detected by the system during on-line operation are logged in the SYSTEM-ERRORS file. Each type of error is described in this section. A report of these errors can be generated by entering a LIST-SYSTEM-ERRORS command from TCL on the SYSPROG account.

NOTE: The occurrence of certain types of errors on an occasional basis is acceptable.

To list the current contents of the SYSTEM-ERRORS file:

1) Logon to SYSPROG.
2) At SYSPROG main menu - press the carriage return.
3) At TCL (> prompt) - enter the LIST-SYSTEM-ERRORS command and press the carriage return.

   >LIST-SYSTEM-ERRORS <CR>

   To the Printer (Y/N) ?Y <CR> An answer of 'Y' will route listing to the printer; otherwise it will print on the CRT.

   System Error listing explanation (Y/N) ?Y <CR> An answer of 'Y' will print this section (5.12).

The System errors report is broken down into four unique sections. The SYSTEM-ERRORS file logs:

1) disk errors,
2) EDAC corrected memory errors detected by the disk controller,
3) EDAC corrected memory errors detected by the CP,
4) illegal MLCP channel errors detected by the Monitor.

Disk Errors

Disk errors are logged when a disk error status is detected by the Ultimate on-line Monitor. Both recovered and non-recovered errors are logged.

A recovered error occurs when the disk controller issues an I/O command to the disk drive and an error status is returned. The system will retry the disk operation a number of times. This type of error may occur occasionally under normal operation of the system and is not considered a problem. Only when the frequency of this type of error increases is there need for concern.

A non-recovered error occurs after the system has retried the disk operation a number of times without success. The on-line Monitor will then re-issue the command to perform the I/O to the disk. The system will continue to retry the disk operation until it has been completed successfully. In other words, bad data on the disk will never be accepted by the system.

Information Reported in Disk Errors

The following is the layout of the SYSTEM-ERRORS listing at the terminal:

```
TIME       DATE     STAT  STAT  CHAN DRIVE PLATTER CYL HD SEC
                    1     2     #    #     SELECT (IN DECIMAL)
                                           BIT
11:54:20  03/30/83 9000  0001  0640  00      1        6  3   4
                   ----  ----  ----
                     |     |     |
                     |     |     |
                     v     |     v
Honeywell status word 1    |  Honeywell channel # or
DEC-based system           |  DEC-based system drive
drive status               |
                           v
          Honeywell status word 2 or
          DEC-based system error status
```

Each column reports the following:

TIME = The time of each error, in military format

DATE = The date of each error

STAT1 = is status word one on Honeywell-Based systems (i.e., the first status word reported by the disk controller); on DEC-Based systems, STAT1 shows the drive status word

STAT2 = is status word two on Honeywell-Based systems (i.e., the second status word reported by the disk controller); on DEC-Based systems, STAT2 is the error status word

CHAN# = On Honeywell-Based systems, this is the channel address of the disk drive with an error (note that reads and writes are performed on different channels in the same drive, i.e., channel 600 = read; channel 640 = write); on DEC-Based systems, CHAN# shows the control status word

DRIVE# = Logical drive number of the disk drive with error

PLATTER SELECT BIT = Used with CMD disk drive to determine if an operation is on a fixed or removable platter. Bit=1 for fixed disk.

CYL = Cylinder number in decimal

HD = Head number in decimal

SEC = Sector number in decimal

The following additional information is also given if the report is routed to the printer. (These columns are appended on the printed report, after the columns shown above.)

READ CONFIG WD A	READ CONFIG WD B	OUTPUT CONFIG WD A	OUTPUT CONFIG WD B	TASK WORD	FRAME ID (DEC.)	SECTOR ID	MEMORY WRDADDR (IN HEX)
0244	0528	0244	0520	A900	352787	02440500	1FF00
0244	0528	0244	0520	A900	352787	02440500	24900
0244	0528	0244	0520	A900	352784	02440500	36C00
028D	0000	028D	0000	A900	396992	00000000	2D400
0244	0528	0244	0520	A900	352787	02440500	26200
0248	0528	0248	0520	A900	355216	02480500	1E300

These columns report the following:

READ CONFIG WD A = Read configuration word A - configuration word returned from the disk controller including platter select bit.

READ CONFIG WD B = Read configuration word B - configuration word returned from the disk controller

OUTPUT CONFIG WD A = Output configuration word A - configuration word sent to the disk controller without platter select bit

OUTPUT CONFIG WD B = Output configuration word B - configuration word sent to the disk controller

TASK WORD = Task issued to the disk controller

FRAME ID = Frame identification number in decimal

SECTOR ID = Sector identification

MEMORY WRDADDR = Memory word address of data transfer, especially significant with EDAC recovered errors. This information can be used to determine which memory board is generating EDAC errors. (Honeywell-based systems only.)

Interpretation of Disk Errors

Interpreting the meaning of disk errors can somtimes be complex, but there are some general trends that you can look for in the error log that will help you to interpret the disk problem you are having.

The first trend to look for is consistency of cylinder numbers. When the cylinder numbers are the same and the heads are the same this is usually an indication of a media or disk pack problem. Formatting the pack and/or manually assigning that cylinder as defective will resolve this type of problem.

On Honeywell-based systems, the AD-ALT command in the Diagnostics Monitor can also be used to assign an alternate track without formatting and restoring.

If the cylinder numbers are random, check if the head numbers are the same. If so, this indicates a possible weak head in the disk drive and a service call will be necessary.

Another trend to look for is consistency in the times at which the errors occur. If several errors are encountered at the same time every day, you may be experiencing some power problems due to local electrical loads in your area. Caution should be taken when looking at and interpreting this trend. The times of the error may be consistent from day to day simply because you do a file-save at the same time every day, etc.

If the data associated with the errors are random, diagnosing the problem becomes more complex. Generally, service would be necessary and the problem could be anywhere in the disk sub-system. The disk sub-system includes the disk controller, disk cables, and the physical drive (both mechanical and electronic). The status words now become helpful. To interpret the status words, the hexadecimal numbers must be broken down into binary and correlated to the following tables.

<u>Disk Status Words on Honeywell-based Systems</u>

Status word #1 on Honeywell-based systems:

Bit #	Description	Bit #	Description
0	Device Ready	8	Missing Clock Pulse
1	Attention	9	Successful Recovery
2	Overrun/Underrun	A	Not Used
3	Device Fault	B	Not Used
4	Read Error	C	Corrected Memory Error
5	Illegal Seek	D	Nonexistent Resource
6	Missing Data Syne	E	Bus Parity Error
7	Unsuccessful Search	F	Uncorrected Memory Error

Status Word #2 on Honeywell-Based systems

Bit #	Description	Bit #	Description
0	Corrected Read Error	8	No Head Select
1	Successful Retry	9	Write Fault
2	Overrun/Underrun Recovery	A	Write or Read Off Cylinder
3	RFU (Reserved for	B	Write and Read Fault
4	RFU Future Use)	C	Voltage Fault
5	RFU	D	Head Select Fault
6	RFU	E	Seek Error
7	RFU	F	Write Protected

For example (on Honeywell-based systems):

```
STAT1 8040 1000 0000 0100 0000 -
     Device Ready, Successful Recovery

STAT2 8000 1000 0000 0000 0000 -
     Corrected Read Error
```

Disk Status Words on DEC-based Systems

NOTE: On DEC-based systems, bit 00 is the least significant bit and bit 15 is the most significant bit. In other words, the bit layout is:

DEC-BASED SYSTEM BIT LAYOUT

```
-------------------------------------------------
| 15 14 13 12 11 10 09 08 07 06 05 04 03 02 01 00 |
-------------------------------------------------
```

DEC systems may be configured with RK06 or RP02 disk drives. The bit usage for RP02 disk drives is the following:

bit	DRIVE STATUS (STAT1)	bit	ERROR STATUS (STAT2)	bit	CONTROL STATUS (CHAN#)
00	Attention bits	00	disk error	00	Go
01	" " "	01	Overflow error	01	Command bits
02	" " "	02	Non-existent memory	02	" "
03	" " "	03	Write check error	03	" "
04	" " "	04	Always zero	04	Memory extended address
05	" " "	05	Checksum error	05	" "
06	" " "	06	Always zero	06	Interrupt on
07	" " "	07	" "	07	Ready
08	Write protect	08	" "	08	Device select
09	Drive fault	09	Not used	09	" "
10	Seek underway	10	Program error	10	" "
11	Seek error	11	Nonexistent sector	11	Header
12	Always zero	12	Always zero	12	Always zero
13	Select unit RP03 if set	13	Seek error or non-existent cylinder	13	Attn. interrupt enable
14	Unit selected	14	File unsafe	14	Hard error
15	Unit ready	15	Write protect error	15	Error

For example (on DEC-based systems):

```
DRIVE STATUS C000 - unit ready; unit selected
ERROR STATUS 0800 - seek error
CONTROL STATUS C084 - error; hard error; ready; command bit
```

The bit usage on RK06 disk drives is the following:

bit	DRIVE STATUS (STAT1)	bit	ERROR STATUS (STAT2)	bit	CONTROL STATUS (CHAN#)
00	Always one	00	Illegal function	00	Drive select
01	Always zero	01	Seek incomplete	01	Drive select
02	" "	02	Always zero	02	Drive select
03	Drive fault	03	" "	03	Always zero
04	Always zero	04	Zero or format error	04	Inhibit bus address
05	Seek error	05	Drive type error	05	System clear
06	Volume valid; or drive select	06	Hard ECC error	06	Input ready; always zero
07	Drive ready	07	Bad sector error	07	Output ready (always zero)
08	RK06/07	08	Header read error	08	Always zero
09	Always zero	09	Cylinder overflow	09	Zero or Multiple Drives
10	" "	10	Invalid disk addr.	10	Program error
11	Write protect	11	Write protect error	11	Nonexistent memory
12	Always zero	12	Always zero	12	Nonexistent drive
13	Zero or positioning in prog.	13	Operation incomplete	13	Always zero
14	Current drive attention or seek complete	14	Drive unsafe	14	Write check error
15	Always one	15	Data check	15	Always zero

EDAC Corrected Memory Error Detected by Disk Controller
(Honeywell-based systems only.)

EDAC is short for Error Detection And Correction. This feature was added to the newer and faster MOS memory to enhance system reliability. EDAC will correct a one bit failure in memory. When this type of memory error occurs, the integrity of your data is maintained and your system will continue to operate without problems. An EDAC error may be detected by the disk controller during disk write operations when data is being transfered from main memory to the disk.

Information Reported in EDAC Errors Detected by Disk

TIME = The time of each error is reported in military format

DATE = The date of each error is reported

STAT1 = Status word one, is the first status word reported by the disk controller

STAT2 = Status word two, is the second status word reported by the disk controller

WRDADDR = Memory word address of data transfer. This information can be used to determine which memory board is generating EDAC errors.

Interpretation of EDAC Errors Detected by Disk

An occasional EDAC corrected memory error is an expected result of normal system operation. If the number of EDAC errors begins to increase, generally service is warranted. Data integrity is not affected by these errors, but any substantial increase in the number of errors should be reported to the Ultimate Support Group.

EDAC Corrected Memory Error Detected by CPU

An EDAC error may be detected by the CPU during normal operations of the system. As the errors are detected, a counter is incremented. Approximately every hour, this counter is interrogated and if changed the increment is logged in the SYSTEM-ERRORS file.

Information Reported in EDAC Errors Detected by CPU

TIME = The time of each error, in military format

DATE = The date of each error

OF ERRORS = Number of errors detected

Interpretation of EDAC Errors Detected by CPU

An occasional EDAC corrected memory error is an expected result of normal system operation. If the number of EDAC errors begins to increase, generally service is warranted. Data integrity is not affected by these errors, but any substantial increase in the number of errors should be reported to the Ultimate Support Group.

Illegal MLCP Channel Errors

On Honeywell-based systems, whenever an I/O operation needs to be done on a terminal line, the Kernel (Monitor) issues the appropriate instructions to the particular MLCP channel and waits for response from the channel. If the device responds that it did not interrupt and does not need to input or output or break, the Monitor reports an illegal MLCP channel error. This type of error may indicate an excessive amount of noise being generated by the terminal. These errors are logged in the SYSTEM-ERRORS file for Support personnel to investigate.

Information Reported in MLCP Channel Errors

The following is the layout of the LIST-SYSTEM-ERRORS listing of MLCP errors at the terminal:

```
TIME       DATE      STAT  STAT   CHAN DRIVE PLATTER CYL HD SEC
                     1     2      #     #    SELECT (IN DECIMAL)
                                             BIT
11:54:20  03/30/86   23    RC     1400  00     1        6  3   4
                    ----  ----   ----
                      |      |     |
                      |      |     |
                      v      |     v
Honeywell status word 1      |  Honeywell channel # or
DEC-based system             |  DEC-based system drive
drive status                 |
                             v
          Honeywell status word 2 or
          DEC-based system error status
```

TIME = The time of each error is reported in military format

DATE = The date of each error

STAT 1 (or INT LEVEL) in Hex = Hardware Interrupt Level, usually X'0023'.

STAT 2 (or TRAN OR RECV) = error on Transmitter channel (TR) or Receiver channel (RC)

CHANNEL NUMBER = the MLCP channel number

The following information is for the Ultimate Support Group and is reported only if the LIST-SYSTEM-ERRORS report is sent to a printer.

CCB STAT	BIT MAP 00-15	16-31	32-47	48-63	64-79	80-95	96-111	112-127
0000	0000	0000	0000	0000	0000	0000	0000	0000
0000	0000	0000	0000	0000	0000	0000	0000	0000
0000	0000	0000	0000	0000	0000	0000	0000	0000
0000	0000	0000	0000	0000	0000	0000	0000	0000

These columns report the following:

CCB STAT = CCB Status

BIT MAP = Bit map of CCB Status

Interpretation of MLCP Channel Errors

There is a high probability that the MLCP control board on the CHANNEL NUMBER is generating an excessive amount of noise. Possible causes are:

(1) A cable does not conform to the Ultimate cabling specification. Consult the appropriate Tech Memo and check all the cables on that MLCP 8-way or 16-way.

(2) Bad MLCP control board. Either one or more "daughter" boards or the mother board is not functioning.

NOTES

APPENDIX A

SYSTEM ERROR MESSAGES

> The master list of Ultimate system error and informational messages is found in the ERRMSG file on the SYSPROG account. These messages are displayed by various system software, and have no standard numbering system or display format. Many messages do, however, fall into general categories, and many are displayed with an identifying number.

The table below summarizes the general message categories in message number order (which is also the item-id order) for easier user reference. Messages displayed without numbers are also included (in item-id order). Displayed message numbers are enclosed in square brackets (e.g., [89]), while message numbers not displayed are shown with no brackets (e.g., 2).

A full listing of all entries in the ERMSG file follows the summary tables.

In addition, messages associated with asynchronous and bisynchrous communications are explained in Appendix C. Messages associated with specific system software such as the EDITOR or BASIC are listed and explained in the appropriate user manual.

Message Type and/or Associated Software	Range of Message Numbers/Unnumbered
RECALL and other system command entry errors	2, and [3] through [120]
Bisync command processing errors	[121] through [151]
RECALL command entry errors	[151] through [173]
System command entry errors	[197] though [204]
System Assembler errors	[205] through [219]
EDITOR messages	220 through 223
System Assembler errors	[225] through [238]
System command messages	[239] through [260]
PROC command processing errors	[265] through [281]
System command messages	[282] through [431]
Transaction Logger messages	451 through 466
System command messages	520, [521] through [558]

RECALL command entry errors	[700] through [708]
Tape and Disk errors	720 through 753
System command messages	780 through 806
Async command entry errors	[911] through [913]
Assembler errors	950 through 954
DUMP command entry errors	[990] through [991]
Restore command entry errors	[992] through [994]
WHAT command messages	998 through 999
Spooler messages	[1004] through [1243]
Printer Diagnostics messages	1250 through 1256
Disk Diagnostics messages	1257
Memory Diagnostics messages	1259 through 1260
UltiNet error messages	[2001] through [2341]
RECALL forms output command entry errors	[5001] through [5012]
UPDATE command entry errors	[7001] through [7134]
BASIC command entry errors	B0, [B1] through [B220]
UltiPlot error messages	[G1] through [G94]
UltiWord error messages	WP-E1 through WP-E5

2 Uneven number of delimiters (' " \).

[3] Verb?
[5] The word "A" is illegal.
[7] A value must follow the HEADING, FOOTING, or GRAND-TOTAL connective.
[8] A window specification string must follow the WINDOW connective.
[9] System D-pointer missing
[10] File name missing
[11] Frame locked at location X'A'
13] Data level descriptor (file-name in dictionary) is missing.
[15] The file-name is preceded by an illegal connective.
[17] WITHIN is valid only in COUNT, LIST, SUM or STAT statements.
[18] The last word may not be a connective.
[19] A value without an attribute name is illegal.
[20] Error in the USING syntax.
[21] Meaningless item-id in statement
[22] TO before item-id valid only in a CHANGE statement
[24] The word "A" cannot be identified.
[25] WITH may not immediately precede a value.
[26] Attribute values may not both precede and follow an attribute name.
[27] The word WITH is missing.
[29] At least one item-id must be specified for a WITHIN-type statement.
[30] Error in M/DICT definition of the verb
[53] Item list required for WITHIN.
[71] An illegal connective modifies the word "A".
[72] The value "A" is meaningless.
[79] The number of separate AND clause sets cannot exceed 9.
[80] A system error has occurred in mode: A
This may be due to sort-key(s) preceding selection criteria.
[82] Your system privilege level is not sufficient for this statement.
[89] Quit accepted
[91] End tape check - A file(s)
[92] End of recorded data - A file(s)
[100] 'A ' is not.B
[111] Item 'A' is not on file.
[120] 'A' negative balance not permitted.
[121] Unexpected interrupt from device X' '
[122] Transmission aborted--RVI received
[123] Transmission aborted--EOT received
[124] Transmission aborted--disconnect sequence received
[125] Transmission aborted--ENQ count exceeded
[126] Transmission aborted--NAK count exceeded
[127] Transmission aborted--communications line was disconnected
[128] -------Waiting for communications line to be connected--------
[129] ---------------Communications line connected------------------
[130] Other station won line bid
[131] Line bid not accepted by other station
[132] Message correctly received
[133] Transmission aborted before message totally received
[134] No BISYNC communications pac found
[135] Transmission aborted--line control character in data
[136] Data has been lost because of truncation.
[137] Entering receive mode
[138] BSC controller already attached to line A

```
[139] Line already attached to BSC controller A
[140] BSC controller attached
[141] Attach the BSC controller
[142] LOAD-COMPAC execution not valid for DEC hardware.

 143  BSC#    Channel#

        A         B
        C         D
        E         F
        G         H
        I         J
 144  Bisync rev.    A. B
 145  Bisync is in use.

[146] No BSC auto-dial unit found.
[147] Auto-dial unit not attached to this BSC line.
[148] BSC auto-dial call attempt not successful.
[149] BSC auto-dial call placement successfully completed.
[150] No power signal reported from auto-dialer
[151] Timer expired, no response from auto-dialer.

[158] An illegal connective of the form "A" modifies "B".
[163] Attribute for SORT, BREAK-ON or TOTAL connective missing
[165] The A-correlative :'A'
      contains an illegal IF expression.
[166] The A-correlative attribute name 'A' is illegal.
[167] Missing terminal quote ("/') in A-correlative: 'A'.
[168] Illegal A-correlative: 'A'.
[169] Missing left paren in A-correlative: 'A'.
[170] Missing right paren in A-correlative: 'A'.
[171] Illegal substring operator in A-correlative: 'A'.
[173] Missing semi-colon in A-correlative.
[197] Cannot delete D-type, Q-type or C-type pointer 'A'
[199]  Insufficient work space for item 'A'.
[200] File name?
[201] 'A' is not a file name
 202  'A' not on file.
[203] Item name?
[204] File definition 'A' is missing.
[205] No statements to be assembled.
[206] 'A' assembled.
 207  Undefined symbols.
[208] Error in item-id list
[209] Columns must be in range 1-80.
[210] File 'A' is access protected.
[211] Mode 'A'               No assembled code can be found.
[212] Mode 'A'               has no FRAME statement.
[213] Mode 'A'               Location counter error at line no. E
[214] Mode 'A'               overflows frame B at line # E
[215] Mode 'A'               Hex error at line no. E
[216] Mode 'A'           loaded;    Frame =    B  Size =   C  Cksum = D
[217] Mode 'A'           verified;  Frame =    B  Size =   C  Cksum = D
[218] Mode 'A'         Frame =    B  has   F  mismatches
[219] Illegal command:  A
```

```
 220   'A' exited
 221   'A' filed.
 222   'A' deleted.
 223   'A' exists on file.

[225] TSYM modulo of A required
[226] Tape format error
[227] Frame #     A    -  verified
[228] Frame #     A    -       B mismatches
[229] 'A' is not a valid account name.
[230] Card reader not ready.
[232] Card reader EBCDIC error.
[233] Card reader hopper empty.
[234] Item size exceeds 32,000 bytes.
[235] Attempt to write into update protected file
[236] No errors
[237] Errors found
[238] Previous errors found
[239] File 'A' is update protected
[240] A SELECT-type statement must immediately precede SAVE-LIST.
 241  Successful compile; A frames used.
[242] 'A' decataloged.B
 243  'A' saved - B frames used.
[244] 'A' cataloged.
[250] 'A' updated.
[251] 'A' options = B
[260] Invalid Batch lock command: A
[265] PROC stack overflow
[267] PROC transfer to 'A' cannot be completed.
[268] The destination of the PROC GO statement:
      A, cannot be found
[270] Format error in the PROC statement:
      A
[272] A value exists for the attribute referenced by the element:
      A
[273] Error in column-number/field-width or format specification at :
      A
[274] Unrecognizable Batch-string element :
      A
[275] Y or F sub-element error at Batch-string element :
      A
[276] D-2 update without D-1 being specified, at Batch-string element :
      A
[277] J element missing at Batch-string element :
      A
[278] Error in processing secondary Batch-string element :
      A
[279] Incorrect scaling factor in F* Batch-string element :
      A
[280] File-definition Batch element error at :
      A
[281] D1 must have Y11 storage correlative ... error at :
      A
[282] Invalid parameter for SELECT list
```

```
 289                Terminal Printer
     Page width:        A        B
     Page depth:        C        D
     Line skip :        E
     LF delay  :        F
     FF delay  :        G
     Backspace :        H
     Term type :        I

[290]  The range of the parameter "A" is not acceptable.
[291] 'A' file-definition is missing.
 293  Total number of contiguous frames    :A
[294] A additional task workspaces initialized
 295  Firmware rev.      A
      Kernel rev.        B
      Async rev.         C
      Abs rev.             D
      Diags rev.         E
[298] Format error in specifications.
 300  Tape-copy complete - A files copied
 301  End of recorded data
      Tape-copy complete - A files copied
 302  From/to cannot be the same device!
      Tape-copy complete - A files copied

 308  Module size: A frames

[316] Numeric parameter missing
[331] The ACC file is missing.
[332] This line may not log on to the specified account.

 335  <<<  Welcome to the Ultimate Computer System  >>>
      <<<  Copyright July,1981  The Ultimate Corp.  >>>
      <<< 14:44:33 Release 10 Rev 170 20 MAR 1986 >>>

 336  < Logged off at 14:44:33        on 20 MAR 1986             >

[337] User is not logged on.
[338] Account file statistics were not updated due to either:
      1. Insufficient work-space to contain the account file item, or
      2. System dictionary changed while you were logged on.
[339] Illegal TCL level

 340  < Connect time= A mins.; CPU= B units; lptr pages= C       >

[341] Ultimate system software verified.
[342] *** Ultimate system software does not verify! ***
 There are A frames with mismatches.
[350] Bad account name or message body
[351] Bad line number
[352] Another user is issuing a message; please do it later.
[398] The file D-pointer requires a V(ertical) correlative for this statement.
[399] The maximum of 20 levels for a WITHIN-type statement has been exceeded.
[401] No items present.
```

403 End of list
404 A items selected.
405 A items listed.
406

Item count=	A,	byte count=	B,	avg. bytes/item=	
avg. items/group=	D,	std. deviation=	E,	avg. bytes/group=	

407 A items counted.
408 One item counted.

[409] 'A' is not on file.
[410] A synonym (Q-type) file cannot be specified in this statement.
[411] DICT or DATA must be specified in a CLEAR-FILE statement.
[412] Insufficient disk space available for the file.
[413] The file name already exists in the master dictionary.
[414] Illegal or missing modifier used in defining the file area(s)
[415] 'A' exists on file.
[416] Range error in modulo or separation parameter.
[417] File 'A' created; base = B, modulo = C, separ = D.
[418] File definition item 'A' not copied
[419] The specified file cannot be cleared or deleted.
[420] Dictionary file deletion cannot be done without deletion of data-section(s) first.

421 Statistics of A :
Total = B Average = C Count = D

422 Byte statistics for : A
Total = B Average = C Items = D Cksum = E Bits = F

423 Total of A = B

[426] Data file already exists.
[427] There is no data section for this file.
[428] 'A' object pointer not found
[429] 'A' program object verifies
[430] 'A' program object does not verify
[431] Pointer item 'A' not copied

451 Invalid option

452 Logger assigned to port A

453 Transaction logger options:

1. Activate logger; start tape
2. Deactivate logger; exit menu
3. Suspend tape
4. Restart tape
5. Change tape attachment parameters

Enter option or <CR> to display status:

454 Not suspended

455 Already suspended
456 Logger already active
457 Logger not active
458 Tape suspended
459 Tape restarted
460 Logger deactivated
461 Logger status: Inactive
462 Logger status: Tape A; seq#B
Logger assigned to port C
Disk queue permanent frames used: D
Disk queue overflow frames used: E
Latest transaction# in disk queue: F 14:44:34 20 MAR 1986
Oldest transaction# in disk queue: G H I

463 T-ATT parameters:

464 Any transactions now queued on disk will be lost.
Continue (Y/N)?

465 Blocksize must be an even number between 30 and 8192

466 Warning: 200 more frames added to overflow queue

520 what?........

[521] Too many characters in word to block
[522] BLOCK-CONVERT file missing or improperly defined
[523] Block output would exceed page width.
[524] Item not found in BLOCK-CONVERT file
[525] Input character 'A' is improperly formatted in BLOCK-CONVERT file
[526] Illegal account name.
[527] Unable to logoff; terminal status unknown.
[528] Another user is doing LOGON/LOGOFF/RESTARTLINE. Do it later.
[529] Already logged off.
[530] Already logged on
[531] Process roadblocked
[532] Unable to logon the line to the account.
[533] Logon successful
[534] Logoff successful
[535] Illegal line number.
[536] Illegal baud rate
[537] RESTARTLINE successful.
[538] Use OFF verb to log off yourself.
[539] Trap/untrap line successful
[540] The line must be trapped before a peripheral read or write.
[550] A required numeric parameter is missing or invalid.
[551] Specify number of lines to skip.
[552] Item 'A' has invalid format
[553] Illegal parity option
[554] Illegal bit length
[555] Illegal baud rate
[556] Illegal echo parameter
[557] Illegal XON parameter
[558] Illegal typeahead parameter

[700] Run-time F-correlative abort.
[701] Invalid function correlative definition : A
[705] Illegal conversion code : 'A'
[708] 'A' cannot be converted.

720 Tape A
721 attached to line #
722 A not attached
723 , on-line
724 , off-line
725 , set at 1600 bpi
726 , set at 800 bpi
727 , write protected
728 , write permit

729 Disc cartridge
730 , not formatted for Ultimate
731 , old Ultimate format

732 Tape
733 , device busy
734 , set at 6250 bpi

[780] Item 'A' not on file.

781 'A' added
782 'A' updated
783 'A' deleted

802 A items dumped.
803 A item(s) loaded
805 A items copied
806 A items updated.

[911] Async-2780 already attached to line A
[912] Not valid on DEC-based systems
[913] Async-2780 not attached.

950 Linkfield error - group at A - Frame B
Links: C D E F
951 Illegal forward link. Group at A Frame B
Links C D E F
952 Illegal backward link. Group at A Frame B
Links C D E F
953 Should link back to: group at A Frame B
Links C D E F

954 A new error items created in TSYM

[990] Error in ABS-DUMP frame limits specifications.
[991] S-module A is missing
[992] A item(s) have been restored
[993] Account name must be specified.
[994] Full file save required

```
 998  Memory   PIBs     PCB0 Wsstart  Wssize Sysbase     Mod      Sep  Maxfid
      System time  System date

 999  Memory PIBs Lines  On PCB0 Wsstart Wssize Sysbase/mod/sep Maxfid     Ov
          AK     B     C  D    E       F      G        H   I   J

[1004]'A' not on file
[1099] Hold entry #
[1100]
 Start code locked.
[1101] Null printer number.
[1102] Printer number too big.
[1103] No form number.
[1104] Illegal character.
[1105] Printer must be stopped.
[1106] Form number too big -- exceeds 125.
[1108] Negative number.
[1109] Too many queues.
[1110] Illegal printer type -- not P or S.
[1111] Illegal line number or parallel printer number.
[1112] Illegal parallel printer number.
[1113] Illegal serial printer number.
[1114]
 The device or line which you specified is being used
 as another printer on the system.
[1115] Allocation attempted on uninitialized printer.
[1116] There is no job enqueued for output on the forms you specified.
 Alignment is not possible.
[1117] Your align was aborted by someone.
 Please start the align process over.
[1118]
 The printer control block has been initialized.
 The correct paper and LPI settings must have been
 previously set to insure proper printing.
[1119] You are attempting to start printer  A on line  B, which is not
stopped.

[1121] You are improperly logged on.
[1122] Your output specification is no output.
 Reassign your line if you wish to output a holdfile.
[1123] Line # A is already logged on.
[1126] Printer #   A Control block in a questionable state.....reset to null
[1127] Illegal printer number.  Must be 0-7.
[1129] A form queue elements unlinked
[1130] Printer list elements                     20 MAR 1986 14:44:46
      Stat Lk Ln       Curpos   Begfid Cp Fo  Frms    Date      Time   Acct
[1131]     Printer list elements                20 MAR 1986 14:44:46
      # Stat Lk Ln Status           Cp Fo  Frms   Date      Time   Acct
 1132  A queue elements.
 1133  A frames in use.
 1134  Printer assignments                                   14:44:46
       Printer         Output queues     Page  Dev or    Status
       Type   Number                     skip  line #
[1135] Form queue    A
[1140] Your open files were closed
```

```
 1141  Line  Status Cop  Form
          #         ies     #
[1143] Align terminated; printer stopped.
[1144] The tape is not available for use just now.
[1145] Illegal specification number A.
[1147] Not attached.
[1148] Tapeout terminated because of assign T or null assignment.
[1149] Fatal spooler abort!
[1150] There is something wrong with the syntax of your verb's options.
[1151] Entry #   A[1160]
       Your output specification is no output.
       Reassign your line if you wish to output a holdfile.
[1161] End of requested print files.
[1162] End of print file control block
[1169] Illegal printer number--must be between 0 and 19 inclusive.
[1170] Printer #  A set to stop and is inactive.
[1172] but is still active.
[1173] Printer # A control block in a questionable state.....reset to null.
[1174] is unallocated.
[1175] Parallel printer # A has been deleted.
[1176] Serial printer # A has been deleted, and its process sent to Logon.
[1177] Printer # A is inactive.
[1178] The job being output on printer # A is not your print file.
[1179] Job aborted on printer #  A
[1180] Print file #   A was not unlinked because it is being output.
[1181] Print file # A was not unlinked because it is unused.
[1182] Print file # A was not unlinked because it is not spooled.
[1183] Print file # A was not created on the account onto which you are
       now logged.
[1184] Print file # A was unlinked and is available as a hold file.
[1185] Can't SP-KILL choked job--kill it by hitting BREAK key at the terminal
[1190] Logical printer stopped.
[1200] The spooler is inactive.
[1201] The spooler is active.
[1202] Needs to start printers
[1203] Needs to log disk errors.
[1209] The control block for printer # A is in an ambiguous state.
       Delete the printer from the spooler system.
[1210] Printer # A is
 1211  unallocated.
 1212  serial
 1213  parallel
 1214  , inactive
 1215  , active
 1216  , stopped
 1217  , and on line.
 1218  , and off line.
 1219  The printer cable is off.
[1220] There is no controller for this printer.
[1221] The printer is defined as parallel printer # A.
[1222] The printer is running on line A.
[1229] Print file being output is element A, an open file for line # A
 closed file for line # A
[1232]   generated on account A, which is A frames long.
 and the output is choked.
```

[1239] Erroneously, the printer has no output queues assigned to it.
[1240] Assigned output queues: A, A, A.
[1243] The number of inter-job pages to eject is A.

1250 This system does not have parallel printers.
1251 Enter printer number to test or <CR> if you do not wish to proceed
1252 Printer controller busy; does not acknowledge I/O instructions.
Unable to get status of printer.
Test aborted.
1253 Printer is not ready. Printer status word is X'A'
1254 Does not get interrupt from printer controller within 10 seconds.
Test aborted.
1255 Illegal printer number. Test aborted.
1256 On-line printer test
These printer controller(s) are numbered 0 (and 1).
(0 = printer at lowest channel; 1 = printer at highest channel.)

The printer test does the following:
(1) Check the status of the printer controller;
(2) If printer is ready, print 50 lines on the printer;
(3) If printer is not ready, display the status returned from controller.

This system has A parallel printer controller(s).

Enter printer number to test or <CR> if you do not wish to proceed

1257 On-line non-destructive disk test
This test has six (6) phases:
Phase 1: read frames from 1 to max in 1000 frame increments.
Phase 2: write frames as above (no data is altered).
Phase 3: read frames from 1 to max in 100 frame increments.
Phase 4: write frames as above (no data is altered).
Phase 5: read all frames from 1 to max in sequential order.
Phase 6: write all frames as above (no data is altered).

You are to look for two 'symptoms':
(1) If the system hangs on a particular frame number, that probably means that there is a bad spot on that frame. You will have to do a cold-start to bring the system back on line.
(2) If you see an ampersand sign (&), that means that you have a disk error. Break-and-END this test. You'll get back to the main menu automatically. Then, select program "List system errors to print the disk error report.

It takes approximately 20 seconds to test 1000 frames.
This system has A frames.
You may break-and-END this test at any time.

Enter Y to start the test or <CR> if you do not wish to proceed

1259 On-line Memory Test
This test exercises the memory by writing a test pattern in memory.
The test pattern is subsequently checked for errors.
Due to the virtual memory management scheme employed by the Ultimate

operating system, the memory test: (1) requires frequent access to disk and (2) is not able to report the exact real memory location in case of errors.

You are to look for three 'symptoms':
(1) an ampersand sign (&) is displayed, which means that you have either a disk error or an EDAC error detected by disk controller.
(2) an error message "MEMORY ERROR" is displayed, which means that you have a "hard" memory error.
(3) on Honeywell-based systems, EDAC errors are reported in real memory location X'1F'. EDAC errors are also logged by the system automatically.
You can break-and-END this test at any time. You'll get back to the main menu automatically. Select program "List system errors" to print the report.

This system has AK memory. This test runs indefinitely.
You may break-and-END this test at any time.

Enter Y to start the test or <CR> if you do not wish to proceed

1260 Insufficient disk space for memory test program.
Test aborted.

[20C1] Aborted to debugger at Remote System
Referencing Frame Zero
System = A, Abort @ B

All Remote Files now closed at this System.
[2002] Received Illegal (Data) Response
from File Server at System 'A'.

All Remote Files now closed at this System.
[2003] Received Illegal Reply to CONNECT-REQUEST from System 'A'.
Remote File has not been opened at this System.
[2004] Remote File Access Denied -- Network Initialization not complete.
[2005] Time-out to a CONNECT-REQUEST for Host 'A'.
Remote File has not been opened at this Host.
2006] Exceeded Maximum Number of Remote Files that can be Opened.
2007] Time-out to a DATA-TRANSFER-REQUEST for Host 'A'.
All Remote Files have been closed at this Host.
2008] Received Illegal Q-Entry during Data Transfer Phase
from File Server at Host 'A'.
All Remote Files now closed at this System.
2009] Group Format Error at Host 'A'
B
2010] File Server received an Illegal Q-Entry Qualifier at System 'A'.
All Remote Files now Closed at this System.
2011] File Server received illegal Indicator Field at System 'A'.
All Remote Files now closed at this System.
2012] File Server at System 'A' received illegal Primary Command.
All Remote Files now closed at this System.
2013] File Server's Connect Table is full at System 'A'.
2014] File Server received a duplicate CONNECT-REQUEST at System 'A'.
All Remote Files now Closed at this System.

[2017] Received Q-Entry from line other
than File Server Process at System 'A'.
All Remote Files now closed at this System.
[2018] Received Response from System 'A'
with incorrect TO/FROM-ADDRESS.
All Remote Files now closed at this System.
[2019] Received a Command from the File Server at System 'A'
with incorrect TO/FROM-ADDRESS.
All Remote Files now closed at this System.
[2020] Aborted to debugger at Remote System
Crossing Frame Limit
System = A, Register = B, Abort @ C
All Remote Files now closed at this system.
[2021] Aborted to debugger at Remote System
Forward Link Zero
System = A, Register = B, Abort @ C
All Remote Files now closed at this System.
[2022] Aborted to debugger at Remote System
Backward Link Zero
System = A, Register = B, Abort @ C
All Remote Files now closed at this System.
[2023] Aborted to debugger at Remote System
Referencing Illegal Frame
System = A, Abort @ B
All Remote Files now closed at this System.
[2024] Response from File Server at System 'A' has a format error.
All Remote Files now closed at this System.
[2025] Update Rejected at System 'A' - D/CC/CL Pointer.
[2031] File Server at System 'A' received Invalid File-entry #.
All Remote Files now closed at this System.
[2032] File Server received a File-entry # assigned
to another user at System 'A'.
All Remote Files now closed at this System.
[2034] System 'A' out of Disk Space.
Try Again or Quit (A/Q) ?
[2036] File Server at System 'A' received illegal Secondary Command.
All Remote Files now closed at this System.
[2037] Illegal Status Field received in
Response from File Server at Host 'A'.
All Remote Files now closed at this System.
[2040] Primary Command was incorrectly echoed from System 'A'.
All Remote Files now closed at this System.
[2041] Secondary Command was incorrectly echoed from System 'A'.
All Remote Files now closed at this System.
[2042] Received Response of incorrect length from System 'A'.
All Remote Files now closed at this System.
[2043] Reserved for Connect ERRMSG's
[2044] Reserved for Connect ERRMSG's
[2045] Reserved for Connect ERRMSG's
[2046] Reserved for Connect ERRMSG's
[2047] Reserved for Connect ERRMSG's
[2048] Reserved for Connect ERRMSG's
[2049] Reserved for Connect ERRMSG's
[2050] Reserved for Connect ERRMSG's
[2051] Reserved for Connect ERRMSG's

[2052] Reserved for Connect ERRMSG's
[2053] Reserved for Connect ERRMSG's
[2054] Reserved for Connect ERRMSG's
[2055] Reserved for Connect ERRMSG's
[2056] Reserved for Connect ERRMSG's
[2057] Reserved for Connect ERRMSG's
[2058] Reserved for Connect ERRMSG's
[2059] Reserved for Connect ERRMSG's
[2060] Reserved for Connect ERRMSG's
[2061] Reserved for Connect ERRMSG's
[2062] Reserved for Connect ERRMSG's
[2063] Reserved for Connect ERRMSG's
[2064] Reserved for Connect ERRMSG's
[2065] Reserved for Connect ERRMSG's
[2066] Reserved for Connect ERRMSG's
[2067] Reserved for Connect ERRMSG's
[2068] Reserved for Connect ERRMSG's
[2069] Network Controller has cleared connection with System 'A'.
All Remote Files now closed at this System.
[2070] No connection with System 'A'.
[2071] Link with Network Controller is down.
All Remote Files now closed.
[2072] Link with Network Controller has been reset.
All Remote Files now closed.
[2073] Network Controller cannot make connection with System 'A'.
[2074] No more logical channels available on Network Controller.
Remote File Access cannot be supported.
[2075] Comm-Spooler Connect Table at System 'A' is full.
Remote File Access cannot be supported.
[2076] Duplicate CONNECT-REQUEST received for System 'A'.
All Remote Files now Closed at this System.
[2077] Comm-Spooler Connect Table is full.
Remote File Access cannot be supported.
[2078] Reserved for Connect ERRMSG's
[2079] Reserved for Connect ERRMSG's
[2080] Reserved for Connect ERRMSG's
[2081] Reserved for Connect ERRMSG's
[2082] Reserved for Connect ERRMSG's
[2083] Reserved for Connect ERRMSG's
[2084] Reserved for Connect ERRMSG's
[2085] Reserved for Connect ERRMSG's
[2086] Reserved for Connect ERRMSG's
[2087] Reserved for Connect ERRMSG's
[2088] Reserved for Connect ERRMSG's
[2089] Reserved for Connect ERRMSG's
[2090] Reserved for Connect ERRMSG's
[2091] Reserved for Connect ERRMSG's
[2092] Reserved for Connect ERRMSG's
[2093] Reserved for Connect ERRMSG's
[2094] Reserved for Connect ERRMSG's
[2095] Reserved for Connect ERRMSG's
[2096] Reserved for Connect ERRMSG's
[2097] Reserved for Connect ERRMSG's
[2098] Reserved for Connect ERRMSG's

```
 2099   REMOTE REMOTE                           CONNECT      CONNECT     CPU       LOCAL
        HOST#  LINE#  REMOTE HOST/ACCT.           DATE         TIME       UNITS     LINE#
        -----  -----  -----------------      --------     --------   -------    -----
```

[2100] Error - KXT11 controller in incorrect state.
[2101] DMA Error During Download to the KXT11.
[2102] SLOADed object code could not be found
[2103] Unable to startup program after download to the KXT11.
[2104] This Program will only work with a DEC computer.
[2105] Error - KXT11 command register has not been cleared.
[2106] Unable to re-intialize the KXT11.
[2107] No HDLC Controller on this System.
[2108] Error - KXT11 register 4 has not been cleared.

```
 2109   Remote   Remote                Local
        host#    Host name             Port#
        -----    --------------------  -----
```

[2200] File 'A' is referencing an Invalid Account name at Remote System.
[2201] 'A' is not a File Name at Remote System.
[2202] Illegal Second Level of Remote-Q found in File 'A' at Remote System.
[2203] Network Disconnect due to Warmstop.
[2210] File 'A' is Access Protected at Remote Host.
[2211] File 'A' is Update Protected at Remote Host.
[2213] Remote File 'A' has Missing/Access Protected Data
Level Descriptor (File Name in Dictionary) at Remote System.
[2215] Remote File Definition 'A' has missing Account Name.
[2331] Cannot Open the NETWORK File.
[2332] The "I-AM" item in the NETWORK file is missing.
[2333] 'A' is not a valid item in the NETWORK File.
[2334] Item 'A' is missing from the NETWORK File.
[2335] Improper format or data in the NETWORK file "I-AM" item.
[2336] You cannot initialize the Network,
The File Server has connections in use.
[2337] Workspace for Networking requirements unavailable.
[2338] Communications File Server processor initialized.
[2339] The System # item corresponding to this local system
is missing from the NETWORK File at System 'A'.
[2340] Network Initialization Successful.
[2341] Network Initialization Not Successful.

4321 Time is A

[5001] Parameter is less than zero or greater than page width in field:
A
[5002] Error in range of x, y, or z parameter in field:
A
[5003] Forms background item 'A' is not on file.
[5004] Format error in forms expression:
A
[5005] Missing colon in forms expression:
A
[5006] Dictionary attribute specified in forms expression:
A
is not on file

[5007] Substring format error in forms expression:
A
[5009] Invalid WINDOW or END-WINDOW:
A
[5010] Missing or invalid @(x,y,z) after the window:
A
[5012] Subpage specification less than one or greater than page depth
A
[7001] Right parenthesis ")" missing in update expression.
A
[7002] Left parenthesis "(" missing in update expression.
A
[7003] Invalid update expression.
A
[7004] DICT word in update expression not specified, or not in dictionary.
A
[7005] IDn item(s) must be specified before any other expressions.
A
[7006] The IDn item specified in the expression above is illegal,
or not in sequence.
[7007] Invalid secondary filename: A.
A
[7008] NEXT-SCREEN connective illegal here - no fields specified in this
screen, or within a window.
A
[7010] DICT word does not have a D/CODE of U or A in update expression.
[7011] An END-WINDOW is missing in the statement.
A
[7012] The step value in the window does not divide exactly into the number
of rows.
[7013] The dictionary of this file has not been validated. You must run
the UPD-VALIDATE program before using the UPDATE processor.
A
[7014] An attribute with a structure code can only be used in a window.
A
[7015] This attribute is illegal in this window because its C or D
structure code is missing or does not match others in the window.
A
[7016] Attribute without a structure code cannot be updated within a window.
A
[7017] Invalid conversion/correlative in above dictionary attribute.
A
[7018] Too many update expressions; either more than 26 with "A" option,
or > 75 total expressions, including windows and implicit fields.

[7020] Automatic item-id generator 'A' is missing from the dictionary.
[7102] Item with a zero AMC must have an A or F correlative.
[7103] Multiple correlatives are not allowed.
[7104] Multiple items with the same AMC, and no group correlative.
[7105] IDn item must have a zero AMC.
[7106] Invalid correlative
[7107] Invalid C or D structure code
[7108] C code references a non-existent dependent.
[7109] D code references a non-existent controller.
[7110] Improper IDn item; must be sequential with no gaps in G correlative.

[7111] Invalid G correlative; must be sequential with no gaps.
[7112] Invalid conversion code
[7113] F or A correlative references a missing AMC or attribute name.
[7114] Improper sorting of AMC's - ensure that the item A/AMC in your M/DICT is right-justified, and repeat the validation.
[7115] Link to the secondary file's item-id is missing.
[7116] Invalid storage code in the secondary file link.
[7117] The "store-to" attribute name specified in the secondary link is non-existent, or does not have a D/CODE of U.
[7118] The file specified in the secondary link is non-existent in this account.
[7119] The store code does not match the definition of the attribute in the secondary file.
[7120] External subroutine has not been cataloged in your M/DICT.
[7121] Controlling attribute cannot have a function code.
[7122] C code must have at least one dependent.
[7123] D code can have only one controller.
[7124] File cannot be opened in translate code.
[7125] Translate code must be C or X.
[7126] Translate code is illegal in an attribute with a function.
[7127] D, MR, ML, or MV codes must be the last conversion code.
[7128] More than 9 secondary files are not supported.
[7129] Store code mismatch: + or - to a secondary multivalue.
[7130] Store code mismatch: multivalued storage into a secondary single valu
[7131] Store code mismatch: SV storage into a secondary multivalue.
[7132] Storage into a secondary attribute with a function field is not allowed.
[7133] Storage into a secondary controlling or dependent attribute is not allowed.
[7134] Duplicate links to secondary file item-id.

B0 A cataloged.
[B1] Run-time abort at line A
[B3] Line A String length exceeds 32,266 characters
[B10] Line A Variable has not been assigned a value; zero used
[B11] Line A Tape record truncated to tape record length
[B12] Line A File has not been opened
[B13] Line A Null conversion code is illegal; no conversion done
[B14] Line A Bad stack descriptor
[B15] Line A Illegal opcode: C
[B16] Line A Non-numeric data when numeric required; zero used
[B17] Line A Array subscript out of range
[B18] Line A Attribute number less than -1 is illegal.
[B19] Line A Illegal pattern
[B20] Line A COL1 or COL2 used prior to executing a FIELD stmt; zero used
[B21] Line A MATREAD: number of attributes exceeds vector size
[B22] Line A Illegal value for STORAGE statement
[B23] Program 'C' must be recompiled.
[B24] Line A Divide by zero illegal; zero used
[B25] Program 'C' has not been cataloged.
[B26] Line A 'UNLOCK C' attempted before LOCK
[B27] Line A RETURN executed with no GOSUB
[B28] Line A Not enough work space
[B30] Line A Array size mismatch
[B31] Line A Stack overflow

[B32] Line A Page heading exceeds maximum of 1400 characters
[B33] Line A Precision declared in subprogram 'C' is different from that declared in the mainline program.
[B34] Line A File variable used where string expression expected
[B41] Line A Lock number is greater than 47.
[B100] Line A Compilation aborted; no object code produced
[B101] Line A Ambiguous ELSE clause
[B102] Line A Bad statement
[B103] Line A Label 'C' is missing
[B104] Line A Label 'C' is doubly defined
[B105] Line A 'C' has not been dimensioned
[B106] Line A 'C' has been dimensioned and used without subscripts
[B107] Line A LOOP statements nested too deep
[B108] Line A NEXT statement missing
[B109] Line A Variable missing in NEXT statement
[B110] Line A END statement missing
[B111] Line A EXIT used outside of LOOP statement
[B112] Line A REPEAT missing in LOOP statement
[B113] Line A Terminator missing
[B114] Line A Maximum number of variables exceeded
[B115] Line A Label 'C' is used before the EQUATE stmt.
[B116] Line A Label 'C' is used before the COMMON stmt.
[B117] Line A Label 'C' is missing a subscript list.
[B118] Line A Label 'C' is the object of an EQUATE statement and is missing.
[B119] Line A Warning - precision value out of range - ignored
[B120] Line A Warning - multiple precision statements - ignored
[B121] Line A Label 'C' is a constant and cannot be written into.
[B122] Line A Label 'C' is improper type.
[B124] Line A Label 'C' has literal subscripts out of range.
[B125] No source statements found; no object code produced
[B126] Line A ELSE clause missing
[B127] Line A NEXT missing
[B128] Line A Item 'C' not found
[B129] Illegal: program name same as dictionary item name
[B199] Source file must have separate DICT and DATA sections
[B209] Line A File is update protected.
[B210] Line A File is access protected.
[B220] Line A 'CSYM' is not a file name or needs a data level.

[G1] Maximum axis size out of range.
[G2] Plot size would require more than 10000 frames working storage. The job was terminated.
[G3] Requested plot width A characters is greater than device maximum.
[G4] Requested plot depth A is greater than device maximum depth.
[G5] Plot size would require A frames working storage. There are not sufficient frames available.
[G6] Out of overflow space!
[G7] Minimum value must be lower than maximum value.
[G11] Plot table definition item 'A' is not on file.
[G12] Plot table definition item 'A' has ended prematurely.
[G13] Plot table item 'A' has too few values.
[G14] Plot table item 'A' has too many values.
[G15] Plot table item 'A' contains non-numeric data.
[G16] Plot device definition item 'A' has illegal type code in line 1.
[G17] There are less than 50 lines in the message table 'A'.

```
[G21] A plot statement requires at least 2 plottable attributes.
[G22] Compiled string format error @ A
[G23] History string format error @ A
[G24] Illegal attribute sequence number.
      The specified number does not have the "bar" option selected.
[G25] Illegal attribute sequence number.
      The specified number is already stacked with another attribute.
[G26] Illlegal attribute sequence number.
      The number must be greater than 0.
[G27] Illegal attribute sequence number.
      The number must be lower than the current attribute.
[G28] Illegal attribute sequence number.
      The specified number is already grouped with another attribute.
[G29] The label area (and optional width) will exceed
      the avaliable plot width.
[G30] The label area (and optional depth) will exceed
      the available plot depth.
[G31] The required label area will cause the
      circle diameter to be less than 2 inches.
[G40] Please contact the Ultimate Corp. immediately.
      Code = 0,A,B,C
[G41] Please contact the Ultimate Corp. immediately.
       Code = 1,A,B,C
[G42] Please contact the Ultimate Corp. immediately.
       Code = 2,A,B,C
[G43] Please contact the Ultimate Corp. immediately.
       Code = 3,A,B,C
[G44] Please contact the Ultimate Corp. immediately.
       Code = 4,A,B,C
[G45] Please contact the Ultimate Corp. immediately.
       Code = 5,A,B,C
[G46] Please contact the Ultimate Corp. immediately.
       Code = 6,A,B,C
[G48] Please notify the Ultimate Corp.
      Serial number A will expire in B days. Code = C.

 WP-E1 Terminal type not known to Word Processor!
 WP-E2 \READNEXT file not \SELECTed
 WP-E3 \SELECT file name missing
 WP-E4 No \SELECT file open for ~INSERT
 WP-E5 Right margin in Ruler is greater than page width or is not present.
```

APPENDIX B

ASCII CODES

This appendix presents a list of ASCII codes for decimal number values from 0 through 255 (see DECIMAL column). The hexadecimal equivalent value and ASCII character generated are given (see HEX and CHARACTER columns).

Note that decimal values 0-31 are assigned as non-printable functions. Decimal values 1-26 may be specified by control key sequences (see TERMINAL KEY column). A "control key sequence" is entered by holding down the <CTRL> key while pressing a second key (e.g., <CTRL>A). Some of the non-printable characters have a special use in Ultimate systems (see SPECIAL USE IN ULTIMATE column).

Decimal values above 127 (Hex '7F') are not defined in the ASCII character set. The functions or characters assigned to these values are dependent on the terminal being used. However, in the Ultimate system, special file structure functions and control key sequences have been assigned to decimal values 251 through 255 (Hex 'FB' through 'FF').

DECIMAL	HEX	CHARACTER	SPECIAL USE IN ULTIMATE	TERMINAL KEY
00	00	NUL		
01	01	SOH		<CTRL>A
02	02	STX		<CTRL>B
03	03	ETX		<CTRL>C
04	04	EOT		<CTRL>D
05	05	ENQ		<CTRL>E
06	06	ACK		<CTRL>F
07	07	BEL	Bell on terminal	<CTRL>G
08	08	BS	Backspace	<CTRL>H
09	09	HT	Tab	<CTRL>I
10	0A	LF	Line feed on terminal	<CTRL>J
11	0B	VT		<CTRL>K
12	0C	FF		<CTRL>L
13	0D	CR	Carriage return on terminal	<CTRL>M
14	0E	SO		<CTRL>N
15	0F	SI		<CTRL>O
16	10	DLE		<CTRL>P
17	11	DC1		<CTRL>Q
18	12	DC2	Retype entire line	<CTRL>R
19	13	DC3		<CTRL>S
20	14	DC4		<CTRL>T
21	15	NAK		<CTRL>U
22	16	SYN		<CTRL>V
23	17	ETB		<CTRL>W
24	18	CAN	Cancel line on terminal	<CTRL>X
25	19	EM		<CTRL>Y
26	1A	SUB		<CTRL>Z

DECIMAL	HEX	CHARACTER	SPECIAL USE IN ULTIMATE	TERMINAL KEY
27	1B	ESC		
28	1C	FS		
29	1D	GS		
30	1E	RS		
31	1F	US		
32	20	SPACE		
33	21	!		
34	22	"		
35	23	#		
36	24	$		
37	25	%		
38	26	&		
39	27	'		
40	28	(		
41	29	)		
42	2A	*		
43	2B	+		
44	2C	'		
45	2D	-		
46	2E	.		
47	2F	/		
48	30	0		
49	31	1		
50	32	2		
51	33	3		
52	34	4		
53	35	5		
54	36	6		
55	37	7		
56	38	8		
57	39	9		
58	3A	:		
59	3B	;		
60	3C	<		
61	3D	=		
62	3E	>		
63	3F	?		
64	40	@		
65	41	A		
66	42	B		
67	43	C		
68	44	D		
69	45	E		
70	46	F		
71	47	G		
72	48	H		
73	49	I		
74	4A	J		
75	4B	K		
76	4C	L		
77	4D	M		
78	4E	N		
79	4F	O		

DECIMAL	HEX	CHARACTER	SPECIAL USE IN ULTIMATE	TERMINAL KEY
80	50	P		
81	51	Q		
82	52	R		
83	53	S		
84	54	T		
85	55	U		
86	56	V		
87	57	W		
88	58	X		
89	59	Y		
90	5A	Z		
91	5B	[		
92	5C	\		
93	5D	]		
94	5E	^		
95	5F			
96	60	'		
97	61	a		
98	62	b		
99	63	c		
100	64	d		
101	65	e		
102	66	f		
103	67	g		
104	68	h		
105	69	i		
106	6A	j		
107	6B	k		
108	6C	l		
109	6D	m		
110	6E	n		
111	6F	o		
112	70	p		
113	71	q		
114	72	r		
115	73	s		
116	74	t		
117	75	u		
118	76	v		
119	77	w		
120	78	x		
121	79	y		
122	7A	z		
123	7B	{		
124	7C	:		
125	7D	}		
126	7E	~		
127	7F	DEL		
128	80			
129	81			

DECIMAL	HEX	CHARACTER	SPECIAL USE IN ULTIMATE	TERMINAL KEY
130	82			
131	83			
132	84			
133	85			
134	86			
135	87			
136	88			
137	89			
138	8A			
139	8B			
140	8C			
141	8D			
142	8E			
143	8F			
144	90			
145	91			
146	92			
147	93			
148	94			
149	95			
150	96			
151	97			
152	98			
153	99			
154	9A			
155	9B			
156	9C			
157	9D			
158	9E			
159	9F			
160	A0			
161	A1			
162	A2			
163	A3			
164	A4			
165	A5			
166	A6			
167	A7			
168	A8			
169	A9			
170	AA			
171	AB			
172	AC			
173	AD			
174	AE			
175	AF			
176	B0			
177	B1			
178	B2			
179	B3			

DECIMAL	HEX	CHARACTER	SPECIAL USE IN ULTIMATE	TERMINAL KEY
180	B4			
181	B5			
182	B6			
183	B7			
184	B8			
185	B9			
186	BA			
187	BB			
188	BC			
189	BD			
190	BE			
191	BF			
192	C0			
193	C1			
194	C2			
195	C3			
196	C4			
197	C5			
198	C6			
199	C7			
200	C8			
201	C9			
202	CA			
203	CB			
204	CC			
205	CD			
206	CE			
207	CF			
208	D0			
209	D1			
210	D2			
211	D3			
212	D4			
213	D5			
214	D6			
215	D7			
216	D8			
217	D9			
218	DA			
219	DB			
220	DC			
221	DD			
222	DE			
223	DF			
224	E0			
225	E1			
226	E2			
227	E3			
228	E4			
229	E5			

DECIMAL	HEX	CHARACTER	SPECIAL USE IN ULTIMATE	TERMINAL KEY
230	E6			
231	E7			
232	E8			
233	E9			
234	EA			
235	EB			
236	EC			
237	ED			
238	EE			
239	EF			
240	F0			
241	F1			
242	F2			
243	F3			
244	F4			
245	F5			
246	F6			
247	F7			
248	F8			
249	F9			
250	FA			
251	FB	SB	Start buffer	<CTRL>[
252	FC	SVM	Subvalue Mark	<CTRL>\
253	FD	VM	Value Mark	<CTRL>]
254	FE	AM	Attribute Mark	<CTRL>^
255	FF	SM	Segment Mark	<CTRL>_

APPENDIX C

ASYNCHRONOUS (ASYNC) AND BINARY SYNCHRONOUS (BISYNC) COMMUNICATIONS MESSAGES

(121) UNEXPECTED INTERRUPT FROM DEVICE

Some device other than the bisync controller or the internal timer interrupted the process, so the message was aborted. Suspected hardware failure.

(122) TRANSMISSION ABORTED--'RVI' RECEIVED

An 'RVI' sequence was received from the other computer requesting this station to abort its transmission and get ready to receive a priority message. The process will turn around and go into the receive mode unless the (T) option was used with the TRANSMIT verb.

(123) TRANSMISSION ABORTED--'EOT' RECEIVED

An 'EOT' was received from the other station which signifies that the other station has discovered some unrecoverable error and has aborted the transmission.

(124) TRANSMISSION ABORTED--DISCONNECT SEQUENCE RECEIVED

A "DLE-EOT" sequence was received from the other station which signifies that the other station has discovered some unrecoverable error and has aborted the transmission and disconnected the communications line.

(125) TRANSMISSION ABORTED--'ENQ' COUNT EXCEEDED

Three tries were made to get a positive acknowledgment from the other computer that the last transmitted block of data was correctly received, and the correct response was not received.

(126) TRANSMISSION ABORTED--'NAK' COUNT EXCEEDED

The same block of data was retransmitted three times without being correctly received by the other station.

(127) TRANSMISSION ABORTED--COMM. LINE WAS DISCONNECTED

The telephone line connection (communications line) was broken during the transmission.

APPENDIX C

ASYNCHRONOUS (ASYNC) AND BINARY SYNCHRONOUS (BISYNC) COMMUNICATIONS MESSAGES

(128) ---WAITING FOR COMMUNICATIONS LINE TO BE CONNECTED---

The TRANSMIT or RECEIVE verb has been executed and the line connection to the other station has not yet been made.

(129) --------COMMUNICATIONS LINE CONNECTED--------

The TRANSMIT or RECEIVE verb has been executed and the line connection to the other station has been established.

(130) OTHER STATION WON LINE BID

The other station also bid for the line to transmit a message and, since this station was designated as a secondary station in the TRANSMIT verb's option list, this station stops trying to transmit its data and goes into the receive mode (if the 'T' option was not used with the TRANSMIT verb).

(131) LINE BID NOT ACCEPTED BY OTHER STATION

Fifteen attempts as one second intervals (three seconds if this station was selected as a secondary station by a TRANSMIT verb option) were made to get the other station to accept the message to be transmitted, without success.

(132) MESSAGE CORRECTLY RECEIVED

There was a normal completion on the message just received.

(133) TRANSMISSION ABORTED BEFORE MESSAGE TOTALLY RECEIVED

Some error condition arose which prevented the complete transmission of the message. Look at the previous system message to determine the reason for the abort.

(134) NO BISYNC COMMUNICATIONS PAC FOUND

The transmit or receive process could find no binary synchronous communications controller attached to the 'Ultimate' machine.

APPENDIX C

ASYNCHRONOUS (ASYNC) AND BINARY SYNCHRONOUS (BISYNC) COMMUNICATIONS MESSAGES

(135) TRANSMISSION ABORTED-- LINE CONTROL CHARACTER IN DATA

A forbidden line control character was encountered in the item being transmitted. The forbidden characters are (in hexadecimal 'X'): X'03' (ETX), X'05' (ENQ), X'16' (SYNC), X'17' (ETB), and X'1F' (ITB).

(136) DATA HAS BEEN LOST BECAUSE OF TRUNCATION

One or more of the attributes of the item being transmitted was truncated to the record size.

(137) ENTERING RECEIVE MODE

Data has been transmitted and the process has entered the receive mode.

(138) BSC CONTROLLER ALREADY ATTACHED TO LINE

An attempt was made to attach to a bisync communications controller that is already attached to another line.

(139) LINE ALREADY ATTACHED TO BSC CONTROLLER

Only one bisync communications controller may be attached to a line. An attempt was made to atttach more than one controller to the process.

(140) BSC CONTROLLER ATTACHED

The B-ATT verb successfully attached a bisync communications controller to the calling process.

(141) ATTACH BSC CONTROLLER

A communications verb was executed with no bisync communications controller previously attached.

(316) NUMERIC PARAMETER MISSING

A bisync controller number must be specified with the "U" option in the B-DET verb.

2780/3780 PROTOCOL (CONTROL CODES)

The following information is written from the perspective of what the Ultimate BSC software generates in order to interface with other hardware using 2780 and 3780 protocol (to be distinguished from the actual 2780/3780 protocol itself).

BSC CONTROL CODES

TERM	HEX	ALSO CALLED	FUNCTION
ACK0	1070		ALTERNATING AFFIRMATIVE
ACK1	1061		ACKNOWLEDGEMENTS
DLE	10		DATA-LINK-ESCAPE CHARACTER
EM	19		END OF MEDIA (SHORT RECORD)
ENQ	2D		ENQUIRY
EOT	37		END OF TRANSMISSION
ETB	26	EOB	END OF TRANSMISSION BLOCK
ETX	03		END OF TEXT
IGS	1D		3780 PROTOCOL SPACE COMPRESSION CHARACTER
IRS	1E		3780 PROTOCOL RECORD SEPARATOR (REPLACES AM)
ITB	1F	IUS	INTERMEDIATE BLOCK CHECK
NAK	3D		NEGATIVE ACKNOWLEDGEMENT
PAD	FF		MSG TRAILING PAD CHARACTER
RVI	7C		REVERSE INTERRUPT
SOH	01		2780/3780 TERMINALS ACCEPT AS EQUIVALENT OF STX
STX	02		START OF TEXT TRANSMISSION
SYNC	32		CHARACTER-PHASE SYNCHRONISM CHARACTER
WACK	6B		WAIT BEFORE TRANSMIT, POSITIVE ACKNOWLEDGEMENT

BSC CONTROL CODES EXPLANATION

ACK0/ACK1

These replies, in proper alternating sequence, indicate that the previous block was accepted without error and the receiver is ready to accept the next block of the transmission.

The normal sequence is an "ACK0" sent in response to an "ENQ" (bid for the line), which would be followed by an "ACK1" in response to the first data block received, continued by alternating "ACK0/ACK1"s.

DLE

The "DLE" - data link escape character is used in conjunction with an "EOT" being sent to signal a transmission abort condition. This will occur with the following errors: unknown device interrupt, an "EOT" sent from the receiver, control character(s) detected in transmission data, or if the "ENQ" or "NAK" count exceeds 15.

EM

The "END-OF-MEDIA" character is used to flag record lengths less than 80-character for use with the 2780 terminal. The "EM" is primary designed for use with the card punch to avoid the necessity of spacing through the rest of the card.

ENQ

the "ENQ" character is used by the transmitter to bid for the line. It is also used to obtain a repeat transmission of the response to a message block if the original response is in question, or when an illegal character is detected in the data to be transmitted (abort condition).

EOT

The transmitter will send an "EOT" to terminate the end of normal transmission or as a response to an "RVI". The receiver will send an "EOT" to signal end-of-transmission due to "NAK" count exceeded (15 in a row); a 20 second lapse in time without receiving something; or if it receives an unknown device interrupt.

ETB

An "ETB" signals the end of a transmission block. It is appended to the end of the block and is included in the cyclical redundancy check.

ETX

an "ETX" signals the end of text to be transmitted. It is also included in the cyclical redundancy check. In a normal transmission it would replace the "ETB" at the end of the last block sent. After proper acknowledgement, the transmitter would then send an "EOT".

IGS

The "IGS" is a 3780 protocol space compression character. It is normally followed by a count of the spaces omitted. This count is OR'ed by a (hex) X'40'. Each "IGS" may account for a maximum of 63 spaces. Both the "IGS" and the count character are included in the cyclical redundancy check.

IRS

The "IRS" is used in 3780 protocol as a record separator and is included in the cyclical redundancy check.

ITB

The "ITB" is used in 2780 protocol as a record separator and is included in the cyclical redundancy check.

NAK

The "NAK" is sent by the receiver to indicate a bad data transmission and request re-transmission of the previous data.

PAD

A (hex) X'FF' is used as a pad character format to reduce the probability of a transmission line error. all transmissions end with three PAD characters. The pad is not included in the cyclical redundancy check.

RVI

The "RVI" is treated as a positive acknowledgement (in place of ACK0/ACK1) and a request to end transmission. (Note that the "R" option will ignore the request to end transmission).

SOH

The "SOH" is a 'start-of-heading' character and is treated by 2780/3780 terminals as the equivalent of an "STX".

STX

The "STX" signals the beginning of a data block, and is not included in the cyclical redundancy check, except with 2780 protocol where each record has its own CRC and is followed by another "STX" at the beginning of the next record. These subsequent "STX"s within the data block are included in the CRC.

SYNC

The "SYNC" character is used to establish and maintain synchronization and as a time fill in the absence of any data or other control characters. The Ultimate will send three "SYNC"s at the beginning of every transmission.

WACK

The "WACK" is a positive acknowledgement of a transmission and a request to wait before further transmission. The Ultimate does not send "WACK"s. If one is received, a maximum of 15 "ENQ"s will be sent.

APPENDIX C (continued)

EXAMPLES OF DATA BLOCK TRANSMISSION

```
ITEM:   WHATEVER

ATTRIBUTE   1. A
ATTRIBUTE   2. B
```

<u>2780 PROTOCOL MODE:</u>

```
S S S
Y Y Y S     E I C C S     E E C C P P P
N N N T  A  M T R R T  B  M T R R A A A
C C C X       B C C X       X C C D D D
        |______|   |__________|
           |            |
   CYCLICAL REDUNDANCY CHECKED
```

<u>3780 PROTOCOL MODE:</u>

```
S S S
Y Y Y S     I     I E C C P P P
N N N T  A  R  B  R T R R A A A
C C C X     S     S X C C D D D
        |_____________|
              |
 CYCLICAL REDUNDANCY CHECKED
```

APPENDIX C (continued)

TRANSMIT DATA PARAMETERS

```
2780 MODE - DEFAULT

      7 RECORDS PER BLOCK OF DATA
    400 BUFFER SIZE
     80 CHARACTERS PER RECORD (WILL TRUNCATE)
```

```
ULTIMATE MODE (U OPTION, WITH 2780 PROTOCOL AS THE DEFAULT)

      7 RECORDS PER BLOCK OF DATA
    503 BUFFER SIZE
    500 CHARACTERS PER RECORD
```

```
3780 MODE (A OPTION)

    100 RECORDS PER BLOCK OF DATA
    500 BUFFER SIZE
     80 CHARACTERS PER RECORD (WILL TRUNCATE)
```

```
ULTIMATE/3780 MODE (A,U OPTIONS)

    100 RECORDS PER BLOCK OF DATA
    500 BUFFER SIZE
    500 CHARACTERS PER RECORD (WILL TRUNCATE)
```

```
THE M OPTION FURTHER AFFECTS THE CHARACTERS PER RECORD.

      M - 397 CHAR/RECORD
    M,U - 500 CHAR/RECORD
    M,A - 497 CHAR/RECORD
```

APPENDIX C (continued)

ASCII TO EBCDIC CONVERSION TABLE

In the transmit mode, the ASCII data in the Ultimate file is translated to EBCDIC using the following translation table:

ASCII	EBCDIC	ASCII	EBCDIC	ASCII	EBCDIC	ASCII	EBCDIC
00	00	20	40	40	7C	60	79
01	01	21	5A	41	C1	61	81
02	02	22	7F	42	C2	62	82
03	ABORT	23	7B	43	C3	63	83
04	ABORT	24	5B	44	C4	64	84
05	ABORT	25	6C	45	C5	65	85
06	2E	26	50	46	C6	66	86
07	2F	27	7D	47	C7	67	87
08	16	28	4D	48	C8	68	88
09	05	29	5D	49	C9	69	89
0A	25	2A	5C	4A	D1	6A	91
0B	0B	2B	4E	4B	D2	6B	92
0C	0C	2C	6B	4C	D3	6C	93
0D	0D	2D	60	4D	D4	6D	94
0E	0E	2E	4B	4E	D5	6E	95
0F	0F	2F	61	4F	D6	6F	96
10	10	30	70	50	D7	70	97
11	11	31	71	51	D8	71	98
12	12	32	72	52	D9	72	99
13	13o	33	73	53	E2	73	A2
14	3C	34	74	54	E3	74	A3
15	3D	35	75	55	E4	75	A4
16	ABORT	36	76	56	E5	76	A5
17	ABORT	37	77	57	E6	77	A6
18	18	38	78	58	E7	78	A7
19	19	39	79	59	E8	79	A8
1A	3F	3A	7A	5A	E9	7A	A9
1B	27	3B	5E	5B	4F	7B	C0
1C	1C	3C	4C	5C	E0	7C	6A
1D	1D	3D	7E	5D	90	7D	D0
1E	1E	3E	6E	5E	5F	7E	A1
1F	ABORT	3F	6F	5F	6D	7F	07

APPENDIX C (continued)

EBCDIC TO ASCII CONVERSION TABLE

In the receive mode, the EBCDIC data in the received message is translated to ASCII using the following translation table:

EBCDIC	ASCII	EBCDIC	ASCII	EBCDIC	ASCII	EBCDIC	ASCII
00	00	40	20	80	5B	C0	7B
01	01	41	41	81	61	C1	41
02	02	42	42	82	62	C2	42
03	ETX	43	43	83	63	C3	43
04	04	44	44	84	64	C4	44
05	09	45	45	85	65	C5	4
06	06	46	46	86	66	C6	46
07	7F	47	47	87	67	C7	47
08	08	48	48	88	68	C8	48
09	09	49	49	89	69	C9	49
0A	0A	4A	5C	8A	5F	CA	5F
0B	0B	4B	2E	8B	5F	CB	5F
0C	0C	4C	3C	8C	5F	CC	5F
0D	0D	4D	28	8D	5F	CD	5F
0E	0E	4E	2B	8E	5F	CE	5F
0F	0F	4F	5B	8F	5F	CF	5F
10	10	50	26	90	5D	D0	7D
11	11	51	51	91	6A	D1	4A
12	12	52	52	92	6B	D2	4B
13	13	53	53	93	6C	D3	4C
14	14	54	54	94	6D	D4	4D
15	15	55	55	95	6E	D5	4E
16	08	56	56	96	6F	D6	4F
17	17	57	57	97	70	D7	50
18	18	58	58	98	71	D8	51
19	19	59	59	99	71	D9	52
1A	1A	5A	21	9A	5F	DA	5F
1B	1B	5B	24	9B	5F	DB	5F
1C	1C	5C	2A	9C	5F	DC	5F
1D	1D	5D	29	9D	5F	DD	5F
1E	1E	5E	3B	9E	5F	DE	5F
1F	ITB	5F	5E	9F	5F	DF	5
20	20	60	2D	A0	5F	E0	5C
21	21	61	2F	A1	7E	E1	5F
22	22	62	62	A2	73	E2	53
23	23	63	63	A3	74	E3	54
24	24	64	64	A4	75	E4	55
25	0A	65	65	A5	76	E5	56
26	ETB	66	66	A6	77	E6	57
27	1B	67	67	A7	78	E7	58
28	28	68	68	A8	79	E8	59
29	29	69	69	A9	7A	E9	5A
2A	2A	6A	7C	AA	5F	EA	5F
2B	2B	6B	2C	AB	5F	EB	5F
2C	2C	6C	25	AC	5F	EC	5F
2D	ENQ	6D	5F	AD	5F	ED	5F
2E	06	6E	3E	AE	5F	EE	5F
2F	07	6F	3F	AF	5F	EF	5F

EBCDIC	ASCII	EBCDIC	ASCII	EBCDIC	ASCII	EBCDIC	ASCII
30	30	70	70	B0	5F	F0	30
31	31	71	71	B1	5F	F1	31
32	SYN	72	72	B2	5F	F2	32
33	33	73	73	B3	5F	F3	33
34	34	74	74	B4	5F	F4	34
35	35	75	75	B5	5F	F5	35
36	36	76	76	B6	5F	F6	36
37	O4	77	77	B7	5F	F7	37
38	38	78	78	B8	5F	F8	38
39	39	79	60	B9	5F	F9	39
3A	3A	7A	3A	BA	5F	FA	5F
3B	3B	7B	23	BB	5F	FB	5F
3C	14	7C	40	BC	5F	FC	5F
3D	15	7D	27	BD	5F	FD	5F
3E	3E	7E	3D	BE	5F	FE	5F
3F	1A	7F	22	BF	5F	FF	5F

INDEX

NOTES

ULT

THE ULTIMATE CORP.

717 RIDGEDALE AVENUE, EAST HANOVER, NEW JERSEY 07936

(201) 887-9222 TWX 710-996-5862 Telecopier (201) 887-6139

Document No. 6964