

**MCDONNELL
DOUGLAS**

**SERIES 6000
Programmer's
Reference Manual**

M6000

Programmer's Reference Manual

Release 1.1

July 1986

Document No. 86-1347

This software, data, documentation or firmware related thereto, and the information disclosed therein are confidential and proprietary to MCDONNELL DOUGLAS COMPUTER SYSTEMS COMPANY (hereinafter referred to as MDCSC). Neither the software, regardless of the form in which it exists, nor firmware, nor such data, nor information, may be used by or disclosed to others for any purpose except as specifically authorized in writing by MDCSC. Recipient by accepting this document or utilizing this software agrees that neither this document, the software nor the information disclosed therein nor any part thereof shall be reproduced or transferred to other documents nor used or disclosed to others for manufacturing or for any other purpose except as specifically authorized in writing by MDCSC.

©Copyright 1986 an unpublished work by MCDONNELL DOUGLAS COMPUTER SYSTEMS COMPANY, all rights reserved.

**MCDONNELL DOUGLAS
INFORMATION SYSTEMS LIMITED**

FOREWORD

=====

This manual provides information on the organisation of the M6000 system and the procedures for its efficient use. The enquiry language ENGLISH and applications languages DATA/BASIC and PROC are detailed in separate documentation, as is information relevant to specific peripheral devices and specific applications. The contents of this manual are:

Section 1, INTRODUCTION, outlines the nature of M6000 systems, the system software supplied, and the organisation of the database. It also describes the conventions employed in documentation.

Section 2, DATA STRUCTURES, gives information on the organisation of data into files and items and the management of virtual memory.

Section 3, FILES AND DICTIONARY FILES, describes how data is organised so as to be easily accessible to ENGLISH, PROC and DATA/BASIC.

Section 4, TERMINAL CONTROL LANGUAGE, introduces the types of command used from a terminal to control the system.

Section 5, LOGON/LOGOFF, details LOGON, LOGOFF and LOGTO procedures and gives information on system security, accounting and related topics.

Section 6, FILE MANAGEMENT PROCESSORS, describes database management processors including CREATE-FILE, CLEAR-FILE, DELETE-FILE and COPY.

Section 7, SPOOLER, details the features and use of the multi-device spooler.

Section 8, MAGNETIC TAPE UNIT COMMANDS, describes the use of magnetic tape units, and relevant TCL commands.

Section 9, SPECIAL SYSTEM UTILITIES, includes utility commands and PROCs.

Section 10, SYSTEM MAINTENANCE PROCEDURES: SYSPROG ACCOUNT, describes special procedures designed to be carried out from the System Programmer account.

Section 11, TERMINAL INDEPENDENT PROCESS HANDLER (TIPH), describes the use of processes not executed from a terminal.

Appendix A lists the ASCII Character Set.

Appendix B lists System Error Messages.

Appendix C shows the STAT-FILE DICTIONARY.

Appendix D describes the creation of the PH-HISTORY file.

Appendix E describes safeguarding save tapes.

Appendix H describes terminal executives.
Appendix I describes TCL PLUS.

An index is included at the end of the manual.

The remaining pages of this foreword summarise the changes between the previous and the current software releases.

Note:

Users of Model 6740, 6745 and 6748 terminals should retain at least the relevant parts of the previous edition of this manual (or equivalent documentation) as detailed information on these models has been omitted from this edition.

CONTENTS

	Page
1. INTRODUCTION	1-1
M6000 COMPUTER SYSTEM	1-1
THE FLEXIBLE FAMILY OF COMPUTER SYSTEMS	1-2
SYSTEM SOFTWARE	1-4
ENGLISH LANGUAGE	1-4
DATA/BASIC AND SCREENPRO	1-4
PROC	1-5
TERMINAL INDEPENDENT PROCESS HANDLER	1-5
EDITOR	1-5
RUNOFF	1-5
HOW TO USE THE SYSTEM MANUALS	1-6
SYSTEM OVERVIEW	1-8
FILE ORGANIZATION	1-10
SYSTEM DICTIONARY (SYSTEM)	1-10
USER MASTER DICTIONARIES	1-10
DICTIONARY LEVEL FILES	1-10
DATA FILES	1-10
INTERNAL DICTIONARY/FILE STRUCTURE	1-11
2. DATA STRUCTURES	2-1
VIRTUAL MEMORY STRUCTURE	2-2
ABS AREA	2-2
PRIMARY WORK SPACE	2-2
LOW DISK SPACE AREA AND FILE AREA	2-4
LOW DISK SPACE	2-4
HIGH DISK SPACE (FILE AREA)	2-4
AVAILABLE SPACE AREA	2-6
ADDITIONAL FILE SPACE	2-6
TEMPORARY WORK SPACE	2-8
FILE STRUCTURE	2-10
ITEM STRUCTURE (PHYSICAL)	2-12
ITEM STRUCTURE (LOGICAL)	2-14
ITEM STORAGE AND HASHING ALGORITHM	2-16
FILE UPDATE: STRUCTURE	2-18

	Page
3. FILES AND DICTIONARY FILES	3-1
FILE HIERARCHY	3-2
FILE DEFINITION ITEMS	3-4
FILE SYNONYM DEFINITION ITEMS	3-6
ATTRIBUTE DEFINITION ITEMS	3-8
SUMMARY OF DICTIONARY ITEMS	3-10
FILE AND ATTRIBUTE DEFINITION ITEMS	3-10
SYSTEM DICTIONARY (SYSTEM) ITEMS	3-10
MASTER DICTIONARY (M/DICT ITEMS)	3-10
INITIAL SYSTEM FILES/DICTIONARIES	3-12
SYSPROG ACCOUNT	3-12
SYSPROG-PL File	3-12
NEWAC File	3-12
TERMINAL-OBJECT File	3-12
SYSTEM-OBJECT File	3-12
ERRMSG File	3-12
ACCOUNTING HISTORY File	3-14
PROCLIB File	3-14
BLOCK-CONVERT File	3-14
POINTER-FILE File	3-14
DEVICES File	3-14
SYSTEM-LOG File	3-14
FEATURES File	3-14
PH-HISTORY File	3-14
4. TERMINAL CONTROL LANGUAGE	4-1
OVERVIEW	4-2
INPUT STATEMENTS	4-4
VERB DEFINITION ITEMS IN M/DICT	4-6
TCL-II INPUT STATEMENTS	4-8
TCL-I INPUT STATEMENTS	4-10
PROGRAM INTERRUPTION (DEBUG FACILITY)	4-12
5. LOGON/LOGOFF	5-1
LOGGING ON AND OFF THE SYSTEM	5-1
LOGGING ON TO THE SYSTEM	5-1
LOGGING OFF THE SYSTEM	5-2
SYSTEM MESSAGE FORMATS: LOGON MESSAGE	5-4
LOGON PROC AND GENERAL SYSTEM MESSAGES	5-6

	Page
THE LOGON PROC	5-6
GENERAL SYSTEM MESSAGE	5-6
COMPUTER USAGE: CHARGES AND CHARGE-TO VERBS	5-8
CHARGES VERB	5-8
CHARGE-TO VERB	5-8
LOGTO VERB	5-10
USER IDENTIFICATION ITEMS	5-12
USER IDENTIFICATION ITEMS: ATTRIBUTES 5 TO 9	5-14
WORK SPACE ASSIGNMENT	5-16
ALLOCATING PERMANENT BLOCKS OF ADDITIONAL WORK SPACE	5-18
THE WORKSPACE VERB	5-18
THE LOGON WORK SPACE TABLE (LWST)	5-18
PERMANENT BLOCKS	5-20
TEMPORARY BLOCKS	5-20
SECURITY	5-22
L/RET AND L/UPD	5-22
USER-ASSIGNED CODES	5-22
SECURITY CODE COMPARISON	5-23
ACCOUNTING HISTORY FILE: INTRODUCTION	5-24
ACTIVE USER ITEMS	5-24
ACCOUNTING HISTORY ITEMS	5-24
ACCOUNTING HISTORY FILE: SUMMARY AND EXAMPLES	5-26
ACCOUNTING HISTORY FILE: PERIODIC CLEARING	5-28
6. FILE MANAGEMENT PROCESSORS	6-1
OVERVIEW	6-1
CREATE-FILE PROCESSOR	6-1
CLEAR-FILE PROCESSOR	6-1
DELETE-FILE PROCESSOR	6-1
COPY PROCESSOR	6-1
EXAMPLES	6-2
ADDITIONAL FILE MANAGEMENT PROCEDURES	6-2
CREATING NEW FILES: CREATE-FILE PROCESSOR	6-4
SELECTING MODULO AND SEPARATION	6-6
CLEARING AND DELETING FILES: CLEAR-FILE & DELETE-FILE PROCESSORS	6-8
COPYING ITEMS: COPY PROCESSOR	6-10
ADDITIONAL COPY PROCESSOR FEATURES	6-14
7. SPOOLER	7-1
OVERVIEW	7-1
SUMMARY OF SPOOLER FUNCTIONS	7-2

PRINT JOBS	7-2
FORM QUEUE	7-2
LINE-AT-A-TIME MODE	7-2
DEVICE ATTACHMENT	7-2
HOLD FILES	7-4
SPOOLER VERBS	7-4
ASSIGNMENT OF PRINT JOBS AND OPTIONS: SP-ASSIGN VERB	7-6
H - CREATE HOLD FILE ON DISK	7-7
I - INSTANT OUTPUT	7-7
N - NO SPOOLING (OUTPUT DIRECTLY TO DEVICE)	7-7
S - SUPPRESS OUTPUT	7-7
GENERAL FORMAT AND EXAMPLES OF SP-ASSIGN VERB	7-8
DISPLAYING SP-ASSIGN PARAMETERS: SP-LOOK VERB	7-10
DISPLAYING STATUS OF SPOOLER: SP-STATUS VERB	7-12
SPOOLER OUTPUT DEVICES	7-14
LINE PRINTERS	7-14
TAPE UNITS	7-14
TERMINAL PORTS	7-14
CREATING A FORM QUEUE	7-16
CHANGING THE NAME OF A FORM QUEUE	7-18
CHANGING TYPE OF FORM USED ON PRINTER	7-20
CHANGING DEVICE ASSIGNMENTS	7-22
DELETING A FORM QUEUE	7-24
EJECTING PAGES FROM PRINTER	7-26
DISPLAYING PRINT JOB STATUS: SP-JOBS	7-28
MOVING A FORM QUEUE	7-30
MOVING PRINT JOBS	7-32
CHANGING PRINT JOB OPTIONS	7-34
CHANGING NUMBER OF PRINT JOB COPIES	7-36
DELETING PRINT JOBS FROM A QUEUE	7-38
STOPPING THE PRINTER BEFORE OR AFTER PRINTING A JOB	7-40
RESTARTING A PRINTER AFTER A FORCED WAIT	7-42
INTERRUPTING A PRINTING JOB	7-44
EXAMINING JOBS WAITING TO BE PRINTED	7-46
EDIT PRINT JOB COMMANDS: SUMMARY	7-48
EDIT PRINT JOB COMMANDS: LIST AND LOCATE	7-50
LOCATE A STRING	7-50
EDIT PRINT JOB COMMANDS: BACKWARD, FORWARD POSITIONING AND TOP	7-52
BACKWARD POSITIONING	7-52
FORWARD POSITIONING	7-52
TOP OF PRINT JOB	7-52
EDIT PRINT JOB COMMANDS: ALIGNMENT, PRINT AND EXIT	7-54
ALIGNMENT	7-54
PRINT LINES	7-54
EXIT	7-54
CHANGING PRIORITY OF PRINT JOBS	7-56
TCL CONTROL OF PORTS: SP-PORTON PROC AND PORTOUT VERB	7-58
PORTOUT	7-59
TCL CONTROL OF PORTS: SP-PORTON EXAMPLES	7-60
TCL CONTROL OF PORTS: SP-PORTOFF VERB	7-62
DRIVING A PRINTER ON THE P _{lan}	7-64
PRINTER CONTROL: P-ATT AND P-DET	7-66

CREATION OF SPOOL JOBS FROM TAPE: T-PRINT PROC AND SP-TAPEOUT VERB	7-68
SPOOLER ACTIVATION: :RESTART-SPOOLER and :SP-NEWTAB Verbs ..	7-70
8. MAGNETIC TAPE UNITS	8-1
MAGNETIC TAPE UNITS	8-1
ASSIGN AND CLEAR-ASSIGN VERBS;TAPE UNIT SEQUENCING	8-2
MAGNETIC TAPE UNIT SEQUENCING	8-3
T-ATT VERB	8-4
RECORD SIZE	8-4
CONTROL: T-DET, T-FWD, T-BCK, T-REW, T-WEOF, T-UNLOAD AND T-CHK VERBS	8-6
CARTRIDGE TAPE UNIT COMMANDS :T-ERASE AND T-RETENSION	8-8
T-ERASE	8-8
T-RETENSION	8-8
MAGNETIC TAPE I/O: T-READ VERB	8-10
MAGNETIC TAPE I/O: T-DUMP AND ST-DUMP VERBS	8-12
MAGNETIC TAPE I/O: T-LOAD VERB	8-14
MAGNETIC TAPE I/O: MT-LOAD AND MT-VERIFY VERBS	8-16
TAPE LABELS	8-18
READING TAPE LABELS: T-RDLBL VERB	8-20
MULTIPLE REEL TAPE OPERATIONS	8-22
T-COPY VERB	8-24
9. SPECIAL SYSTEM UTILITIES	9-1
OVERVIEW	9-1
ARITHMETIC VERBS	9-2
PORT CONFIGURATION MAINTENANCE (PCM)	9-4
PCM COMMAND	9-4
PCM : RE-DEFINITION	9-6
PCM : RE-DEFINE LINE CHARACTERISTICS	9-8
PCM : LINE CHARACTERISTICS	9-10
PCM : RE-DEFINE TERMINAL CHARACTERISTICS	9-12
PCM : REDEFINE USER CHARACTERISTICS	9-14
PCM : DEFINE LINE, TERMINAL AND USER CHARACTERISTICS	9-16
PCM : LISTING STANDARD DEFINITIONS	9-18
PCM : LIST-PORTS AND PORT-SETUP	9-20
MODIFYING 'TYPEAHEAD' CHARACTERISTIC DURING A SESSION	9-22
SETTING DEVICE CHARACTERISTICS: TERM VERB	9-24
TERM	9-24
SETTING DEVICE CHARACTERISTICS: SET-TERM VERB	9-26
SETTING DEVICE CHARACTERISTICS: TERMINAL PROC	9-28

	Page
DISABLING AND RE-ENABLING USE OF 'BREAK' FUNCTION	9-30
BLOCKING PRINTING: BLOCK-PRINT AND BLOCK-TERM VERBS	9-32
DISPLAYING ERRMSG-FORMAT ITEMS: PRINT-ERR VERB	9-34
DUMPING TO TERMINAL: I-DUMP AND S-DUMP VERBS	9-35
DUMP VERB	9-38
FID INFORMATION: ITEM AND GROUP VERBS	9-40
ITEM	9-40
GROUP	9-40
HASHING INFORMATION: ISTAT AND HASH-TEST VERBS	9-42
ISTAT	9-42
HASH-TEST	9-42
INFORMATION: MSG, CLEAR-MSG, WHO, AND WHAT VERBS	9-44
MESSAGE OR MSG	9-44
CLEAR-MSG	9-44
WHO	9-45
WHAT	9-45
INFORMATION: TIME, SET-TIME, SET-DATE AND DATE-FORMAT VERBS	9-46
TIME	9-46
SET-TIME	9-46
SET-DATE	9-46
DATE-FORMAT	9-46
INFORMATION: WHERE AND SLEEP VERBS	9-48
WHERE	9-48
SLEEP	9-48
INFORMATION: LIMITS, USER AND DISCIO VERBS	9-50
LIMITS	9-50
USER	9-50
DISCIO	9-50
SETTING PROCESS TIMESLICES: TIMESLICE VERB	9-52
SET PRIVILEGE LEVEL AT A PORT: SET-PRIV VERB	9-54
AVAILABLE SYSTEM SPACE: POVF VERB	9-56
POVF	9-56
GENERATING CHECKSUMS: CHECK-SUM COMMAND	9-58
GROUP FORMAT ERRORS: FIX-FILE-ERRORS VERB	9-60
SEARCHING A FILE FOR A CHARACTER STRING: SEARCH AND ESEARCH	
VERBS	9-62
COMPARISON OF FILES: COMP VERB	9-64
FILE MANAGEMENT PROCS: DELETE AND SET-FILE	9-66
DELETE PROC	9-66
SET-FILE PROC	9-66
PRINTRONIX PRINTER ADJUSTMENTS: PRINTRONIX UTILITY	9-68
MISCELLANEOUS PROCS/VERBS	9-70
CT PROC	9-70
CP PROC	9-70
CHOO-CHOO PROC	9-70
LISTACC PROC	9-70
LISTCONN PROC	9-71
LISTDICTS PROC	9-71
LISTFILES PROC	9-71
LISTPF PROC	9-71
LISTPROCS PROC	9-71
MISCELLANEOUS PROCS/VERBS	9-72

	Page
LISTU PROC	9-72
LISTVERBS PROC	9-72
SPACE PROC	9-72
COMPARE VERB	9-72
CHECK PROC	9-73
SOURCE COMPACTION: ECOPY, EEDIT (EED) AND EBASIC VERBS	9-74
10. SYSTEM MAINTENANCE PROCEDURES: SYSPROG ACCOUNT	10-1
SYSTEM PROGRAMMING (SYSPROG) ACCOUNT	10-1
MINIMIZING DATA LOSS DURING RESTORES: SET-WRITES VERB	10-2
USING SET-WRITES FORMULA #1	10-4
FORMULA #1: DETERMINING A SET-WRITES SETTING ('k')	10-4
USING SET-WRITES FORMULA #2	10-6
ACCOUNT-SAVE PROC	10-8
MULTIPLE ACCOUNT SAVE	10-10
ACCOUNT-RESTORE	10-12
ACCOUNT-RESTORE	10-12
MULTIPLE ACCOUNT RESTORE	10-14
SELECTIVE-RESTORES	10-16
SEL-RESTORE	10-16
TCL INITIATION OF FILE-RESTORE: :FILES AND :ABS/FILES VERBS ..	10-18
COMPLETE FILE-RESTORE	10-18
INITIALISING THE SYSTEM: SYSTEM-SETUP PROC	10-20
SYSTEM-SETUP	10-20
FORMATTED FILE-SAVES AND FILE-SAVE PROC	10-22
FILE-SAVE TAPES: FORMAT	10-24
TAPE SEGMENTS IN FORMATTED FILE-SAVE	10-26
OPERATION OF FORMATTED FILE-SAVE	10-28
PLANNED SHUTDOWN PROC	10-30
RESTARTING THE SYSTEM	10-31
COLDDUMP AND SAVE VERBS	10-32
GROUP LOCKS	10-33
TREATMENT OF GROUP FORMAT ERRORS	10-33
NON-MAILED ITEMS AND SKELETON FILES	10-33
USE OF OVERFLOW	10-33
BOOTDUMP VERB	10-34
CREATING AND ENABLING USER-ACCOUNTS	10-36
CREATE-ACCOUNT PROC	10-36
SETUP-ASSY PROC	10-36
DELETING USER-ACCOUNTS: DELETE-ACCOUNT PROC	10-38
SYSTEM GENERATION AND VERIFICATION	10-40
SYS-UPDATE PROC	10-40
UPDATE-ACCOUNT PROC	10-40
VERIFY-SYSTEM PROC	10-40
VERIFYING, AND MAKING CONTIGUOUS, AVAILABLE SYSTEM SPACE	10-42
DISK AND PLAN ERROR RECORDING	10-44
SYSTEM ERROR RECORDING	10-46

	Page
DISK MAPPING: CALCFID AND DSKAD VERBS	10-48
FILE STATISTICS REPORT: LIST-, ALL-, AND ACCOUNT-FILE-STATS	10-50
FILE STATISTICS REPORT: STAT-FILE DICTIONARY	10-52
DETERMINING MEMORY BUFFER STATUS: BUFFERS VERB	10-54
SYSPROG UTILITY VERBS: LOCK-FRAME AND UNLOCK-FRAME	10-56
LOCK-FRAME VERB	10-56
UNLOCK-FRAME VERB	10-56
GROUP LOCKS AND EXECUTION LOCKS	10-58
LIST-LOCKS	10-58
CLEAR-BASIC-LOCKS	10-58
:INIT-SYSTEM	10-58
ITEM LOCKING: OVERVIEW	10-60
ITEM LOCKING: LIST-ITEM-LOCKS VERB	10-62
ITEM LOCKING: CLEAR-ITEM-LOCKS VERB	10-64
ITEM LOCKING: SIZE-ITEM-LOCKS AND :IL-NEWTAB VERBS	10-66
VERIFICATION OF ASSEMBLY LANGUAGE ROUTINES FROM TAPE	10-68
11. TERMINAL INDEPENDENT PROCESS HANDLER (TIPH)	11-1
OVERVIEW	11-2
TIPH COMMANDS	11-3
PORT ASSIGNMENT: PH-ALLOCATE, PH-DELETE, PH-LINES	11-4
PH-ALLOCATE	11-4
PH-DELETE	11-4
PH-LINES	11-4
INITIATING A TIPH PROCESS: THE PH-START VERB	11-6
PH-START OPTIONS	11-8
HALTING AND REACTIVATING A TIPH PROCESS	11-10
DISPLAYING THE TIPH PROCESSES: THE PH-STATUS VERB	11-12
TIPH INITIALISATION MESSAGES	11-14
TIPH TERMINATION MESSAGES	11-16
APPENDIX A	A-1
ASCII CHARACTER SET	A-1
APPENDIX B	B-1
SYSTEM ERROR MESSAGES	B-1

	Page
APPENDIX C	C-1
STAT-FILE DICTIONARY	C-1
APPENDIX D	D-1
CREATING THE PH-HISTORY FILE	D-1
APPENDIX E	E-1
SAFEGUARDING SAVE TAPES	E-1
INDEX	INDEX-1

	Page
Special Edit Options	G-4
Abbreviations	G-5
DUMP-FILES AND LOAD-FILES - DUMP AND LOAD FILES FROM MAGNETIC TAPE	G-6
PD - PRINT DICTIONARY ITEMS	G-9
LIST4, SORT4 AND LISTF4 - FOUR COLUMN OUTPUT	G-9
PROC TRACE FACILITY	G-9
AUTOMATIC REGENERATION OF SPOOLER FORM-QUEUES	G-9
 APPENDIX H	 H-1
 TERMINAL EXECUTIVES	 H-1
TERMINAL EXECUTIVE FOR MODEL 12080	H-1
Straight-through Printing	H-3
Autoprint Printing	H-11
TERMINAL EXECUTIVE FOR MODEL 6749 COMMUNICATIONS TERMINAL	H-13
TERMINAL EXECUTIVE FOR MODEL 6762 REMOTE PRINTER CONTROLLER	H-13
TERMINAL EXECUTIVE FOR MODEL 9747	H-14
Print Screen	H-15
Straight-Through Printing	H-16
Autoprint Printing	H-16
VERIFYING T.E. CODINGS	H-17
 APPENDIX I	 I-1
 TCL PLUS (Version 8)	 I-1
INTRODUCTION	I-1
'ENTER' MODE	I-1
'RECALL' MODE	I-2
SAVE AND RECALL FEATURE	I-3
USE OF SELECT AND SSELECT COMMANDS	I-3
 INDEX	 INDEX-1

1. INTRODUCTION

=====

M6000 COMPUTER SYSTEM

The M6000 computer system is a generalised system for data base management. It provides multiple users with immediate access to information stored on-line. Users communicate via local or remote terminals with computer files that may be private, common, or security-controlled. Each user's vocabulary of commands can be individually tailored.

M6000 systems are built of field-proven computers and peripherals, using microprograms to provide performance and reliability. Microcode includes virtual memory management, multiuser operating system, and special data management instructions.

The system includes the powerful, yet simple to use, ENGLISH retrieval language, DATA/BASIC and PROC high level languages, the EDITOR processor, program development and file maintenance tools and a host of other user amenities. M6000 systems run in an on-line, multiuser environment efficiently managed by a largely microprogrammed virtual-memory operating system.

System software includes:

- . ENGLISH, DATA/BASIC and PROC languages
- . Selectable/automatic report formatting
- . Dynamic file/memory management
- . RUNOFF text processing
- . SCREENPRO language
- . EDITOR processor

Optional facilities include:

- . 'ALL' (Application Language Liberator) fourth generation language for application generation
- . Connection of PCs for access to the database and stand-alone use
- . Networking and data communications

The file structure provides:

- . Variable length files/records/fields
- . Multivalued (and subvalues) in a field
- . Efficient storage utilisation
- . Fast access to data items
- . Selectable degrees of data security
- . File size limited only by size of disk
- . Item size up to 32,267 bytes

THE FLEXIBLE FAMILY OF COMPUTER SYSTEMS

The expandable family of M6000/M9000 high-performance data base management processors ranges from an economical system for first-time users with limited data processing requirements and/or experience, to the high capacity systems used by some of the largest companies in the world.

Besides superb performance, the entire range offers unmatched growth advantages. As your company grows, you can add equipment to meet its increased data processing needs without the costly replacement and conversion costs usually associated with updating computer facilities. All M6000/M9000 systems are both hardware- and software-compatible.

A basic M6000 system has:

- . Central processing unit (CPU and cabinet)
- . Mass storage disk drive
- . Input/output Processor
- . Magnetic tape drive
- . Peripheral LAN controller (P-LAN)
- . Video display, printing or other terminals

All REALITY-family systems include the easy-to-use ENGLISH retrieval language, as well as the more advanced DATA/BASIC and PROC, and are fully compatible with other REALITY-family systems.

Many additional features are available as options, such as connection of PCs, the fourth-generation language ALL for application creation, and extensive networking capabilities.

- . Complete business computer capabilities
- . Microprogrammed virtual memory operating system
- . Up to 64 users and greater than 500 million characters of file storage
- . On-line file update/retrieval
- . ENGLISH retrieval language
- . Fast terminal response
- . Multiple printer spooling
- . Optional communications and networking capabilities
- . Optional fourth-generation language for application development
- . Facilities for integration of PCs into the system
- . Special data management processors
- . High-speed sort capability
- . Small computer price
- . Big computer performance
- . Top to bottom computer/peripheral compatibility within the REALITY operating system family.

Figure A: M6000 System Advantages

INTRODUCTION

SYSTEM SOFTWARE

Processors available on the system comprise the most extensive data base management software available on any minicomputer. Overviews of the standard processors and their typical uses follow.

ENGLISH LANGUAGE

ENGLISH is a generalized data retrieval/report generator language. A typical ENGLISH inquiry consists of a relatively free-form sentence containing verbs, file-names, data selection criteria, and control modifiers. An easy-to-use, dictionary-based language that uses simplified prose statements, ENGLISH permits you to produce original reports rapidly and efficiently.

ENGLISH applications are limitless because of the ease with which output can be accessed from user files. Since nonprogrammers can master the process quickly, ENGLISH is a valuable information management tool for many people in an organization, from sales personnel to top-level executives. Its major uses are report generation and enquiry/response applications. You may also use ENGLISH to produce output after file updates with DATA/BASIC or PROC software, as well as printing one-of-a-kind reports without writing a program.

DATA/BASIC AND SCREENPRO

BASIC (Beginners All-purpose Symbolic Instruction Code) is a simple, yet versatile, programming language suitable for expressing solutions to a wide range of problems. DATA/BASIC, an extension of Dartmouth BASIC, is especially easy for the beginning programmer to learn.

DATA/BASIC is the primary method of updating user files on a M6000 system. Because of its flexibility, DATA/BASIC is used for a variety of business applications including accounts payable/receivable, general ledger, inventory control, payroll, sales forecasting/analysis, order processing, invoicing, claims processing, data entry, and other projects.

The SCREENPRO processor provides the means for the creation and processing of screens to display text, inputs, validations and updates.

Because SCREENPRO requires fewer program statements, it greatly simplifies program maintenance while increasing operator and programmer efficiency. Data throughput is accelerated. A screen can be designed, displayed, tested and changed without affecting the program.

With the addition of SCREENPRO, DATA/BASIC programs are even easier to write - and run faster - since screen handling and data validation can be removed from the program.

PROC

The PROC processor enables you to prestore a complex sequence of operations which can then be evoked by a single word command. Any sequence of operations that can be executed from the terminal can be prestored in a PROC. Although PROC is similar to the Job Control Language (JCL) used in larger computer systems, it is less cryptic and has far greater capabilities including interactive (optionally formatted) terminal prompting, input validation, printer formatting, and file input/output.

PROCs are typically used to create special user-defined functions by combining execution of DATA/BASIC programs, ENGLISH data retrieval operations, and PROC argument passing.

TERMINAL INDEPENDENT PROCESS HANDLER

The Terminal Independent Process Handler (TIPH) initiates a process on a port without an associated terminal, thus freeing the terminal for user interaction. Any terminal output (such as error messages, logon/off messages) will be placed in a spooler hold file. You may "stack" input in the command stream sent to the TIPH processor to enable execution of a program which requires operator input.

EDITOR

The EDITOR permits on-line interactive modification of any item in the data base. Primarily, the EDITOR is used to create and/or modify DATA/BASIC or PROC programs. The EDITOR may also be used to enter and update text for processing by RUNOFF.

RUNOFF

RUNOFF is a text processing facility offering many special features. Your RUNOFF text is entered and modified with the EDITOR. RUNOFF numbers pages automatically and can print text headings and footnotes.

Another RUNOFF feature is chapter and section numbering. New chapters and/or sections may be added to a document, and the subsequent updated publication, with changes and additions, will be completely renumbered automatically. You may instruct RUNOFF to assemble and print a table of contents and an index based on specified words and phrases. Any changes you make to the text will automatically be reflected in both the table of contents and index.

RUNOFF also performs tabulations, centering, selective left/right justifications, underlining, and boldface printing.

INTRODUCTION

HOW TO USE THE SYSTEM MANUALS

| This manual is written in modular format with each pair of facing pages |
presenting a single topic.

The approach taken in this and most other system manuals differs substantially from the typical reference manual format. Here, each pair of pages discusses an individual topic. Generally the left-hand page is devoted to text, while the right-hand page presents figures referred to by the text.

The advantage of this format will become readily apparent as you begin to use this manual. First of all, the figures referred to in the text are always conveniently in front of you at the point where the reference is made. Secondly, there is a psychological advantage knowing that when a topic is completed and the page is turned, you are done with one idea and are ready for another.

In presenting general command formats and examples throughout this and other M6000 manuals, certain conventions apply. Conventions used in presenting general command formats are listed in Figure A, while conventions used in the examples are listed in Figure B.

<u>Convention</u>	<u>Meaning</u>
UPPER CASE	Characters printed in upper case are required and must appear exactly as shown.
lower case	Characters or words printed in lower case are parameters to be supplied by you (i.e., file-name, item-id, data etc.).
{ }	Braces surrounding word and/or parameter indicates that the word and/or parameter is optional and may be included or omitted at your option.
{ }...	Three dots following the terminating brace indicate that the enclosed word or parameter may be omitted or repeated an arbitrary number of times.
item-list*	An asterisk following an item-list indicates that the list of item-ids may be omitted if supplied by a previous SELECT, SSELECT, GET-LIST, FORM-LIST, or other SELECT-list generating statement.

Figure A: Conventions Used in General Command Formats

<u>Convention</u>	<u>Meaning</u>
<cr>	This symbol represents a carriage return.
<lf>	This symbol represents a line feed.
<c> or <ctrl>	This symbol specifies that the following character is a control character generated by pressing the <CTRL> key while typing the character.

Figure B: Conventions Used in Examples

INTRODUCTION

SYSTEM OVERVIEW

M6000 is a complete system of computer hardware, software, and firmware, specifically oriented to implement cost-effective data base management. Data base management systems implemented in M6000 afford two major benefits: (1) accurate and timely information to significantly improve the decision-making process, and (2) substantially reduced clerical and administrative effort associated with the collection, storage, and dissemination of information.

M6000 is a complete computer system combining both proprietary hardware and software to create an effective tool for on-line data base management. Two major components of the system have been implemented directly in CPU firmware:

- . Virtual memory operating system
- . Software level architecture

Virtual memory operating systems, long used in larger computer systems, have been impractical for minicomputers due to the large amount of main memory needed for the operating system itself. In M6000, the virtual memory operating system has been directly implemented in firmware (i.e., high-speed read-only memory), which executes many times faster than a comparable system normally implemented in software. With the virtual memory manager directly implemented in read-only memory, only a small amount of main CPU memory is needed to run the system.

Slightly over 6,000 bytes of main memory must be allocated for the operating system monitor. Everything else (system software and data) is transferred automatically into main memory from virtual memory (i.e., disc) by the virtual memory operating system in a demand-paged environment. The M6000 computer system is organized into 512-byte pages (frames) stored in the disc. As a frame is needed for processing, the operating system automatically transfers that frame from the disk unit (virtual memory) to main memory; this concept is illustrated in Figure A. The virtual memory feature of M6000 allows you to have access to a programming area not constrained by a main memory, but rather can be as large as the entire available disk storage on the system.

The second feature implemented directly in firmware is the software level architecture of the machine itself. M6000 has a machine architecture expressly designed and optimized for data base management. The assembly language architecture has very powerful instructions expressly designed for character moves, searches, compares, and all supporting operations to manage variable length fields and records. In addition, this software architecture has a very large field-proven software base written for data base management.

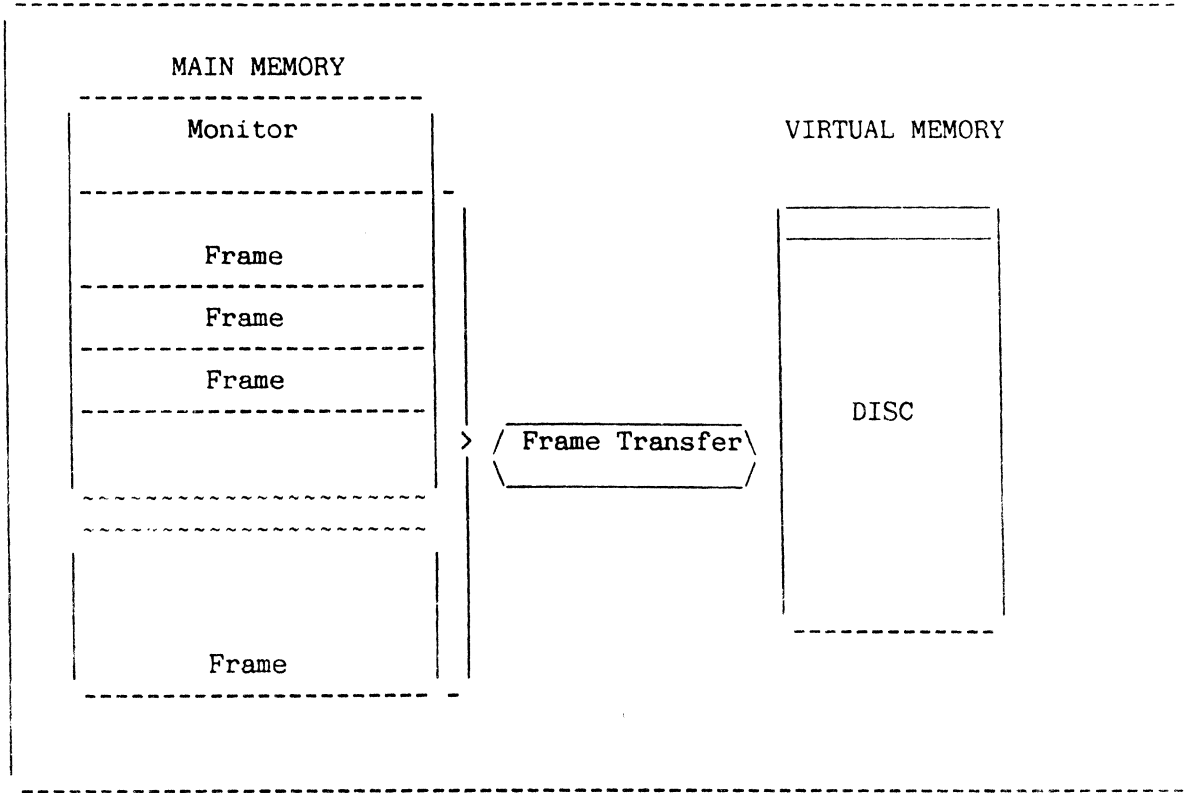


Figure A: Demand-Paged Environment

INTRODUCTION

FILE ORGANIZATION

Files are organized in a hierarchical structure, with files at one level pointing to multiple files at a lower level. Four distinct file levels exist: System Dictionary (SYSTEM), User Master Dictionary (M/DICT), Dictionary (DICT) Level File and Data File.

The hierarchical file structure is illustrated in Figure A. Each level is described briefly below.

SYSTEM DICTIONARY (SYSTEM)

The highest level file on each system is the System Dictionary. This file is used for system control; it contains pointers to each user's Master Dictionary and to various system support files.

USER MASTER DICTIONARIES

Master Dictionaries (M/DICTs) comprise the next level. Each user's account has an M/DICT associated with it; the M/DICT defines all user vocabulary, all accessible file-names and attributes describing the structure of the information in lower level dictionaries. File-name pointers can reference any file or dictionary in the system, subject to security restrictions.

DICTIONARY LEVEL FILES

Dictionary Level Files describe the structure of data in associated data files.

DATA FILES

Data Files contain the actual data stored in variable-length format. In addition to the normal record/field data structure, a field (called an attribute) can contain multiple values, and a value, in turn, can consist of multiple subvalues. Thus, data may be stored in a three-dimensional variable-length format.

INTERNAL DICTIONARY/FILE STRUCTURE

The term "file" refers to a mechanism for maintaining a set of like items logically together. The data in a file is normally accessed via the dictionary associated with it. Since the dictionary itself is also a file, it contains items just as a data file does. The items in a dictionary serve to define lower level dictionaries or data files. Several types of items are used in dictionaries:

- . File definition items: define files
- . File synonym definition items: point to files
- . Attribute definition items: describe fields

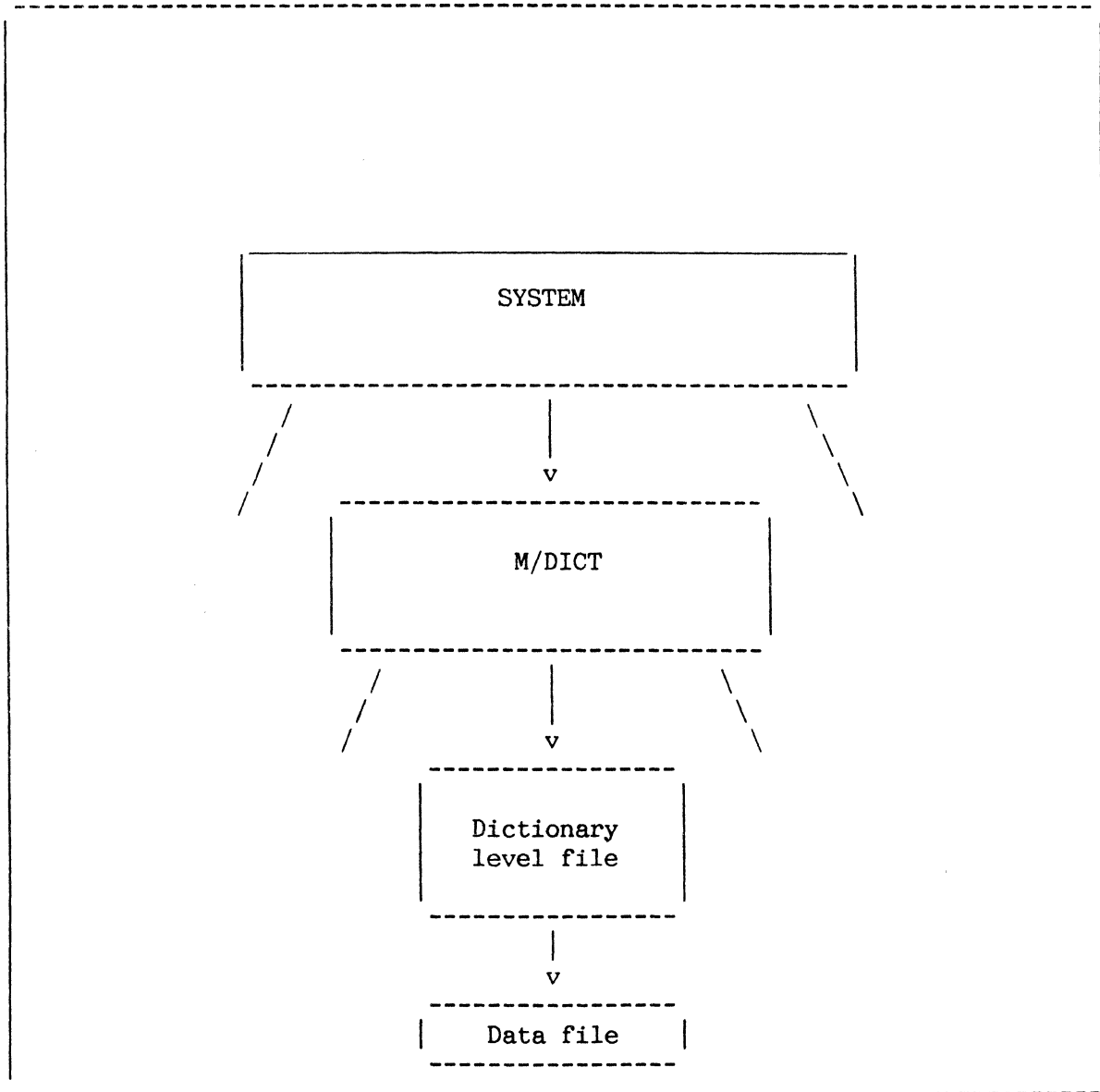


Figure A: Hierarchical File Structure

2. DATA STRUCTURES

=====

This section describes the use of virtual memory and associated topics

The topics covered in this section are as follows:

- allocations of virtual space.
- management of available space.
- file structure.
- physical and logical item structure.
- item storage and the hashing algorithm.
- file updates.

DATA STRUCTURES

VIRTUAL MEMORY STRUCTURE

The system's virtual memory structure allows disk space to be immediately available as main memory. The entire set of data including executable programs, process work space, and all system and user files resides in the virtual memory. The monitor is resident in main memory.

Prior to understanding the data structures used it is useful to have a general understanding of the virtual memory structure. The following is a brief overview of that structure.

The virtual memory consists of a set of 512-byte frames, each addressable by a positive integer called the Frame-ID (FID). The virtual memory contains absolute, work, and file areas, as illustrated in Figure A. Frames move between disk and main memory as required by the system.

ABS AREA

Starting with the first frame (FID=1) and continuing upward is the Absolute Area (called ABS). In this area 1535 frames are reserved. Frames 1 through 399 and 1000-1535 of the ABS Area are reserved for current and future system software. The remaining frames of this area (400 through 999) are reserved for special purpose and user-written assembly language programs.

PRIMARY WORK SPACE

You interact with the system via a terminal attached to a port. The ongoing dialogue with any port is called a process. (Additionally, certain processes not actually connected with a port, known as TIPH processes, may be defined.) Uniquely associated with each process is a Primary Control Block (PCB) which is a one-frame block that defines the state of the process at any instant. Each PCB is followed by a 63-frame work space used by the process. Thus, 64 frames are initially reserved for each defined process.

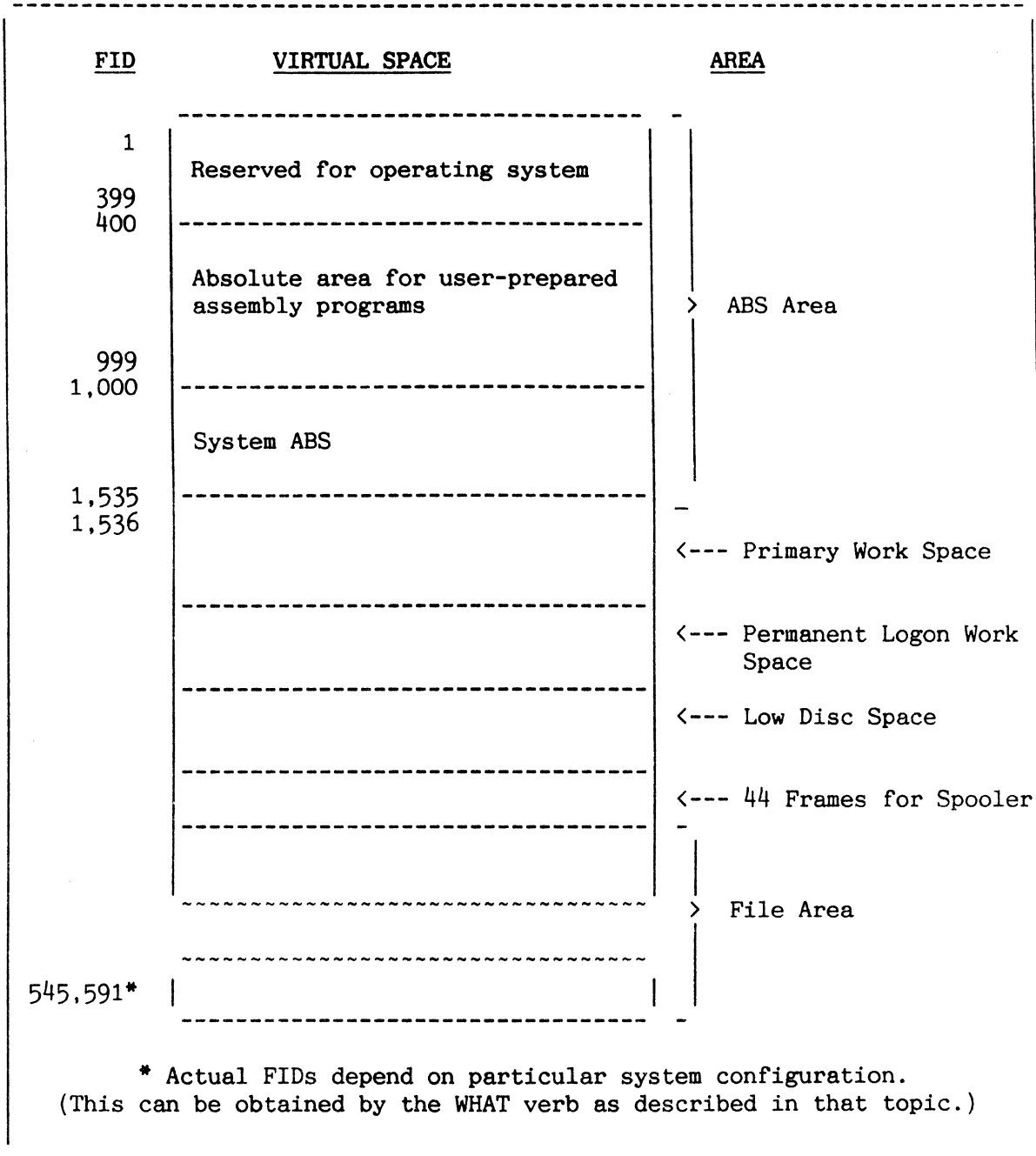


Figure A: Overview of Virtual Memory Structure

LOW DISK SPACE AREA AND FILE AREA

Beginning immediately after the primary work space is the low disk space area. This area is used primarily for additional process work space. After the low disk space, the remainder of the virtual memory (called the file area) is available for storage of data in files. If the main file area is full, the low disk space is used for data storage. Figure A illustrates these areas.

LOW DISK SPACE

During a file restore, a pool of frames (called low disk space area) is allocated for additional LOGON work space. Low disk space consists of 381 frames for each process (line) as specified by hardware, which corresponds to the maximum number of frames of additional workspace which a process may require (3 times the maximum additional workspace parameter of 127).

The spooler process is allocated a fixed number of frames (44) of additional workspace instead of the 381 available to be allocated for ordinary processes.

The LOGON processor attempts to obtain additional work space frames from low disk space; if no low disk space is available, the LOGON processor attempts to obtain additional work space from the high disk space. Additional work space assignment is detailed in Section 5.

HIGH DISK SPACE (FILE AREA)

Immediately following Low Disk Space is an area called High Disk Space. Most processors, such as SORT, spooler and file processors, attempt to get space from this area; if no high space is available, an attempt is made to obtain low space.

Frames which are not in use at a given time are maintained as a pool of available space. Management of available space is discussed in the next topic.

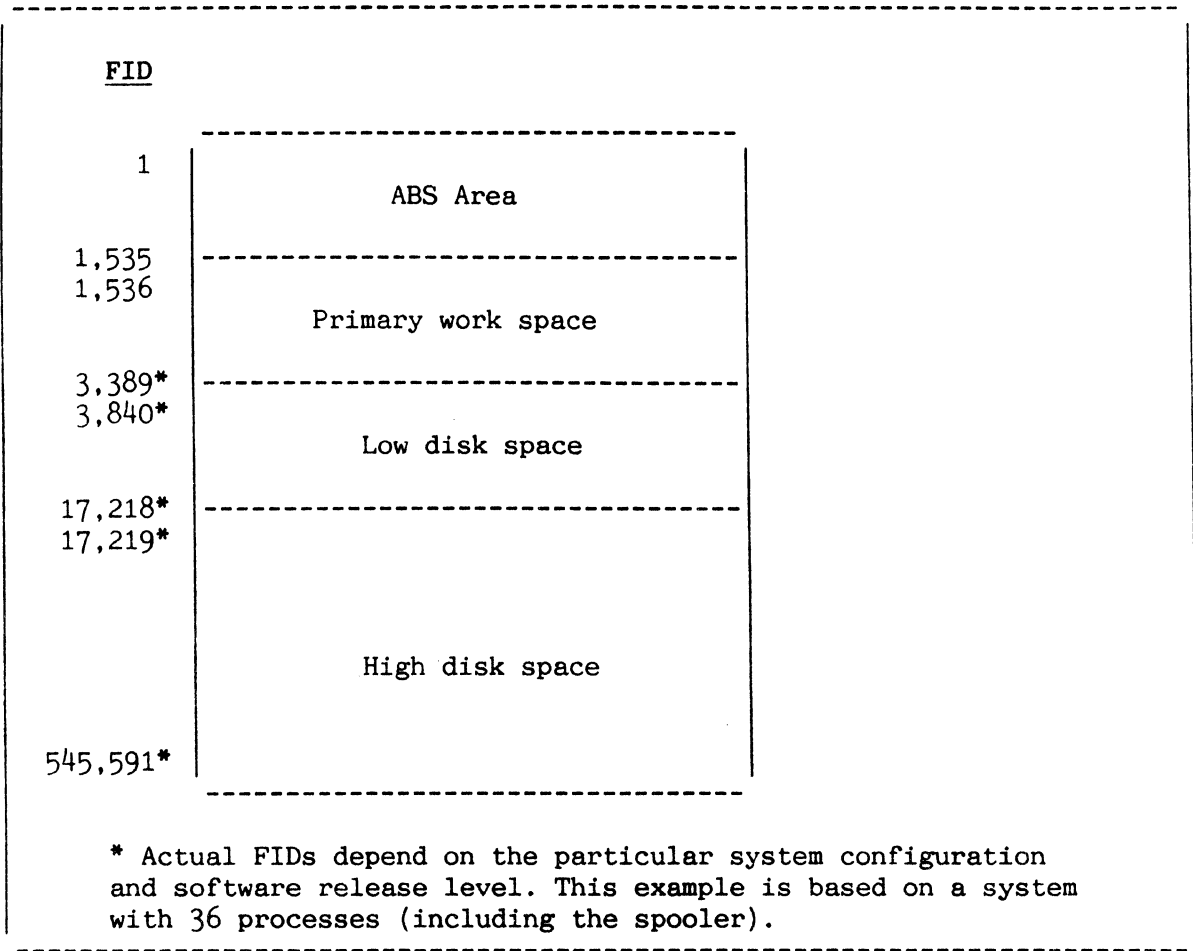


Figure A: Location of Low Disk Space and File Areas

DATA STRUCTURES

AVAILABLE SPACE AREA

| File area frames (from high and low disk space) which are not allocated |
to the files are maintained as a pool of available space.

Available space is used for two purposes:

1. for additional file space, and
2. for temporary work space used by such processors as the spooler, PROC, and ENGLISH.

Available space may be either in a "linked" form or in a "contiguous" form. Contiguous available space, as the name implies, consists of blocks of contiguous frames (defined by starting and ending numbers) that can be taken out of the pool either singly or as a block. Linked available space can only be taken a frame at a time. Conversely, space may be released by processors to the linked available pool a frame at a time or to the contiguous pool as a block.

Four tables of pointers define the available space. Two tables define contiguous blocks of space, one for file space and the other for temporary work space. The other two tables define linked chains, one for file space and the other for temporary space.

ADDITIONAL FILE SPACE

The table defining contiguous file space has up to 32 entries. When a block of contiguous file space is requested, this table is searched for a block of space equal to the amount requested, or the next largest block of space. The purpose is to allocate the smallest block that fits the size of the requested space.

When a linked frame is requested, the frame is taken from the linked chain of file space. If there is no linked chain, the frame is taken from a contiguous block of file space. When there is no contiguous file space, the linked chain of temporary space is searched. If still unsuccessful, the frame is taken from the contiguous block of temporary space.

File space (both linked and contiguous) is returned to the two file space tables. The system attempts to match the space up with an existing block in one of the tables. If the attempt fails, the space is returned to the linked chain of temporary space.

- THIS PAGE INTENTIONALLY LEFT BLANK

DATA STRUCTURES

TEMPORARY WORK SPACE

Requests for temporary space are made in the following order:

1. the temporary linked chain
2. the temporary block of contiguous space
3. the contiguous block of file space (if successful, this space is placed in the temporary block of contiguous space), and
4. the linked chain of file space.

The system attempts to give the space back to the temporary block of contiguous space and to the contiguous block of file space. If the attempt fails, the frame is added to the linked chain of temporary space.

At the conclusion of the file-restore process an initial condition may be said to exist: there will be one principal block of available high disk space, extending from the end of the current data space through the last available data frame. This is illustrated in Figure A; the shaded area represents the used portion of the file area. As the system obtains and releases available space, the available space may get fragmented; at any particular time there may be several blocks of contiguous available space and a chain of linked frames. (The low disk space area is not generally subject to fragmentation because its space is usually obtained and released in large blocks as processes log on and off). This is illustrated in Figure B; here the linked available chain starts at FID 33,562 and contains 857 frames. Linked available space may be reorganized into contiguous available space via the CLAIM verb (refer to Section 10). The POVVF verb may be used to display the available space (refer to Section 9).

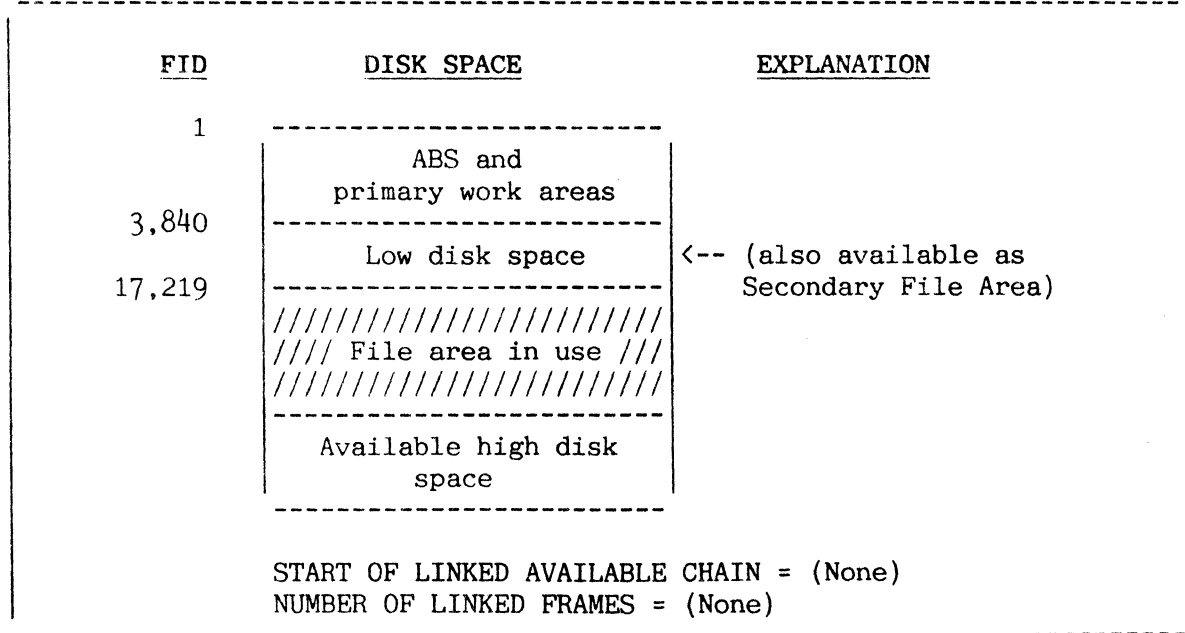


Figure A: Example of Contiguous Available Space After a File-Restore

<u>FID</u>	<u>DISK SPACE</u>	<u>EXPLANATION</u>
1	----- ABS and primary work areas	
3,840	----- Low disk space	<-- (also available as Secondary File Area)
17,219	----- //// File area in use //// -----	
	-----	<-- Start of available linked chain
	-----	<-- Contiguous available block
	//// File area in use //// -----	
	-----	<-- Available linked chain
	-----	<-- Contiguous available block
	-----	<-- Available linked chain
	//// File area in use //// -----	
	-----	<-- Available linked chain
	-----	<-- Contiguous available block

START OF LINKED AVAILABLE CHAIN = 33,562
NUMBER OF LINKED FRAMES = 857

Figure B: Example of Fragmented Available Space

FILE STRUCTURE

| A file is a related group of items which can be accessed for retrieval |
and update operations by referencing the item-id.

To understand the file structure, the following terms must first be defined:

- Bit - A digit in binary notation (i.e., either 1 or 0).
- Byte - A unit of information consisting of 8 bits. Each character stored represents one byte.
- Item - Synonymous to record. Usually made up of various attributes (fields) that pertain to or further describe an entity. For example, an inventory item with description and quantity fields or a charge account item containing balance, name and address fields. Maximum size of an item is 32,267 bytes.
- Item-id - A unique datum (key) within a file item by which all of the data in the item is identified or referenced, such as a part number or account number. Size limit is 50 characters.
- Attribute - A field within the item.
- Multivalues - One of several values in the same attribute (field).
- Subvalues - One of several values within a multivalue within an attribute.
- File - A set of like items brought together for a like purpose.
- Group - A set of linked frames (i.e, frame = 512 bytes) where items may be sequentially stored. It consists of one or more linked frames and can vary in size from file to file and within files.
- Base - The FID (frame identification) of the first frame of the first group in a given file.
- Modulo - The number of groups allocated for a given file.
- Separation - The number of frames initially allocated for each group in a file.

Thus, a set of items (each with a unique item-id) is stored in a file. A file, in turn, is defined by a base, a modulo, and a separation. This basic structure is illustrated in Figure A.

You select the file modulo and separation when the file is initially created. Items are created and modified by the EDITOR, PROC, DATA/BASIC or other processors. Data is stored in items in terms of attributes, multivalues, or subvalues.

A hashing algorithm is used which operates on the item-id (using base, modulo, and separation) to produce an index to the FID of the first frame of the group in which the item is stored. For further information regarding this hashing algorithm, refer to the topic "Item Storage and Hashing Algorithm" in this section.

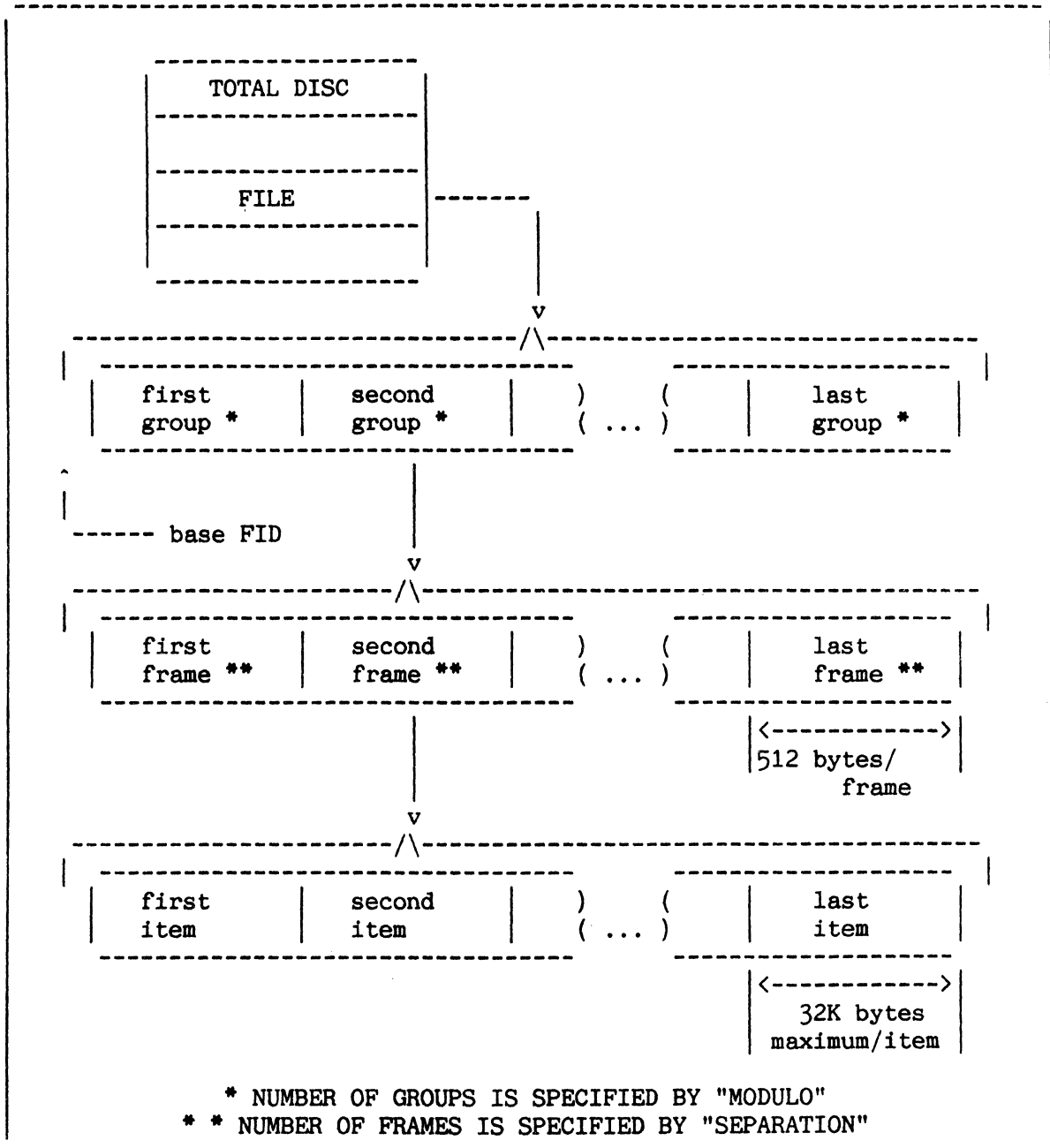


Figure A: General File Structure

ITEM STRUCTURE (PHYSICAL)

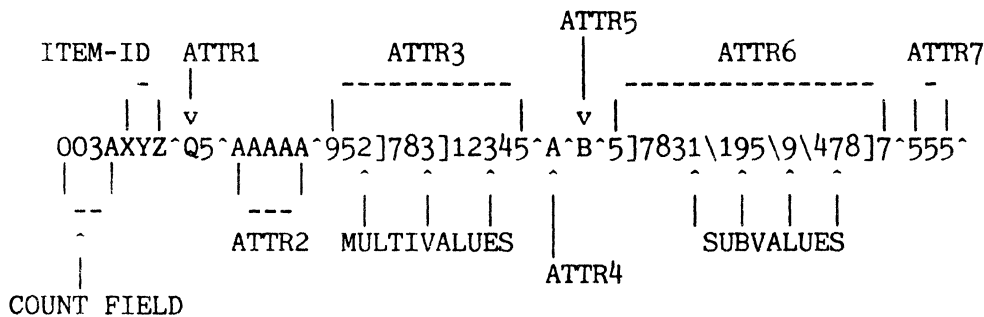
Data within an item is stored as attributes (i.e., fields), multivalued, and subvalues, all of which provide variable length storage. This topic describes the physical item format.

An item consists of one or more variable-length attributes (also known as fields) separated by attribute marks. An attribute mark is an eight-bit byte with a value of X'FE', which prints as an up-arrow or "^". An item is preceded by a four-character hexadecimal count which specifies the total number of characters in the item including the count field, the item-id, and an attribute mark to terminate the item. For example, consider the following item:

002EITEMX^LINE 1^SMITH, JOHN^1234 MAIN STREET^

In this example, the item-id is "ITEMX". It is preceded by "002E" which specifies that there are 002E (46) bytes in the item. Attribute 1 is "LINE 1". Attribute 2 is "SMITH, JOHN". The last attribute (attribute 3) is "1234 MAIN STREET".

An attribute, in turn, may consist of any number of variable-length multivalued separated by value marks. A value mark has an eight-bit value of X'FD', which usually prints as "]". Finally, a multivalued may consist of any number of variable-length subvalues (also known as secondary values) separated by subvalue marks. A subvalue mark has an eight-bit value of X'FC', which usually prints as "\". For example, consider the following item:



The absence of an attribute value is specified by an attribute mark (to maintain the proper attribute sequence) immediately following the attribute mark indicating the end of the previous attribute. The "space" between two adjacent attribute marks can be thought of as representing the absent or null attribute.

Note:

An item-id must be less than 50 characters and the total size must not exceed 32,267 characters.

Figure A illustrates the general form of an item.

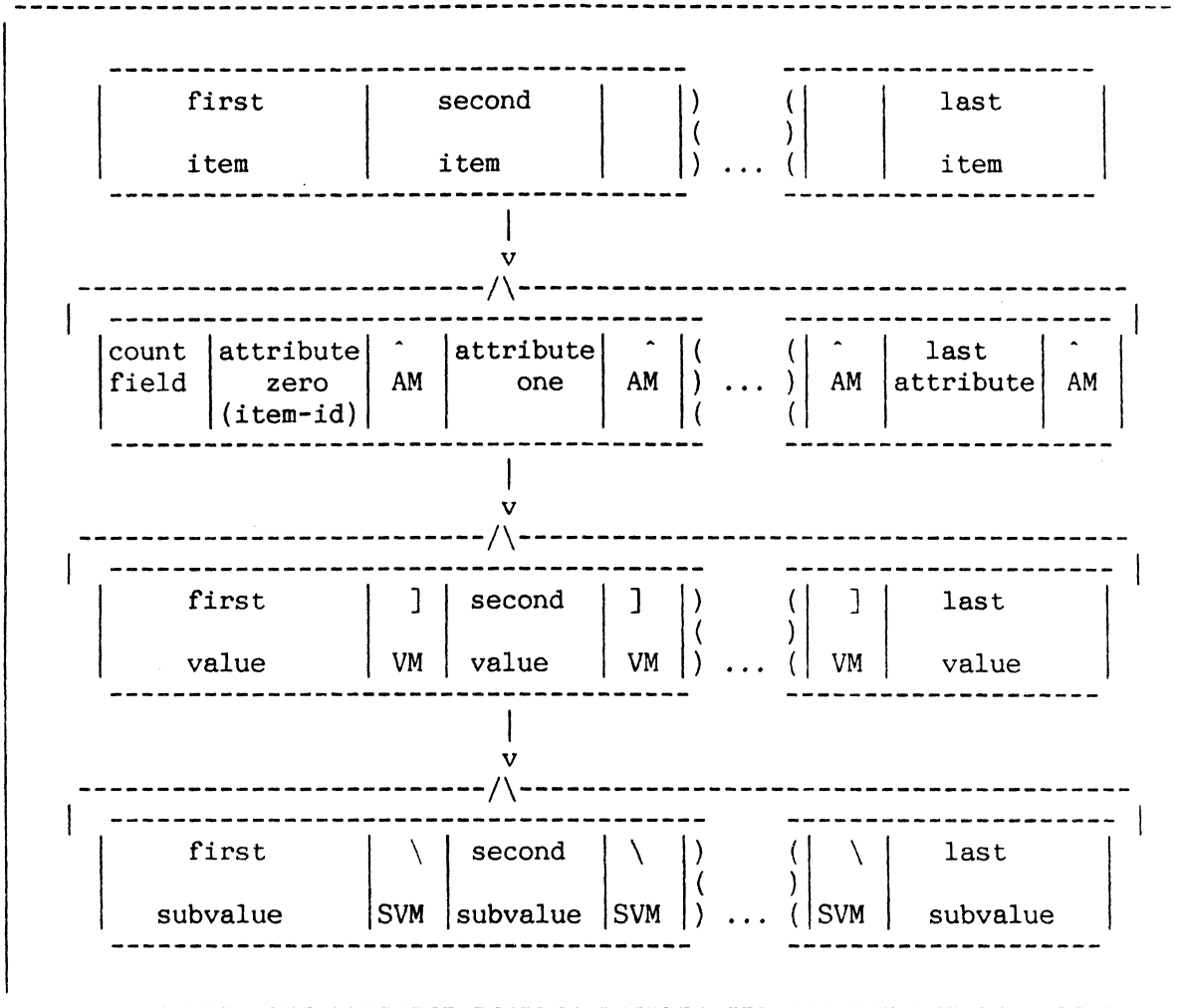


Figure A: General File Item Structure

ITEM STRUCTURE (LOGICAL)

This topic describes the item structure at the logical level.

While it is important to understand the physical item format, in normal system usage items are always accessed at a more abstract level. Files are identified by a file-name. Within a file, items are referenced by the item-id. Attributes are referred to as lines (i.e., Attribute 1 is called line 1). Figure A shows a sample COPY operation where the item with item-id ITEMX (in the file SAMPLE-FILE) is being copied to the terminal. The item is shown to have three attributes (lines) each with sample data.

Utility processors like COPY and the EDITOR deal at the file-item-line level. They make no logical distinction in definition between various lines in an item, other than their implied line numbers.

ENGLISH processors, however, add an additional dimension through the use of the dictionary. The dictionary informs ENGLISH as to the nature of information stored for each of the attributes.

The logical item format is identical for ENGLISH and non-ENGLISH processors. It is your responsibility to ascertain further qualifications, if any, of the various attributes. For example, the item listing in Figure A, is shown in Figure B, as produced by the ENGLISH LIST processor. Here, for example, the file dictionary defines Attribute 2 (line 2) as NAME and Attribute 3 (line 3) as ADDRESS. This permits you to reference data symbolically when, in fact, the actual data stored on file is the same regardless of the processor accessing it. (For further information, refer to the ENGLISH Reference Manual.)

```

: COPY SAMPLE-FILE ITEMX (T) <cr>

      ITEMX <----- Item-id
001  LINE 1 <----- Attribute 1
002  SMITH, JOHN <----- Attribute 2
003  1234 MAIN STREET <----- Attribute 3
  
```

Figure A: Item Listing Via COPY Processor

```

: LIST SAMPLE-FILE 'ITEMX' ATTRIBUTE 1 NAME ADDRESS <cr>

PAGE 1                                09:16:40  12 FEB 1985

SAMPLE-FILE..ATTRIBUTE 1..NAME.....ADDRESS.....

ITEMX      LINE 1      SMITH, JOHN      1234 MAIN STREET
  
```

Figure B: Item Listing Via ENGLISH LIST Processor

ITEM STORAGE AND HASHING ALGORITHM

A computational group hashing technique operates on the item-id, using file parameters, to produce an index to the FID of the group where the item is stored.

The hashing formula used by the system to store or retrieve items is shown in Figure A. The item-id is treated as a variable length string (up to 50 characters) of binary-bytes; these bytes are accumulated sequentially with each partial sum multiplied by 10. Dividing this value by the positive integer MODULO yields an unsigned integer remainder in the range:

$$0 \leq \text{remainder} < \text{modulo}$$

This is then the group number (i.e., 0, 1, 2 ..., up to MODULO-1) where the item is to be stored. Multiplying by the SEPARATION and adding the BASE yields the actual FID of the first frame in the group. (Group, modulo, and separation are defined in the topic titled FILE STRUCTURE.)

After computing a FID to locate the specific group in which the item resides, each item's item-id in the group must be compared for a match. The frames comprising a group are linked both forward and backward. This system facility makes the group appear as a physically sequential string, where items are stored one immediately after another. In fact, any portion of an item may spill across a physically noncontiguous frame boundary. After the last item in the group, there is an additional attribute mark called the group terminator.

When a file is created, it is allocated a primary area of (MODULO * SEPARATION) frames. Thus, this amount of contiguous disk-space is permanently allocated to the file. As the file grows, individual groups may fill up. When this happens, an additional frame is added to the group from a pool of available space. This frame is linked into the group to increase the length of the logically sequential group. Additionally, if a delete or update causes the group to shrink, any unused frames outside the primary area are returned to the pool of available space.

The output in Figure B was generated using the DUMP processor. It shows one group of a sample file with BASE=38186, MODULO=7, and SEPAR=1.

$$FID = BASE + [Remainder ((Function\ of\ item-id)/MODULO)] * SEPARATION$$

Figure A: Hashing Formula

```

      --- forward link                               --- hex link
      |                                               |
      v                                               v
FID:  38186  :  0 38197      0 0  ( 952A : 0 9535  0 0 )

000 .....5.....0011T-ATT^P^80A8^001D*A3^A^03^^^^^T10
050 ^L^10^1^0013SP-STOP^P^10A6^001DSAMC^S^04^S/AMC^^^^
100 ^^L^2^000CSUPP^CH^000BNOT^CD^002ALISTPROCS^PQ^(DIC
150 T PROCLIB LISTPROCS)^001DV/CORR^A^08^^^^^^L^10^1^
200 0018T-RDLBL^P^6033^8024^000AEQ^CC^001E9^S^09^V/TYP
250 ^09^^^^L^2^1^0016BLOCK-TERM^P^1041^0026ERRMSG^Q^S
300 YSPROG^ERRMSG^1^^^^R^5^0016ITEM^P^2^30A0^^NP^0013
350 CHARGES^P^5032^0017C-READ^P^2^94^^CUN^001DV/EDIT^A
400 ^12^^^^^^L^10^1^0011SUM^PW^35^4F^^-1ASSELECT^PB^3
450 5^76^3076^001E2^S^02^A/AMC^02^^^^L^2^1^0029PE^PQ^
500 HPRINT-ERR E

FID:  38187  :  0 0 38186  0      ( 9535 : 0 0 952A 0 )
                                     |
                                     v
000 .....*..RRMSG ^F^A(,999)^P^000EBEFORE^CB^^LW.
050 P2..p3.A..G.Q..]p..L.....Qp...p...O..60..NO..fO.p.
100 F6..B.%..B.A....Q.p....6.r...N.>P&..G.VB.ZB.U.W.G.
150 W.G.P..&G.FB.JB.Ep.....Mw.....q.P....P.^p
200 .p.G.F@DN.IX.P.@DJ:J....1..p.....P..J.G/VF!AOG.t
250 D....!CAPTQMR!R.w..%GZ...+AN^N_ BNG?^K@.FBGT..F6
300 BGD.3..G.FP...U.....G[ >.X.#BJV6.;BGV@HFANVBmVW
350 Z.QG ?.^6."U....B.VWZ.5XBmV..Bgb.BjF.ANV666F..
400 ....F'..F@..FO..F...F...F..&F..'F^...+f.@.OBgVw...
450 ..N:.7NO .....B5..B1..000000..B5..B1..00
500 00..... S
                                     |
                                     group-
                                     terminating
                                     attribute
                                     mark

```

NOTE: The data after the group-terminating attribute mark is random garbage and is ignored by the system.

Figure B: Dump of First Group of Sample File

DATA STRUCTURES

FILE UPDATE: STRUCTURE

A single system routine handles all additions and deletions of items to files. Frames are added to groups as they grow and released as they shrink. Updates to items occur "in place".

Figure A shows a file sparse enough to be contained in its primary space. It has a modulo of three and a separation of one. Items are represented by arrows.

As items are added to the file, new items will be added to the end of some groups (the groups to which these items hash). Also, items within the groups may shrink or grow. The position of items which shrink or grow is not changed, relative to other items in the group. (This is called "update in place").

As groups grow, frames are taken from overflow and appended to the ends of groups (Figure B). The primary space for the file remains unchanged. The appended frames will be singly linked together, as opposed to being contiguous blocks.

When items are deleted, groups shrink and remaining items maintain their relative order. To make unused secondary frames available for other needs of the system, these frames are released to the system available space pool as soon as they become available.

Figure C illustrates the file update routine.

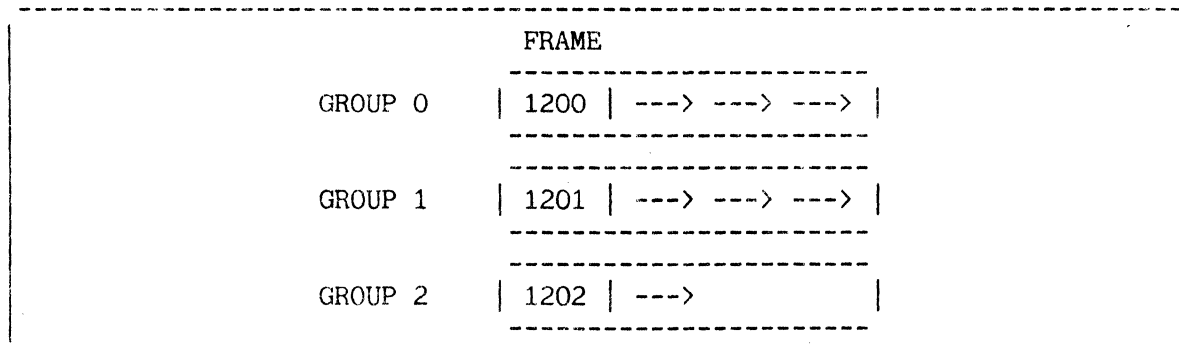


Figure A: File with Only Primary Space

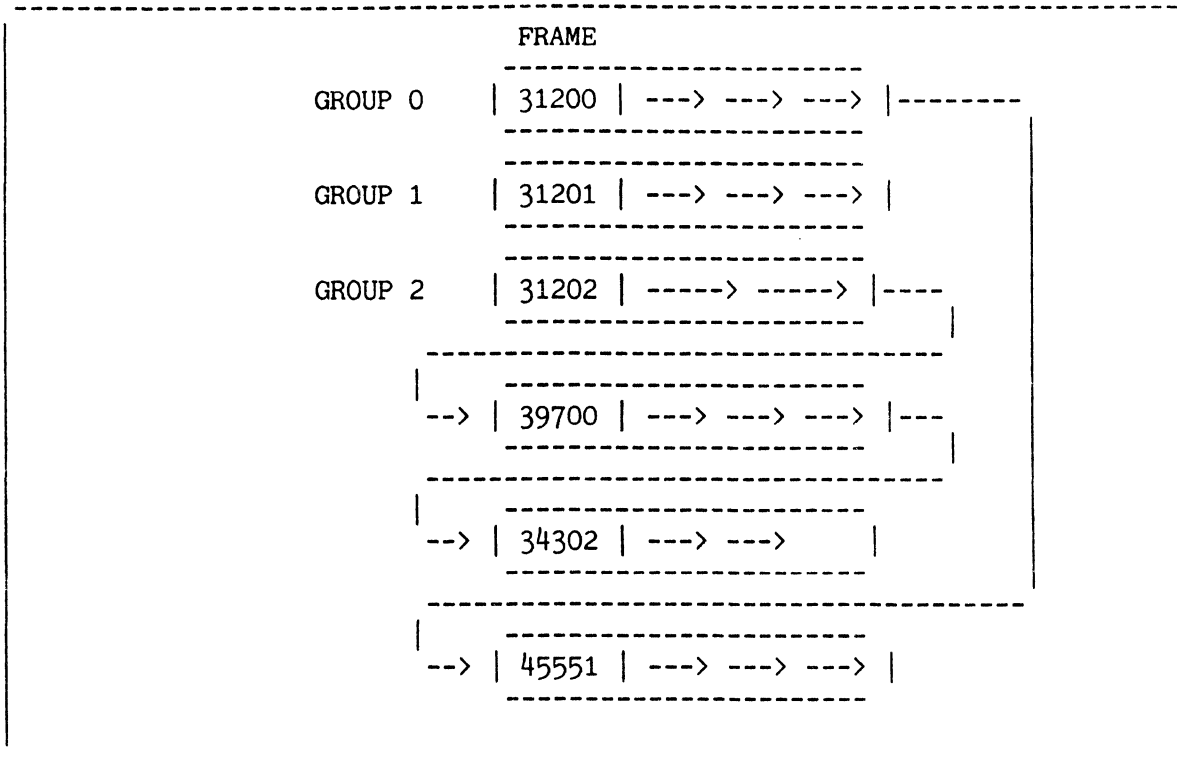


Figure B: File Growth into Secondary Space

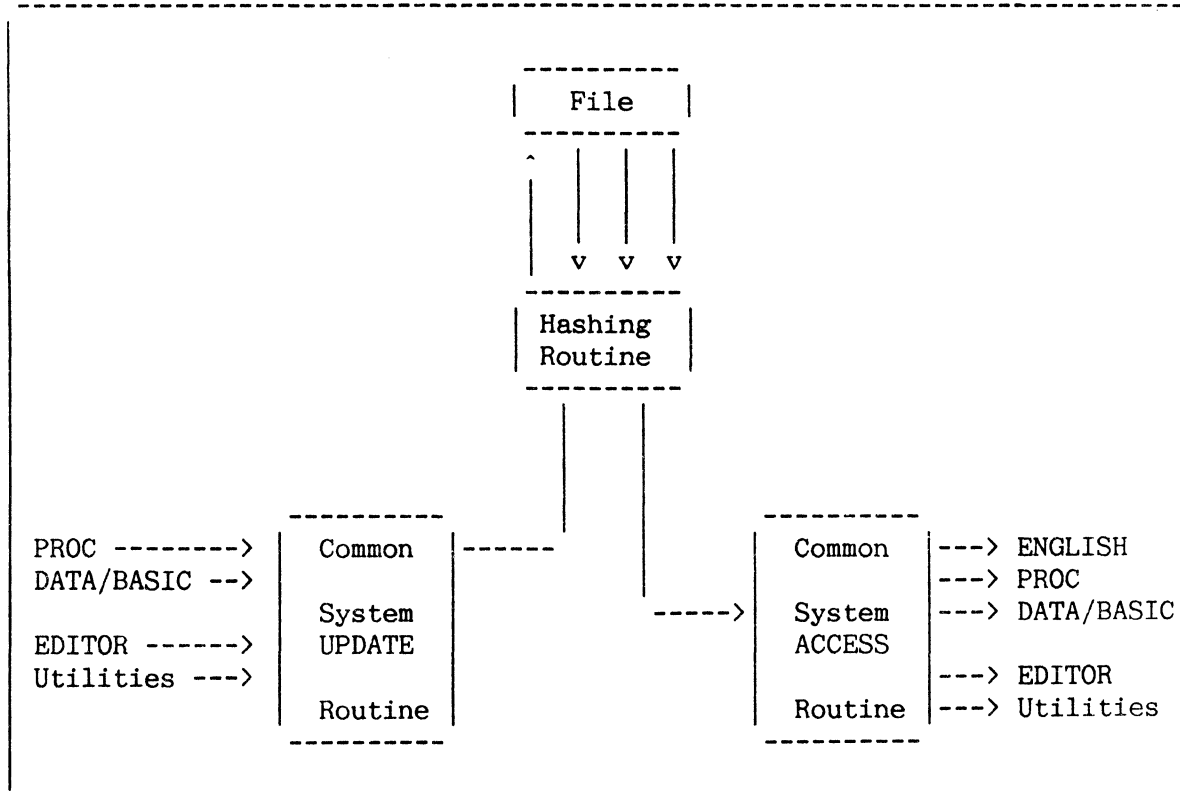


Figure C: Update Routine

3. FILES AND DICTIONARY FILES

=====

| This section principally describes the structure and function of |
dictionary files in the system.

The topics included in this section are as follows:

- the hierarchy of files in the system.
- file and file synonym definition items.
- attribute definition items.
- a summary of the items in the system dictionary, the master dictionaries and in file dictionaries.
- description of the initial files (including dictionary files) on the system and their organisation.

Note: Special items in Master Dictionaries which define verbs, PROCs and various ENGLISH language elements are defined in the following locations respectively:

- . Section 4: TERMINAL CONTROL LANGUAGE
- . PROC Reference manual.
- . ENGLISH Reference manual.

FILE HIERARCHY

| Dictionaries define and describe data within their associated file.
Dictionaries exist at several levels within the system.

The highest level file is called the System Dictionary (SYSTEM). This file is used for system control; it contains pointers to each user's Master Dictionary (M/DICT). The Master Dictionaries comprise the next dictionary level. Each user's account has an M/DICT associated with it; the M/DICT points to dictionary level files in the user's account. Each dictionary level file, in turn, points to its associated data file. Thus, each file has a dictionary level file which defines the file and its contents. In rare cases, the file is a "single-level" file, in which case actual data is stored in the dictionary level file. This structure is not recommended, however, because it does not, in general, improve the processing performance of the system. On the contrary, it can significantly degrade the performance of the formatted save and restore processors. Such "single level" files point back to themselves so they may be referenced as dictionary or data sections interchangeably. Figure A illustrates this general dictionary hierarchy.

The term "file" refers to a mechanism for maintaining a set of like items logically together. The data in a file are normally accessed via the dictionary associated with it. Since the dictionary itself is also a file, it contains items just as a data file does (see Figure B). Items in a dictionary serve to define lower level dictionaries or data files. Some items used in dictionaries are:

- . File definition items
- . File synonym definition items
- . Attribute definition items

File definition items and file synonym definition items are used to define files. Attribute definition items are used to define attributes within items. Each dictionary item consists of attributes (just as file items do); these attributes are used to define a file or data item.

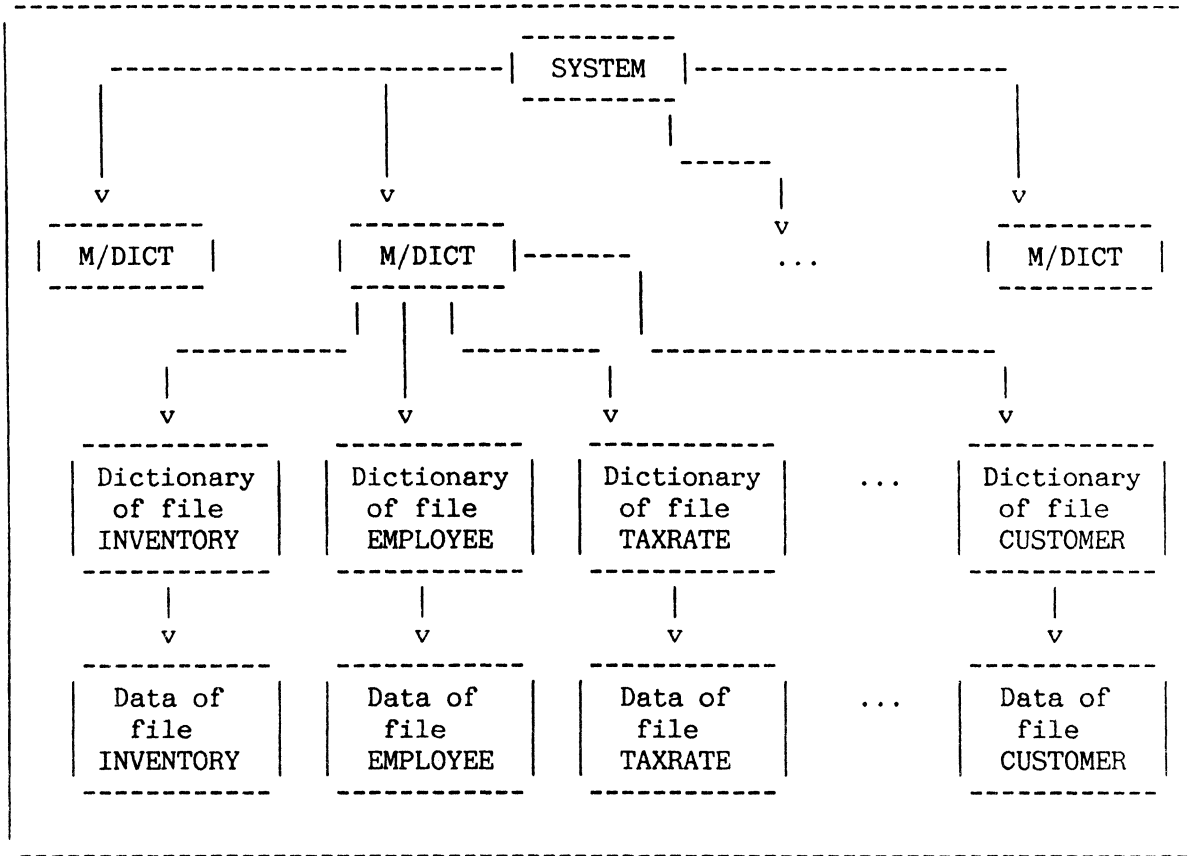


Figure A: Dictionary Hierarchy

<u>D/CODE</u> (Attribute 1)	<u>TYPE OF DICTIONARY ITEM</u>
D	File definition item.
DX	File definition item; the X indicates that the file will be skipped by the file-save processor.
DY	File definition item; the Y indicates that it is not required to save the data portion of the file on a file-save.
A, S, or X	Attribute definition item.
Q	File synonym definition item.

Figure B: Dictionary Items

FILE DEFINITION ITEMS

 | File definition items define lower level dictionary files or data files. |
 | File definition items are specified by a D/CODE of "D", "DX", or "DY". |
They are created automatically by the CREATE-FILE verb.

At the SYSTEM dictionary level, file definition items define various system files (see the topic INITIAL SYSTEM FILES/DICTIONARIES), each user's M/DICT, and an item containing the logon message. File definition items in the M/DICT define the dictionary level files which in turn may contain a file definition item which defines the associated data area. The item-id and each attribute of the file definition item contains the information used to describe the lower level dictionary or file:

- Item-id - The item-id of a file definition item is the file-name of the dictionary or data file being pointed to.
- Attribute 1 - This is the D/CODE attribute; it must contain a "D", "DX", or "DY". See description of each in previous topic.
- Attribute 2 - This is the F/BASE attribute; it must contain the Base FID (as a decimal number) of the defined file.
- Attribute 3 - This is the F/MOD attribute; it must contain the modulo (as a decimal number) of the defined file.
- Attribute 4 - This is the F/SEP attribute; it must contain the separation (as a decimal number) of the defined file.
- Attribute 5 - This is the L/RET attribute; it may contain an optional retrieval security code (see Section 5, in the topic SECURITY).
- Attribute 6 - This is the L/UPD attribute; it may contain an optional update security code (See Section 5).
- Attributes 7 through 8 - These attributes are identical to those used in attribute definition items; refer to the topic, ATTRIBUTE DEFINITION ITEMS, in this section.
- Attribute 9 - This is the V/TYP attribute which indicates the type of justification for the column of item-ids printed by ENGLISH. A value is mandatory and may be one of the following:

- L - Left justified
- R - Right justified
- U, R, T, B - May be included in user identification item in System Dictionary (see Section 5)

- Attribute 10 - This is the V/MAX attribute which indicates the width of the item-id column printed by ENGLISH. A numeric value is mandatory.
- Attributes 11 and 12 - Reserved.
- Attribute 13 - This is the F/REALLOC attribute, which allows for the reallocation of the physical extent of a file during a system file-restore process. The format of this specification is (m,s); where "m" and "s" are decimal numerics specifying the new modulo and new separation parameters of the file. Another way the physical extent of a file can be altered is to COPY the file to a new file. The first character may be a 'B' indicating to the file save/restore processor that the file is a pointer-file to saved lists of item-ids, catalogued DATA/BASIC programs and shared data items. The 'B' is only valid at the System Dictionary level.

WARNING: Attributes 2, 3 and 4 of a file definition item should never be altered under any circumstances.

Figure A illustrates a sample file definition item which defines the dictionary level file for the INVENTORY file; this item has an item-id of INVENTORY and is stored in the user's M/DICT. The figure also shows the file definition item which defines the data area of the INVENTORY file; this item has the special item-id 'DL/ID' and is stored in the file dictionary.

Note that in a single-level file, the DL/ID item may be absent in the file dictionary, or it may point back to the dictionary itself.

<u>Item 'INVENTORY' in M/DICT</u>	<u>Item 'DL/ID' in dictionary of INVENTORY</u>
001 D<-----	001 D<----- D/CODE=D (file definition item)
002 2537<-----	002 15132<- Base FID of file.
003 1<-----	003 13<---- Modulo of file.
004 1<-----	004 1<---- Separation of file.
005 /;P123<-----	005 <----- File access retrieval lock-code.
006 UPDATE*LOCK!<-----	006 <----- Update lock/code.
007 <-----	007 <----- Conversion (null).
008 <-----	008 <----- Correlatives (null).
009 L<-----	009 L<---- Left justification.
010 10<-----	010 10<---- Column width.
	011 <----- (null).
	012 <----- (null).
	013 (3,1)<- Reallocation parameters.

Figure A: Sample File Definition Item in M/DICT.

FILE SYNONYM DEFINITION ITEMS

File synonym definition items allow access to files in another user's account or provide alternate file-names. File synonym definition items are specified by a D/CODE of "Q".

The item-id and attributes of a file synonym definition item (also known as a 'Q-pointer') are:

- Item-id - The item-id of a file synonym definition item is the synonym name by which the defined file may be referenced.
- Attribute 1 - This is the D/CODE attribute; it must contain a "Q".
- Attribute 2 - This attribute may contain the name of the account in which the actual file definition is to be found (the account name is an entry in the System Dictionary). If null, the account containing the Q-pointer is implied.
- Attribute 3 - This is the S/NAME attribute; it must contain the item-id of the actual file definition item to which the synonym equates (i.e., the actual file-name). If null, the file containing the Q-pointer is implied.
- Attribute 4 through 12 - Must be null.

Note that a 'Q-pointer' must point directly to a 'D-pointer'; one 'Q-pointer' cannot point to another 'Q-pointer'.

Figure A illustrates a sample INVENTORY file synonym definition item which allows you access to the file in the account named USER2; you can reference this file via the synonym file-name INV. Figure B illustrates the overall relationship of this plus an additional file synonym definition item. Note that SYNUSER3 points to the M/DICT for USER3 since the value for Attribute 3 is null.

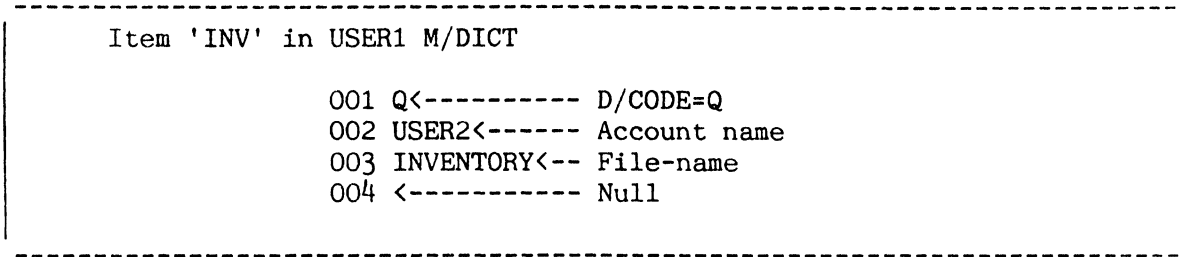


Figure A: Sample File Synonym Definition Item

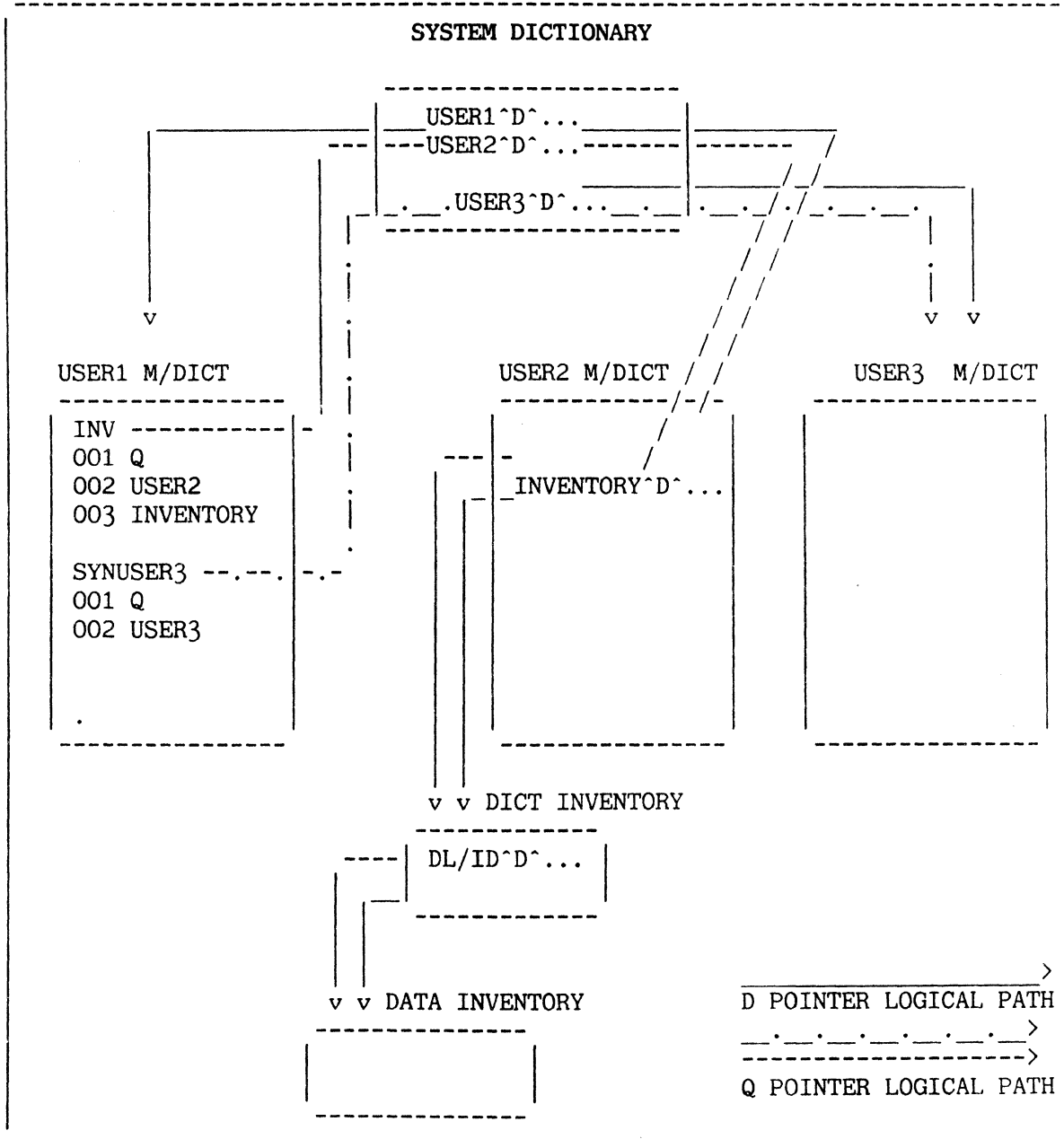


Figure B: Sample File Synonym Definition Interrelationship

ATTRIBUTE DEFINITION ITEMS

 Attribute definition items define various attributes (fields) in the data items for use by the ENGLISH processor. Attribute definition items are specified by a D/CODE of "A", "X" or "S", and are typically formed with the EDITOR.

An attribute definition item defines a specific attribute (field) for subsequent access via the ENGLISH processor. Each attribute definition item has a value, called the Attribute Mark Count (AMC) which acts as a pointer to the data field defined by it. The AMC is simply the attribute number (for example, AMC=5 defines Attribute 5, AMC=22 defines attribute 22, etc). An attribute definition item defines a corresponding attribute for all items in the file and provides a symbolic name for that attribute. Attribute items with numeric and sequential item-ids (1,2...) are used for the automatic attribute inclusion feature of ENGLISH (when the output-specification is omitted). "A" or "S" codes cause columns to be listed, and act identically with one minor exception, as noted under "Attribute 3" below. "X" codes preserve the chain of numeric item-ids but are otherwise ignored. Attribute definition items are constructed as follows:

- | | | |
|------------------------|---|---|
| Item-id | - | The item-id is the name desired for the defined attribute. This name may be used in ENGLISH input statements to reference the attribute. |
| Attribute 1 | - | This is the D/CODE attribute; it must contain an "A", "S" or "X". |
| Attribute 2 | - | This is the A/AMC attribute; it contains the AMC of the defined attribute (i.e., specifies which attribute is being defined). |
| Attribute 3 | - | This entry will be used as the column heading. A multiple-line heading can be specified by separating individual lines with value marks (<c>], X'FD'). If this attribute is null and the attribute definition item is an "A" type, then the column heading will default to the item-id of the attribute definition item. If this attribute is null and the attribute definition item is an "S" type, then there will be no column heading. <u>This is the only distinction between "A" and "S" type attribute definition items!</u> |
| Attributes 4 through 6 | - | Reserved |
| Attribute 7 | - | This is the V/CONV attribute; it contains the conversion specification used to convert from the processing format to the output format, and in some cases, vice-versa. Refer to the ENGLISH Reference Manual for further details. |

- Attribute 8 - This is the V/CORR attribute; it contains the correlative specification used to convert from the internal format to the processing format (refer to the ENGLISH Reference Manual).
- Attribute 9 - This is the V/TYP attribute; it defines the justification for output. A value is mandatory and may be one of the following.
- I - Left justified, wrapped text is indented one space
 - L - Left justified, no specified type
 - R - Right justified, no specific type
 - T - Text data, left justified. Fold at blanks.
 - U - Left justified, print entire value on line, ignoring column boundaries.
- Attribute 10 - This is the V/MAX attribute; it defines the column width in which values for the attribute will be printed. An entry is a decimal numeric, and is mandatory. Note, however, that if this attribute is less than the width of the column heading specified in Attribute 3, the width of the column heading will override the width specified here.
- Attributes 11 through 20 - Used by SCREENPRO.

Figure A illustrates two sample attribute definition items: an item which defines the third attribute of each item in the INVENTORY file, and an item which defines the fifth attribute of each item in the EMPLOYEE file.

<u>Item QUANTITY in dictionary of file INVENTORY</u>	<u>Item RATE in dictionary of file EMPLOYEE</u>
001 A <--- D/CODE = A	001 A <----- D/CODE = A
002 3 <--- AMC (defines 3rd attrib.)	002 5 <----- AMC(defines 5th attrib.)
003 QTY <- Column heading	003
004	004
005	005
006	006
007 MD2 <- Conversion spec.	007 MD2& <-- Conversion spec.
008 G*1 <- Correlative spec.	008
009 R <--- Right justified	009 L <----- Left justified
010 10 <-- Column width	010 10 <---- Column width

Figure A: Sample Attribute Definition Items

FILES AND DICTIONARY FILES

SUMMARY OF DICTIONARY ITEMS

This topic summarizes items used in various dictionaries.

FILE AND ATTRIBUTE DEFINITION ITEMS

The file definition items, file synonym definition items and attribute definition items which may be used as dictionary entries are summarized in Figure A.

SYSTEM DICTIONARY (SYSTEM) ITEMS

There is one and only one System Dictionary for each system. The SYSTEM dictionary should contain only items with D/CODE=D, DX, DY, or Q, representing user accounts or special system files. (The exception is the LOGON message; refer to the topic SYSTEM MESSAGE FORMATS: LOGON MESSAGE.) The Logon processor uses these "D" type items to verify users attempting to logon to the system. Only one "D" type item should be present for each account; if more than one user-name is to be established for the same user-account, additional names should be file synonym definition ("Q" type) items.

MASTER DICTIONARY (M/DICT ITEMS)

There is one M/DICT for each account. The M/DICT, like any other file or dictionary, is made up of items. Items with D/CODE=A or S define the attribute formats for all dictionaries including the M/DICT itself; their formats are identical to those for dictionary level files. The file-defining items (D/CODE=D) point to various files defined for the account.

In addition to elements in the M/DICT which define files and attributes, there are items which define verbs, PROCs, and various ENGLISH language elements. Each of these items has a coding structure which uniquely identifies it; refer to the following sources for their respective definitions:

- . Section 4: TERMINAL CONTROL LANGUAGE
- . PROC Reference Manual
- . ENGLISH Reference Manual

ATTRIBUTE NUMBER	NAME	FILE DEFINITION ITEM	FILE SYNONYM DEFINITION ITEM	ATTRIBUTE DEFINITION ITEM	ATTRIBUTE CONTINUATION DEFINITION ITEM
1	D/CODE	D,DX or DY	Q	A or S	X
2	F/BASE or A/AM	Base FID of file	Account name (null implies current acct)	AMC	AMC
3	F/MOD or S/NAME	Modulo of file	File-name (null implies file containing Q-pntr)	Alternate heading	Not used and reserved
4	F/SEP or S/AMC	Separation of file	Must be null	<---- Not used and ----> reserved	
5	L/RET	Retrieval lock code(s)	<----- Not used and reserved ----->		
6	L/UPD	Update lock code(s)	<----- Not used and reserved ----->		
7	V/CONN	Conversion specification(s)	Not used and reserved	Conversion specification(s)	
8	V/CORR	Correlative specification(s)	Not used and reserved	Correlative specification(s)	
9	V/TYP	Justification type code	Not used and reserved	Justification type code	
10	V/MAX	Column width	Not used and reserved	Column width	
11	<----- Not used and reserved ----->				
12	<----- Not used and reserved ----->			Used by SCREENPRO	
13	F/REALLOC	Reallocation specification	<----- Not used and reserved ----->		
14-20	<----- Not used and reserved ----->			Used by SCREENPRO	Not used and reserved

Figure A: Summary of File and Attribute Definition Items

FILES AND DICTIONARY FILES

INITIAL SYSTEM FILES/DICTIONARIES

| Certain files are essential to the operation and maintenance of the |
system. Figure A shows some of these files.

SYSPROG ACCOUNT

The System Programmer account SYSPROG is the only account needed to maintain the system. The prototype master dictionary NEWAC, which is used to create new user M/DICTs, is defined in this account, as are SYSPROG-PL, SYSTEM-OBJECT and TERMINAL-OBJECT.

SYSPROG-PL File

This file, defined from SYSPROG, contains the system maintenance PROCs used to perform file-saves, account-saves, account-creations and so on. These PROCs can be used from the SYSPROG account. Refer to Section 10, SYSTEM MAINTENANCE PROCEDURES, for descriptions of the facilities available on this account.

NEWAC File

This file, defined from SYSPROG, is a prototype M/DICT used as the model on which a new user's M/DICT is based. The NEWAC file is used by the CREATE-ACCOUNT PROC.

TERMINAL-OBJECT File

This file, defined from SYSPROG, contains the object code for Terminal Executives (not used in the US).

SYSTEM-OBJECT File

This file contains assembly object code for system software. These items are loaded via the MLOAD verb into the ABS region from which the code can be executed. Some items in this file are used to create the coldstart (bootstrap) portion of a file-save tape.

ERRMSG File

This single-level file, defined from SYSTEM, contains the system messages. Items in the ERRMSG file are listed in an appendix to this manual; they consist both of error messages and of informative messages. See SYSTEM MESSAGE FORMATS in Section 5 for the format of items in this file.

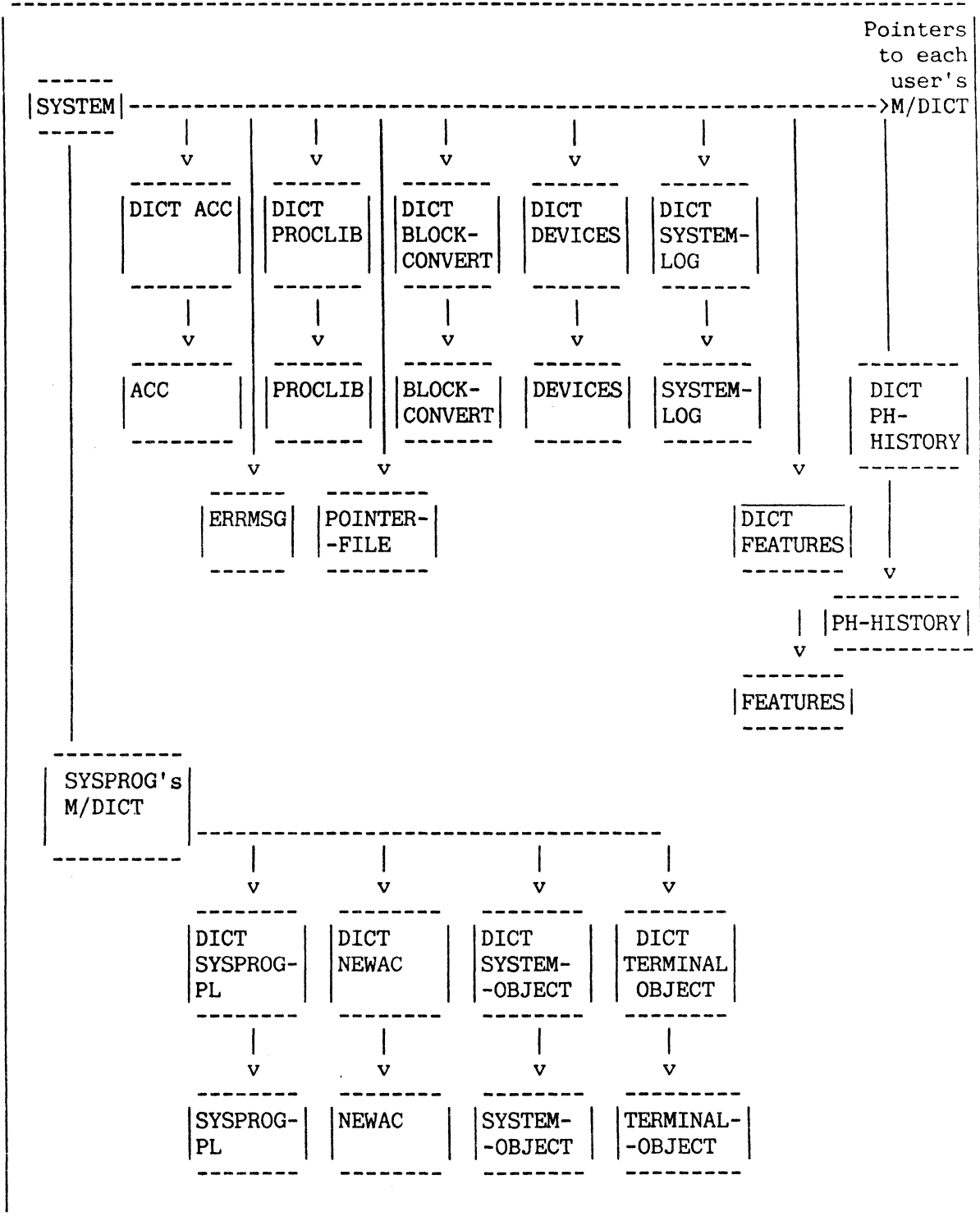


Figure A: General Organization of Initial Files/Dictionarys

FILES AND DICTIONARY FILES

ACCOUNTING HISTORY File

The Accounting History File (ACC) contains the accounting history for the system, as well as entries that describe currently active users. The formats of these entries are described in Section 5, LOGON/LOGOFF. The Accounting History File should be cleared periodically.

PROCLIB File

The PROCLIB file contains a number of common PROCs (for example, LISTU, CT, etc.). Each M/DICT contains a Q-pointer to PROCLIB, along with short versions of the common PROCs (using the same PROC names), which merely transfer control to the corresponding PROCs in PROCLIB. For further information, refer to the PROC Reference Manual.

BLOCK-CONVERT File

This file contains items used by the BLOCK-TERM and BLOCK-PRINT verbs to convert characters to a block format.

POINTER-FILE File

This file contains pointers to saved select lists, cataloged DATA/BASIC programs, etc. Do not modify this file! Attribute 13 of the file definition item in the SYSTEM dictionary must contain a 'B' as the first character.

DEVICES File

This file contains parameters for peripheral devices which are connected to the system. See PORT CHARACTERISTICS MAINTENANCE in Section 9 for details.

SYSTEM-LOG File

This file is a log of system and disk errors which have occurred.

FEATURES File

This file contains details of features on the system.

PH-HISTORY File

This file contains details of all TIPH (phantom) processes.

4. TERMINAL CONTROL LANGUAGE
=====

| This section discusses the interface between a terminal and the |
associated process.

The topics included in this section are as follows:

- input statements.
- verb definition items in master dictionaries.
- TCL-II input statements.
- TCL-I input statements.
- program interruption (debug) facility.

TERMINAL CONTROL LANGUAGE

OVERVIEW

| The Terminal Control Language (TCL) is the primary interface between |
you and various processors.

Most processors are activated directly from TCL by a single input statement, and return to TCL after completion of processing. Some processors (the EDITOR for example) retain control of the terminal until explicitly exited, at which point they return control to TCL. TCL prompts you by displaying a colon (:). This is referred to as the "TCL prompt character". Input statements are constructed by typing one character at a time on the terminal until the carriage return or line feed key is pressed, at which time the entire line is processed by TCL.

A type-ahead feature allows further commands to be entered while the first command is being processed.

TCL expects the first word of an input statement to be a "verb" or a PROC name. There are three types of verb:

- . ENGLISH
- . TCL-I
- . TCL-II

One of the system's powerful features is your ability to customize each user vocabulary. Since verbs and PROCs reside in each user's Master Dictionary (M/DICT), the vocabulary may be added to or deleted from without affecting the other users. In addition, an unlimited number of synonyms may be created for each verb or PROC.

Figure A presents an overall view of the intraprocessor linkages within the system. This figure illustrates the flow of control between TCL and various processors.

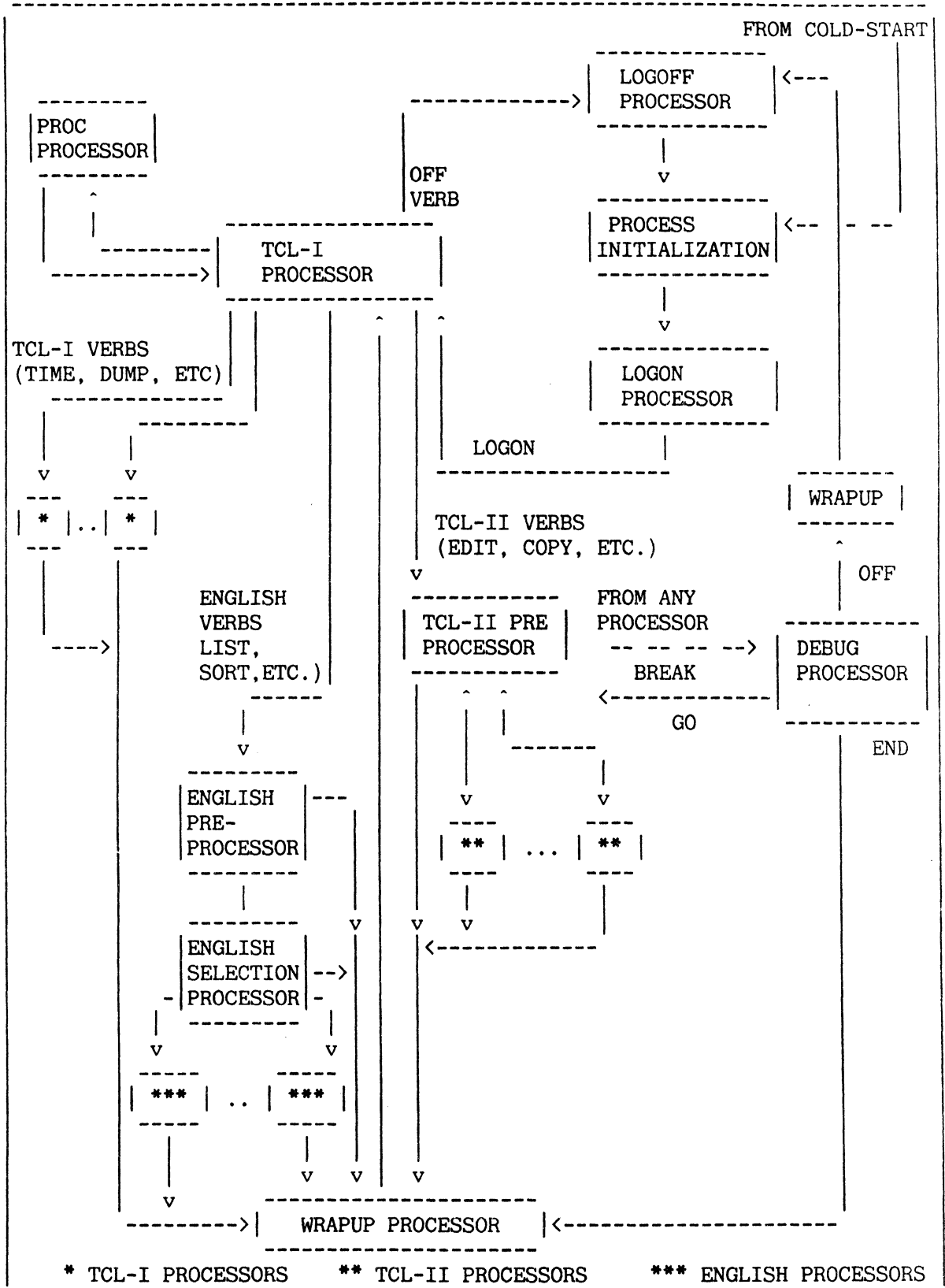


Figure A: Overall View of Intraprocessor Linkages

TERMINAL CONTROL LANGUAGE

INPUT STATEMENTS

TCL operates on one input statement at a time. An input statement may
comprise multiple lines. However, the statement must begin with a
verb or PROC name and may contain only that one verb or PROC name.

The system normally operates in the "full-duplex" mode of communication with each terminal. "Full-duplex" means that data is being transmitted between the terminal and the computer in both directions simultaneously. Additionally, the system operates in what is known as an "Echo-Plex" environment. This means that each data character input by the terminal is echoed back to the terminal by the computer. You are thus assured that the data character displayed on the terminal is identical to the data character stored by the computer. It is also possible to stop terminal echo by using the HUSH verb. Terminal echo is restored with the NOHUSH verb.

TCL handles only complete input lines to be processed by the software. You compose your input statement with no action being taken until TCL detects a carriage return character. If a carriage return is not detected, data characters are assembled in your input buffer until 140 characters have been entered, at which time TCL automatically generates a carriage return.

In addition to the standard ASCII character set recognized by TCL (refer to the Appendix), special operations are performed when control characters are detected. The control characters listed in Figure A perform editing functions; all other control characters are deleted from the input line that is passed to lower processors, but remain in the original input line. Control characters are represented in this manual by preceding the character with the symbol <c>. This means that the <CONTROL> or <CTRL> key must be pressed while typing the desired character. (The <SHIFT> key may also have to be pressed simultaneously if the desired character appears on the upper half of a key.) For example, <c>] indicates that "]" must be generated as a control character.

While analyzing your input statement, TCL copies the statement into a buffer while performing the following:

- . The first word of the sentence is looked up in your Master Dictionary (M/DICT), but is not copied.
- . Redundant blanks surrounding all words in the statement are deleted.
- . Character strings surrounded by single quotes (') or double quotes (") are identified and copied verbatim, including redundant blanks.

Figure B defines the options that may be used with TCL-I and TCL-II verbs.

Unless defeated with an "N" option (see Figure B), listings to the terminal normally pause at the end of each screen and wait for a keystroke before continuing with the next screen. The control character <c>E has a special meaning if input at this pause, or (when typeahead is enabled) immediately before this pause: the listing is terminated and control is returned to PROC or TCL. DATA/BASIC also responds to this control character if a heading or

footing is enabled, i.e. a paged listing is being produced.

<u>CONTROL CHARACTER</u>	<u>FUNCTION</u>
<c>H	Backspace. Deletes last character typed in and allows you to re-enter character. (May also be set to non-deleting backspace.)
<c>X	Cancels all typed-ahead commands. (May also be set to cancel just the line currently being typed in.) Also terminates a paged listing, similarly to <c>E.
<c>R	Retype. Causes entire line currently being typed in to be retyped.
<c>_	Line Continuation. Typed as the last character in a line, allows continuation of the line onto the next physical line. Must be immediately followed by a carriage return or line feed.
<c>E	The listing is terminated and control returned to PROC or TCL. DATA/BASIC responds to this control character if a heading or footing is enabled.

Figure A: Special Control Characters

N	Inhibit automatic pause for carriage-return at end of page (NOPAGE).
I	List item-ids while performing copy, dump, etc.
P	Produce output on printer (spooler).

Figure B. TCL-I Option Specifications

VERB DEFINITION ITEMS IN M/DICT

Each TCL-I, TCL-II, or ENGLISH verb is defined as an item in your Master Dictionary (M/DICT).

Each verb definition resides as an item in your M/DICT. The item-id of a verb definition item is the verb name itself. The attributes used in a verb definition item are defined in Figure A.

As an example of a verb definition item, the following item (stored as item 'LIST' in your M/DICT) defines the ENGLISH verb LIST:

Item 'LIST' in M/DICT

001 PA
002 35
003 4D

The following verb definition item defines the TCL-II verb MLIST:

Item 'MLIST' in M/DICT

001 PY
002 2
003 20
004
005 CU

As a final example, the following verb definition item defines the TCL-I verb TIME:

Item 'TIME' in M/DICT

001 PZ
002 3033

You may create any number of synonyms for verb definition items (and may even remove the predefined verb definition items), thereby creating your own vocabulary. Synonyms may be created by copying the verb definition item into another M/DICT item with the desired synonym name as the item-id.

<u>ATTRIBUTE NUMBER</u>	<u>DESCRIPTION</u>
0 <-----	This is the item-id, which is the name of the verb
1 <-----	Must contain: Pc P identifies the M/DICT item as a verb definition item. The single character c is passed to the defined processor. (Note: If c is "Q", the item is a PROC, not a verb; in this case, the "Q" may be followed by an "N". Refer to the PROC Manual for further information.)
2 <-----	This attribute defines the processor entry point to which TCL passes control (i.e., the mode-id in hex). An ENGLISH verb has an entry of: 35 A TCL-II verb has an entry of: 2 A TCL-I verb has an entry of: xxxx where xxxx is the entry point.
3 <-----	Secondary transfer point. Use depends on Attributes 1 and 2.
4 <-----	Tertiary transfer point. Use depends on Attributes 1 and 2.
5 <-----	TCL-II parameter string. These parameters govern treatment of the items retrieved by TCL-II verbs to be passed to the processor whose entry point is defined in Attrib.3. Parameter may be any of the following: C - Copy item to a work area E - Blanks are compressed on file, expanded for processing F - Pick up file parameters only (ignore item-list) N - Okay if item is not on file P - Print item-id if multiple items are specified U - Items will be updated by processor Z - Final entry required after processing last item.

Figure A: Verb Definition Item in M/DICT.

TCL-II INPUT STATEMENTS

The format for forming a TCL-II input statement is more restrictive than for an ENGLISH statement (refer to the ENGLISH Reference Manual). This enhances processing speed since statement parsing is quicker.

Selection-criteria and output-specifications are not allowed in TCL-II statements. The file-name (or DICT file-name) must immediately follow the TCL-II verb. Item selection is more restricted, since each item-id must be explicitly named in the statement (or, alternatively, all items may be specified via use of the asterisk (*) character, or an item-list must be supplied by a SELECT-type command).

Figure A illustrates the general form of a TCL-II statement. The file-name specifies the desired file. The DICT option specifies the dictionary portion of the file. The item-list is made up of one or more item-ids, separated by one or more blanks. If an item-id contains embedded blanks or parentheses, it must be surrounded by single quotes. All items in a file may be specified by using an asterisk (*) as the item-list. Alternatively, the item-list may be supplied by an immediately-preceding SELECT, SSELECT, GET-LIST or other command which generates a 'SELECT' list. Options, if specified must be enclosed in parentheses at the end of the input line. Multiple options may be separated by commas. The specified options are passed to the appropriate TCL-II processor. Figure B illustrates a sample TCL-II input statement.

Some TCL-II verbs are listed in Figure C. For further information on these verbs, refer to the following sections of this manual:

- . FILE MANAGEMENT PROCESSORS
- . MAGNETIC TAPE UNITS
- . SPECIAL SYSTEM UTILITIES
- . SYSTEM MAINTENANCE PROCEDURES: SYSPROG ACCOUNT

Also, refer to the following manuals:

- . ENGLISH Reference Manual
- . DATA/BASIC Reference Manual
- . PROC Reference Manual
- . EDITOR Reference Manual

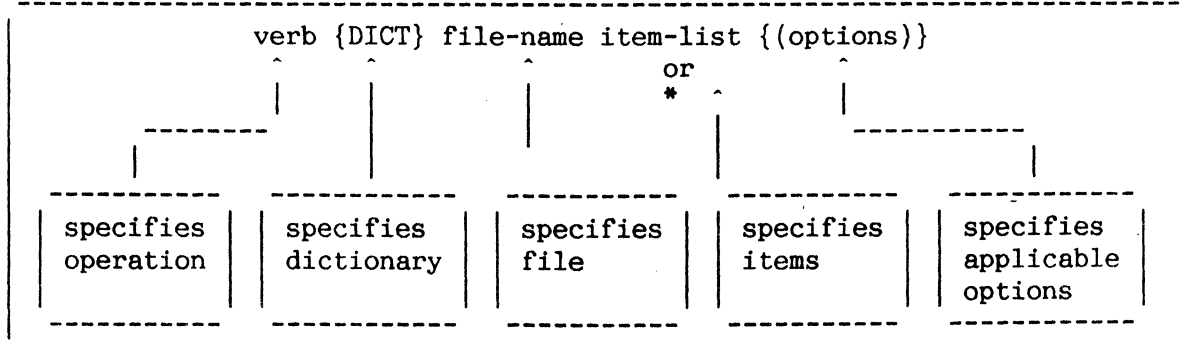


Figure A: General Form of TCL-II Input Statement

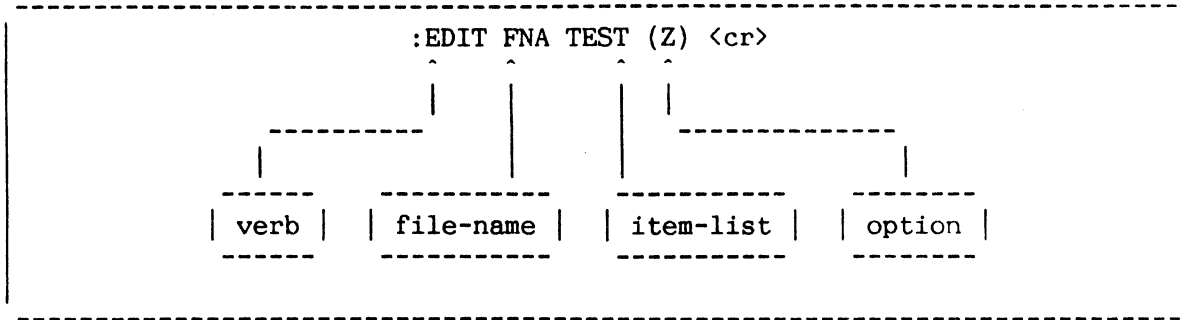


Figure B: Sample TCL-II Input Statement

<u>OPTION</u>	<u>DESCRIPTION</u>
N	Inhibit automatic pause for <cr> at end of page (NOPAGE).
I	List item-ids while performing COPY, DUMP, etc.
P	Produce output on printer (spooler).

Figure C: Sample TCL-II Option Specifications

<u>VERB</u>	<u>DESCRIPTION</u>
COLDDUMP	Create coldstart (bootstrap) tape.
COMP	Compare items
COPY	Copy data files and dictionaries.
EBASIC	Compile compressed DATA/BASIC program.
ECOPY	Expand compressed items.
EED	Edit compressed items.
FIX-FILE-ERRORS	Repair group format errors.
GROUP	Display a file's group and item structures.
ITEM	Display an item's group structure.
SAVE	Backup system.

Figure D: Some TCL-II Verbs

TCL-I INPUT STATEMENTS

Verbs which have a code other than "2" or "35" in Attribute 2 of their respective M/DICT verb definition items are called TCL-I verbs. TCL-I verbs are used to form TCL-I input statements. They may have an option specification of the form: (options).

A TCL-I input statement must begin with a TCL-I verb and end with a carriage return. Some TCL-I verbs additionally allow for various parameter specifications. A selection of TCL-I verbs is listed in Figure A. For further information regarding these verbs, refer to the following sections in this manual:

- . LOGON/LOGOFF
- . FILE MANAGEMENT PROCESSORS
- . SPOOLER
- . MAGNETIC TAPE UNITS
- . SPECIAL SYSTEM UTILITIES
- . SYSTEM MAINTENANCE PROCEDURES: SYSPROG ACCOUNT

<u>VERB</u>	<u>DESCRIPTION</u>
:ABSLOAD	Loads executable assembly code (ABS) from tape.
:ABS/FILES	Loads ABS and files from tape.
:FILES	Restores files from tape.
:INIT-SYSTEM	Initialise system locks.
:RESTART-SPOOLER	Restarts spooler process.
ACCOUNT-RESTORE	Restores user account from tape.
ADDD, ADDX	Adds decimal or hexadecimal numbers.
ASSIGN	Assigns a device to a port.
BLOCK-PRINT	Sends block characters to printer.
BLOCK-TERM	Sends block characters to terminal.
BUFFERS	Displays status/contents of memory buffers.
CHARGE-TO	Keeps track of computer usage.
CHARGES	Prints current computer usage.
CLAIM	Returns linked overflow to contiguous overflow.
CLEAR-BASIC-LOCKS	Resets execution locks (0-63).
CLEAR-FILE	Removes all file items from a file or dictionary.
CREATE-FILE	Creates a new file.
DELETE-FILE	Deletes an entire file.
DISCIO	Produces disk activity report.
DIVD, DIVX	Divides decimal or hexadecimal numbers.
DTX	Converts decimal number to hexadecimal.
DUMP	Dumps virtual frames to terminal.

Figure A: Some TCL-I Verbs

<u>VERB</u>	<u>DESCRIPTION</u>
HUSH	"Hushes" terminal echo.
LIST-LOCKS	Displays group locks.
LOGTO	Switches to another user account.
MESSAGE	Communicates to other users.
MSG	Same as MESSAGE.
MULD, MULX	Multiplies decimal or hexadecimal numbers.
NOHUSH	Restore terminal echo.
OFF	Terminates user's session.
P	Toggles printing at terminal on and off.
P-ATT	Attaches line printer.
P-DET	Detaches line printer.
POVF	Prints available overflow report.
SET-DATE	Sets system date.
SET-TERM	Sets terminal/printer characteristics for all lines.
SET-TIME	Sets system time.
SLEEP	Sets a process into quiescent state for specified time.
SP-ASSIGN	Assigns print job(s) to form queue and assigns option(s) to print job(s).
SP-CHECKQ	Counts number of hold files.
SP-EDIT	Examines closed print jobs.
SP-EJECT	Ejects line printer pages.
SP-KILL	Cancel current spooler output on designated device.
SP-RESUME	Recovers from spooler halts.
SP-SKIP	Sets number of form feeds.
SP-STATUS	Displays status of spooler form queues.
SP-STOP	Halts spooler after current print job.
SP-TAPEOUT	Prints tape file on line printer.
SUBD, SUBX	Subtracts decimal or hexadecimal numbers.
T-ATT	Attaches magnetic tape unit.
T-BCK	Backspace tape.
T-CHK	Checks tape for parity errors.
T-DET	Detaches magnetic tape unit.
T-FWD	Forward-spaces tape.
T-RDLBL	Reads tape label.
T-READ	Reads one record from tape.
T-REW	Rewinds magnetic tape.
T-WEOF	Writes EOF on tape.
TERM	Sets terminal characteristics.
TIME	Prints time and date.
USER	Displays privilege level and workspace assignment.
WHAT	Displays current system parameters.
WHERE	Shows user points of execution.
WHO	Prints the line number and account name to which the terminal is logged on.
XTD	Converts hexadecimal number to decimal.

Figure A: Some TCL-I Verbs

PROGRAM INTERRUPTION (DEBUG FACILITY)

Processing can be interrupted by pressing <c>BREAK or F8 on the terminal. This causes an interrupt in current processing and an entry into the DEBUG state. However, such an interruption is inhibited during critical stages of processing.

When the break function key has been pressed and the DEBUG state has been entered, the following message is displayed:

```
I x.d  
!
```

where "x" and "d" describe the software location of the interruption. The DEBUG prompt character (!) prompts you for a DEBUG command. The commands listed in Figure A are the only DEBUG commands allowed (except for users with access to the assembly language).

Figure B illustrates a sample terminal interaction with the DEBUG facilities.

Note that pressing the break function key while in the terminal input or output mode causes a loss of an entire line of characters. If in the input mode, the retype-line character (<c>R) should be used to check the loss of data after returning from DEBUG via the 'G' command.

Upon encountering one of the hardware abnormal conditions, the system automatically traps to the DEBUG state with a message indicating the nature and location of the abort. If you have system privileges level zero or one, you must type "END" or "OFF" to exit from the DEBUG state.

When either "END" or "OFF" is entered while in the DEBUG state, the locks set by that process are unlocked. These locks include execution, group and item locks.

<u>COMMAND</u>	<u>DESCRIPTION</u>
P <-----	Print on/off. Each entry of a "P" command switches (toggles) from print suppression to print nonsuppression. The message OFF is displayed if output is currently suppressed. The message ON is displayed if output is resumed. This feature is useful when debugging.
H <-----	Toggles terminal echo off and on.
G or <lf> <-----	Causes resumption of process execution from the point of interruption. "G" cannot be used if a process ABORT condition caused the entry to DEBUG.
END <cr> -----	Terminates current process and causes an immediate return to TCL.
END <lf> <-----	Terminates current processing, but if a DATA/BASIC program was executing, control will pass to next statement in a PROC, if executing within a PROC.
OFF <-----	Terminates current process and causes you to be logged off the system.

Figure A: Selected DEBUG Commands

:LIST STAFF CITY HDR-SUPP <cr>	ENGLISH LIST statement
STAFF..... CITY.....	
1432 DALLAS	
2391 PHILADELPHIA <-----	<BREAK> key pressed.
I 6.1A3 <-----	Interrupts message.
!P <cr> OFF <-----	Turns Print off.
!G <cr> <-----	Resume execution without printing.
I 3.FB <-----	<BREAK> key pressed, interrupts message.
!P <cr> ON <-----	Turns Print back on.
!G <cr> <-----	Resumes execution with printing.
2951 LOS ANGELES)	
3051 NEW YORK)	Listing output resumes.
4260 CHICAGO)	
5913 MIAMI <-----	<BREAK> key pressed.
I 6137 <-----	Interrupts message.
!END <cr> <-----	Terminates LIST execution.
: <-----	TCL prompt.

Figure B: Sample Use of DEBUG Facilities

5. LOGON/LOGOFF

=====

LOGGING ON AND OFF THE SYSTEM

The logon processor initiates your session by identifying valid users and their associated passwords. The logoff processor terminates the session and should always be evoked via the verb OFF. These processors can accumulate accounting statistics for billing purposes and also associate you with your privileges and security codes.

LOGGING ON TO THE SYSTEM

You may log on to the system when the following (or a user-created) message is displayed:

LOGON PLEASE:

You then enter the user-id of the account established for you in the System Dictionary. If a password has also been established, you may follow your identification with a comma, and then the password; otherwise, if the specified account requires a password, the system will display the message:

PASSWORD: Entered password is not echoed.

The "hushed" area allows you to enter your password and not have it observable. The system validates your identification against the entries in the System Dictionary; if it is illegal, the following message is returned:

USER-ID?

LOGON PLEASE:

You must then re-enter your identification and password. If your identification is valid, but the password is not acceptable, the following message is displayed:

PASSWORD?

LOGON PLEASE:

You must then re-enter your identification and password. If you have successfully logged on to the system (i.e. both the identification and the password have been accepted), the following message (#335) is displayed:

```
*** WELCOME TO etc. ***
*** time RELEASE x.y date ***
```

PROPRIETARY/COPYRIGHT MESSAGE

LOGON/LOGOFF

where "time" is the current time, "date" is the current date, and "x.y" is the current release level. Figure A illustrates a sample logon interaction where the valid identification is TEST and the valid password is XYZ.

LOGGING OFF THE SYSTEM

Logoff is achieved by entering the word OFF, either at the TCL level or at the DEBUG level. A message indicating the connect time (i.e., number of minutes you were logged on) and the appropriate charge units will be displayed. The system then displays the LOGON PLEASE message and waits for the next session to be initiated. The general form of the logoff message is:

```
*****  
***      CONNECT TIME AT 14:38:19 = 1533 MINUTES      ***  
***      CPU MS. = 253          DISC I/O = 443        ***  
***      NUMBER OF ACTIVATIONS = 785                ***  
*****
```

Figure A illustrates sample logon interaction. Figure B illustrates a sample logoff interaction.

```

LOGON PLEASE:TEXT <cr>

USER-ID?

LOGON PLEASE:TEST <cr>
PASSWORD:--- <cr>

*** WELCOME TO MCDONNELL DOUGLAS M6000 ***
*** 14:07:00 RELEASE x.y 06 JAN 1986 ***
*** Insert the name of your system here ***

PROPRIETARY/COPYRIGHT MESSAGE

```

Figure A: Sample Logon Interaction

```

:OFF <cr>

*****
*** CONNECT TIME AT 14:38:19 = 1533 MINUTES ***
*** CPU MS. = 253 DISC I/O = 443 ***
*** NUMBER OF ACTIVATIONS = 785 ***
*****
*** LOGGED OFF AT 15:50:33 ON 06 JAN 1986 ***

LOGON PLEASE:

```

Figure B: Sample Logoff Interaction

SYSTEM MESSAGE FORMATS: LOGON MESSAGE

 Messages in the ERRMSG file have a defined format. The item 'LOGON' in the SYSTEM Dictionary is a prompt message for logging on to the system.

The ERRMSG file contains items which format system messages. This file is (improperly) called the Error Message File. System software uses items in this file to print most of the messages associated with the various processors (the Appendix SYSTEM ERROR MESSAGES lists these messages). The following codes may be present at the first character of attributes within these items:

A{(n)}	Parameter insertion code. The next parameter passed to the system message print routine is inserted at this point in the message. If "n" is specified, the parameter is left-justified in a field of "n" blanks
C	Clear screen.
D	Adds system date in the format DD MMM YYYY.
E{char-string}	The message item-id, surrounded by brackets, is printed followed by the optional character string.
H{char-string}	The character string is printed. If the string is to be a prompt, a plus sign (+) may be added to hold the cursor on the current line.
L{(n)}	The specified number of line feeds are output (one if "n" not specified).
R{(n)}	As in "A" above, except the parameter is right-justified in a field of "n" blanks.
S{(n)}	The specified number of spaces (counting from the beginning of the line, not from current position) is skipped. If the end of the line is reached, the line is printed and a new line is started.
T	Adds system time in the format HH:MM:SS.
X	Skips the next parameter.

The item called 'LOGON' in the System Dictionary follows this format. This message is the prompt invitation to log on to the system. Figure A displays a sample logon item in the System Dictionary. Figure B lists error message 335; you see this message immediately after logging on to the system.

System error messages may be recorded, for use by a DATA/BASIC program, within a POINTER-FILE item with item-id: account-name*L*%port#%

CAUTION: Items in the System Dictionary must not be altered while other users are logged on because the position of items might change, invalidating critical pointers.

```

LOGON
01 L(2)
02 T
003 S(21)
004 D
005 L
006 HLOGON TO THE DEVELOPMENT SYSTEM:+

```

Figure A: Sample Alternate Logon Message in System Dictionary

```

335
001 L
002 H*****
003 L2
004 H    ***                Welcome to MCDONNELL DOUGLAS M6000                ***
005 L
006 H***
007 S(18)
008 T
009 S(30)
010 HRelease x.y
011 S(48)
012 D
013 S(74)
014 H***
015 L
016 H***                Insert the name of your system here
017 L
018 H***                ***
019 L
020 H*-----*
021 L
022 H* THIS SOFTWARE, DATA, DOCUMENTATION OR FIRMWARE RELATED THERETO, AND
    etc.
046 H* (COPYRIGHT) 1986 AN UNPUBLISHED WORK BY MCDONNELL DOUGLAS COMPUTER
047 L
048 H* SYSTEMS COMPANY. - ALL RIGHTS RESERVED.
049 L
050 H*****

```

Figure B: Listing of ERRMSG 335

LOGON PROC AND GENERAL SYSTEM MESSAGES

| Upon logon, the system allows for the execution of a PROC with an item-id |
| identical to your identification. The system also allows a general |
message to be sent to all users as they log on to the system.

THE LOGON PROC

When you have logged on to your account, the system permits the automatic execution of a PROC whose item-id is the same as your identification. That is, the Master Dictionary of the account is checked for a PROC matching the identification which was used to log onto the account; if found, it is executed. (For information regarding PROCs, refer to the PROC Reference Manual.)

Typically, the Logon PROC performs standard functions always associated with your particular needs. For example, assume that the PROC listed in Figure A (which includes a TERM operation) is stored as item TEST in your Master Dictionary (M/DICT). If your identification is the word TEST, then the TEST PROC is executed automatically every time you log on (i.e., your particular terminal characteristics are automatically set). This is illustrated in Figure B.

GENERAL SYSTEM MESSAGE

The system allows you to send the same message to all users as they log on. The item 'LOGON' in the ERRMSG file may define such a message which is typically used to transmit information pertaining to system up-time, and the like. It should be noted that the LOGON message item must be present in the ERRMSG file even if no general system message is to be sent; in this case, the item should have no attribute values (i.e., an item-id only). The format of the LOGON message item is the same as any other message item in the ERRMSG file (see the topic SYSTEM MESSAGE FORMATS in this section).

```
-----  
Item 'TEST' in M/DICT  
001 PQN  
002 HTERM 118,44,7,6  
003 P  
004 X*** TERMINAL CHARACTERISTICS SET ***  
-----
```

Figure A: Sample Logon PROC

```
-----  
LOGON PLEASE:TEST,XYZ <cr> <----- Logon sequence.  
*** WELCOME TO MCDONNELL DOUGLAS M6000 ***  
*** 17:09:50 RELEASE x.y 06 JAN 1986 ***  
*** TERMINAL CHARACTERISTIC SET *** <----- Message from TEST PROC.  
: <----- TCL prompt character.  
-----
```

Figure B: Automatic Execution of Sample PROC from Figure A

COMPUTER USAGE: CHARGES AND CHARGE-TO VERBS

The CHARGES verb prints the current computer usage since logon. The CHARGE-TO verb charges the current computer usage since logon to a specified project or account.

CHARGES VERB

The CHARGES verb displays the current computer usage since logon as connect time in minutes, CPU milliseconds, the number of disk reads, and the number of process activations (number of timeslices). The general form of the verb is:

CHARGES

CHARGE-TO VERB

The CHARGE-TO verb keeps track of computer usage for several projects associated with the same LOGON name. The general form of the verb is:

CHARGE-TO project-name

This verb performs the following functions:

1. Terminates the current charge session by updating the Accounting History File with the appropriate connect time in minutes, CPU milliseconds, and the number of disk reads.
2. Creates a new item in the Accounting History File to which computer usage is subsequently logged, with item-id comprising: the account name, an asterisk, the project-name, a '#' sign, then the process number to which the user is currently logged-on.

For example, if you are currently logged onto SYSPROG at port 7, and you type in:

CHARGE-TO PROJECT1 <cr>

the corresponding item-id in the Accounting History File is changed to "SYSPROG*PROJECT1#7".

Figure A shows the general form of the CHARGES and CHARGE-TO verbs. Figure B shows examples of these verbs.

```

-----
CHARGES <----- Displays current computer usage.
CHARGE-TO project-name <----- Charges current computer usage to
                               a project.
-----

```

Figure A: General Form of CHARGES and CHARGE-TO Verbs

```

-----
:CHARGES <cr>

*****
***      CONNECT TIME AT 14:38:19 = 33 MINUTES      ***
***      CPU MS. = 253          DISC I/O = 123        ***
***      NUMBER OF ACTIVATIONS = 173                ***
*****

:

:CHARGE-TO PROJECT1 <cr>

*****
***      CONNECT TIME AT 14:39:41 = 34 MINUTES      ***
***      CPU MS. = 261          DISC I/O = 449        ***
***      NUMBER OF ACTIVATIONS = 197                ***
*****

:
-----

```

Figure B: Examples of CHARGES and CHARGE-TO Verbs

LOGTO VERB

| The LOGTO verb allows you to log to another account without first having |
to log off. The work space assigned to the original account is retained.

The general form of the verb is:

LOGTO account-name{,password}

where "account-name" is the account to log to. After entering the LOGTO command, the following occurs:

1. The system checks to see if the specified account is appropriately defined in the System Dictionary.
2. If you did not enter a password and one is required, you are then prompted for the password.
3. The accounting statistics are updated if being kept for the present account.
4. The security codes and system privilege level are changed to those of the new account.
5. The work space assignment of the original account is retained.
6. The spooler assignment and device attachment are retained.
7. If found, the Logon PROC is executed.

The LOGTO verb must not be executed from a PROC. Figure A shows the form of the LOGTO verb. Figure B shows an example of the LOGTO verb.

```
LOGTO account-name{,password}
```

Figure A: General Form of LOGTO Verb

```
:LOGTO SYSPROG,PWD <cr>
```

```
*****  
***          CONNECT TIME AT 14:38:19 = 1533 MINUTES      ***  
***  CPU MS. = 253          DISC I/O = 443              ***  
***          NUMBER OF ACTIVATIONS = 785                ***  
*****
```

```
:
```

Figure B: Sample usage of LOGTO Verb

USER IDENTIFICATION ITEMS

Each user has a user identification item stored in the System Dictionary. These items, which are either file definition items or file synonym definition items, define users who can log on to the System.

Account definition items are initially created via the CREATE-ACCOUNT PROC (see the section on SYSTEM MAINTENANCE PROCEDURES). These items may subsequently be updated via the EDITOR (see the EDITOR Reference Manual).

Account synonym definition items may be created by editing the System Dictionary to create a Q-pointer (with Q as attribute 1, original account name as attribute 2, attributes 3 and 4 null; remaining attributes as described in next topic). Account synonym definition items with null values for attributes 5 to 9 inclusive take zero parameters, i.e. no lock-codes, no password, SYS0 privileges with no workspace and no special characteristics.

Entries in the System Dictionary should only be updated (from the SYSPROG account) when no other users are logged on to the system. This is because the system software maintains pointers to data in the System Dictionary when users log on, and updating the System Dictionary will invalidate the pointers. An exception to this rule is when creating a new account, or a synonym to an existing account, which can be done at any time since new items are added to the end of the existing System Dictionary data, and thus do not disturb any existing items.

The next topic describes attributes 5 to 9 of user identification items.

Figures A and B illustrate typical items in the System Dictionary corresponding to an original account definition item MORRIS and an account synonym definition item MORRIS-II respectively.

Item MORRIS in SYSTEM

```

001 DX <----- D/CODE
002 205647 <----- Base FID
003 29 <----- Modulo
004 1 <----- Separation
005
006
007 PASSWRD <----- Password
008 SYS1(50) <----- Sys.Priv.Level (Add. W/S)
009 L
010 10
011 1

```

Figure A: User Identification Item For User MORRIS

Item MORRIS-II in SYSTEM

```

001 Q <----- D/CODE (Q indicates synonym
                                account)
002 MORRIS <----- Original account name

```

Figure B: User Identification Item For User MORRIS-II

USER IDENTIFICATION ITEMS: ATTRIBUTES 5 TO 9

Attributes 5 to 9 of a user identification item are used to give the corresponding user facilities and restrictions as explained below.

Attributes 5 through 9 of a user identification item contain data associated with a user's security (lock) codes, password, privileges and special characteristics:

- Attribute 5: Contains the set of retrieval lock-codes associated with a user. Multiple values (separated by value marks) are allowable. There is no restriction on the format of individual lock-codes. This attribute may be null, indicating no lock-codes. (See also the topic SECURITY.)
- Attribute 6: Contains the set of update lock-codes associated with a user (see retrieval lock-codes above).
- Attribute 7: Contains a user's password, which is a single value. This attribute may be null. There is no restriction as to the format of the password.
- Attribute 8: Contains a code which indicates the level of "system privileges" and work space assigned to a user. Three levels of system privileges are available: they are referred to as zero (lowest), one, and two (highest). Lower levels of system privileges restrict usage of certain facilities of the system, as listed in Figure B. SYS0 designates Level 0, SYS1 designates Level 1 and SYS2 designates Level 2. The system privilege code may be immediately followed by an "additional work space" parameter enclosed in parentheses (see the topic, WORK SPACE ASSIGNMENT in this section).
- Attribute 9: May contain one or more of the following codes:
- L - default on setting up new account (has no effect).
 - U - indicates that logon/logoff times should be collected in ACC file by the system.
 - R - on BREAK/END sequence execute logon PROC.
 - B - BREAK function remains disabled after logon completed.
 - T - on return to TCL from non-SELECT process a PROC is executed (if it exists) determined as follows: attribute of item ':portnumber' in DICT ACC is used as item-id of MD item, which is executed if it is a PROC.

Note: The PROC should not try to exit to TCL, as this will cause it to be re-executed indefinitely. Exit should be to, for instance, a user-defined program or to TCL+.

Attributes 1 through 4 and Attribute 13 are as defined for regular file definition items (see the topic FILE DEFINITION ITEMS in Section 3). Figure A shows a sample user identification item (for user XYZ).

<u>Item XYZ in System Dictionary</u>	
001 D <-----	D/CODE
002 2537 <-----	Base FID
003 13 <-----	Modulo
004 1 <-----	Separation
005 ABC <-----	Retrieval Lock Code (L/RET)
006 1234 <-----	Update Lock Code (L/UPD)
007 PW5 <-----	Password
008 SYS2(25) <-----	Sys. Priv. Level (Add. Workspace)
009 U <-----	Update ACCOUNT File for this user

Figure A: Sample User Identification Item (For User XYZ)

<u>FACILITY</u>	<u>MINIMUM PRIVILEGE LEVEL REQUIRED</u>
Updating of M/DICT and use of magnetic tape.	One
Use of DEBUG (other than P, OFF, END and G commands), use of DUMP processor, use of assembler and loader, use of FILE-SAVE and FILE-RESTORE processors and use of various restricted commands.	Two

Figure B: Required System Privilege Levels

WORK SPACE ASSIGNMENT

 | Additional work space may be specified immediately following the system |
privilege code in Attribute 8 of the user identification item.

There are three "linked" work spaces symbolically named HS, IS and OS. These work spaces are set to an initial size of six frames each (3000 bytes) at logon time (these frames are part of the primary workspace allocated to each process). Some users require additional work space to run DATA/BASIC programs and manipulate larger items; this requirement may be specified in Attribute 8 of the user identification item, immediately following the system privilege code.

The format of the "additional work space" parameter is:

(n)

where "n" is the additional work space requirement (in number of frames) for each of the three linked work spaces (HS, IS and OS). The "n" parameter must be a decimal number between one and 127 inclusive. For example, if the requirement for line 8 is 40 frames, $40*3=120$ frames must be available as a contiguous block.

Additional work space is obtained from the pool of contiguous overflow space. The first priority is to obtain this work space from low disk space (see topic LOW DISK SPACE AND FILE AREA in Section 2). If no space is available, the following message is displayed immediately after the "WELCOME TO M6000" message:

[334] n FRAMES OF ADDITONAL LOGON WORKSPACE ARE NOT AVAILABLE.

where "n" is the additional work space parameter in Attribute 8 of the user identification item.

If the format of the work space parameter is illegal, then the following message is returned immediately after the "WELCOME TO M6000" message:

[333] THE FORMAT IF THE ADDITIONAL WORK-SPACE PARAMETER: parameter IS
 ILLEGAL FOR THIS ACCOUNT NAME.
 ADDITIONAL WORK-SPACE HAS NOT BEEN ASSIGNED.

Consider the sample situation where the entry in the eighth attribute of a user identification item is SYS1(10). This gives the user level one system privileges and 10 additional frames of work space per linked set. Thus, 30 additional frames have to be obtained from contiguous overflow space. This situation is illustrated in Figure A. It is assumed that the user logs in on process one, which corresponds to a PCB FID of X'640' (1600). The arrows in the figure represent frame linkages set up after logon.

<u>WORK SPACE NAME</u>	<u>FID(s)</u>	<u>WORK SPACE</u>
PCB	1600	
SCB	1601	
TCB	1602	
QCB	1603	
BMS, etc.	1604	
TS	1605	
PROC work area	1606-1609	
HS	1610-1625	Original HS
IS	1626-1631	Original IS
OS	1632-1637	Original OS
HS extension	9024-9033	Additional HS
IS extension	9034-9043	Additional IS
OS extension	9044-9053	Additional OS

Figure A: Sample Additional Work Space Assignment

ALLOCATING PERMANENT BLOCKS OF ADDITIONAL WORK SPACE

You may allocate a permanent block of additional work space via the WORKSPACE verb. The Logon Work Space Table (LWST) keeps track of each "block" of space by listing the starting and ending frame id, the size of the block, its status, whether the block is linked, and the port currently using the block (if any).

THE WORKSPACE VERB

This verb allocates and deallocates permanent blocks of additional work space. Its general format is:

WORKSPACE {size} {(D) or (quantity) or (D,quantity)} {(P)}

where:

- size allocates one block of space three times 'size' from disk lowspace. This number corresponds to the number of additional frames requested on Line 8 of the account definition item in the System Dictionary.
- (D) deallocates one block of space and returns it to overflow.
- (quantity) specifies the number of ports that will be allocated (or deallocated) the specified block of space.

Figure B shows several examples of how the WORKSPACE verb is used.

THE LOGON WORK SPACE TABLE (LWST)

This table defines the permanently allocated blocks of work space. Logged in this table are the starting and ending frame ids of each block and the status of that block. The integrity of the block links is guaranteed because this table is written back to disk every time the table is updated (i.e, this space is not lost following a coldstart).

The LWST provides the added benefit of fast logon. Blocks of space are taken from available space and linked the first time they are used, rather than every time a terminal is logged on. Therefore, when a terminal is logged on, the additional work space frames do not have to be read and linked.

WORKSPACE {size} {(D) or (quantity) or (D,quantity)} {(P)}

Figure A: General Format of the WORKSPACE Verb

WORKSPACE 66	Permanently allocates one block of 198 frames (66 x 3) from Low Disk Space. The number 66 corresponds to the number found in Line Eight of the account definition item in the SYSTEM Dictionary. If a temporary block of 198 frames is found before a null LWST entry is encountered, it is converted to a permanent block.
WORKSPACE 33 (10)	Allocates 10 blocks, each of which is 99 frames.
WORKSPACE (D)	Deallocates one block of space and returns it to overflow. If the first permanent block found is "IN USE", it will be converted to temporary space and returned to overflow when the line using it logs off.
WORKSPACE (D,5)	Deallocates 5 blocks of space.
WORKSPACE 66 (D)	Deallocates a block of space that contains 198 frames. This command has no effect if a block of 198 frames cannot be found.
WORKSPACE 33 (D,5)	Deallocates 5 blocks of space that each contain 98 frames.

Figure B: Sample Usage of the WORKSPACE Verb

PERMANENT BLOCKS

When you log on, the system first attempts to use a block of space permanently allocated with the WORKSPACE verb. The LWST is searched for the smallest block of space not "IN USE" by another terminal that satisfies the size requested by Line Eight of the account definition item in the System Dictionary. If one is found, it is marked "IN USE" and linked to the terminal's primary work space.

Permanent blocks of space are not linked until the first time they are used. Therefore, the first time you log on after a coldstart or file restore, there is some delay while these frames are contiguously linked. Subsequently, they will already be linked and you will be immediately logged on. Permanent blocks of space may also be marked "RELINK" when you log off from the assembly or DATA/BASIC debugger or when the 'L' option is used with the OFF command. If you are concerned about work space links, press the <c>BREAK or equivalent key then enter OFF. Alternatively, enter "OFF (L)" at TCL. Subsequent logon relinks your work space.

TEMPORARY BLOCKS

When a suitable block of permanently allocated space is not available, a temporary block of overflow space is used. Note that you will encounter a delay during logon while the temporary space is contiguously linked. When you log off, the temporary block is returned to overflow. Coldstarting the system while assigned to a temporary block causes the system to lose that block of space.

To view the current LWST table, execute the WORKSPACE verb without any parameters or options (see Figure C).

The SYSTEM-SETUP PROC, as supplied, reserves a maximum size block (127) for each terminal on the system. You should modify this PROC to accommodate your system needs.

```
-----  
:WORKSPACE {(P)} <cr>  
  
      FROM      TO      SIZE/3  STATUS  LINKED  PORT  
      7307 -    7396      30      T      Y       3  IN USE  
      8409 -    8468      20      Y      Y  
      9176 -    9220      15      Y      Y  
      9300 -    9680     127      Y      Y       12  IN USE  
      9681 -   10061     127  
  
Note: The "T" listed under STATUS means that block of space is temporary  
and will be returned to overflow when the user logs off.  
-----
```

Figure C: Sample LOGON Work Space Table (LWST)

SECURITY

Security codes may optionally be placed in the L/RET and L/UPD attributes of a file definition item to restrict access and update. At logon time, you are assigned the set of security codes which are in your user identification item. During the session, whenever an L/RET or L/UPD code is encountered, a search is made of the assigned codes for a match. If no match is found, you are denied access. A code may consist of any combination of ASCII characters.

L/RET AND L/UPD

File definition ("D" code) items have L/RET (retrieval lock) and L/UPD (update lock) attributes (attributes 5 and 6 respectively). When these attributes are non-null, they are known as security codes (if multivalued, only the first value is significant). Since each file may be individually locked for both update and retrieval, your user identification item must contain the same security codes which protect the data you wish to access. Using this feature, a complex "mask" can be constructed for each user, giving you a different subset of files which you may access.

Security at the file level is activated at the processor level. The following processors are assumed to be updating processors and, therefore, require a match on the L/UPD attribute in the file definition item: COPY, EDITOR, and the Assembler. Other processors, with the exception of DATA/BASIC, are assumed to be retrieval processors and require a match on the L/RET attribute in the file definition item. (See DATA/BASIC Reference Manual for an explanation of DATA/BASIC security features.) Failure to match one of the user security codes with either the L/RET or L/UPD attribute value generates the following message (and returns control to TCL):

[210] FILE 'file-name' IS ACCESS PROTECTED

*

USER-ASSIGNED CODES

Each user identification item in the System Dictionary (see the topic USER IDENTIFICATION ITEMS in this section) contains the list of security codes assigned to that particular user. The lock code in the file being verified must match.

Security codes may be assigned initially when an account is created via the CREATE-ACCOUNT PROC (refer to Section 10). Security codes may be added or deleted by updating the appropriate user identification item using the EDITOR (assuming one has the appropriate security codes); however, updates to the user identification item should only be performed when no one else is logged on to the system.

Care must be taken when securing the System Dictionary and the SYSPROG account. SYSPROG (and any synonyms, such as SP) should be protected first. Then SYSTEM may be secured, usually with the same codes as SYSPROG. Remember that there are two items in SYSTEM that should be secured with the same codes; their item-ids are SYSTEM and DL/ID. If SYSTEM were to be secured before SYSPROG, then SYSPROG would not be able to access SYSTEM.

SECURITY CODE COMPARISON

Security codes are verified by comparing the value in the file dictionary against the corresponding string of values in the user identification item. Characters are compared from left to right. An equal (verified) compare occurs when the value in the file dictionary is exhausted and all characters match up to that point. This is illustrated in Figure A.

When referencing a file using a Q synonym (refer to Section 3), a security code match is made at all levels (i.e., SYSTEM, M/DICT, and file dictionary). Therefore, a correspondence must be maintained at all levels to process the Q synonym files. Because the user identification item for the account containing the primary file is verified for security codes, the user referencing the Q synonym must have a code defined in this user identification item which will verify with the first code in the equated account's user identification item. Thus, in a user identification item, only the first code is used to protect the account from Q synonym accesses, while all the codes in the item are assigned to you when you log on.

<u>FILE DICTIONARY CODE</u>	<u>USER IDENTIFICATION CODE</u>	<u>RESULT</u>
123	123	MATCH
12	123	MATCH
123	12	NO MATCH
XYZ	XYZ5	MATCH
AQ2	AQ	NO MATCH

Figure A: Sample Security Code Comparisons

ACCOUNTING HISTORY FILE: INTRODUCTION

The Accounting History File is one of the mandatory files in the system. This file contains accounting history for the system, as well as entries that describe currently active (logged-on) users.

The System Dictionary must contain the file definition item ACC and the file synonym definition item ACCOUNT for the Accounting History File, as illustrated in Figure A. The ACC dictionary is set up for examining and listing the data in Accounting History File via ENGLISH (see the topic titled ACCOUNTING HISTORY FILE: SUMMARY AND EXAMPLES in this section). There are two types of entries (items) in the Accounting History File: those that represent active (logged-on) users, and those that keep track of accounting history.

ACTIVE USER ITEMS

The item-id of an active user item in the Accounting History File is the six-character hexadecimal FID of the PCB of the user's process, right justified with leading zeros as necessary. If the PCBs start at FID 1536, (they proceed in steps of 64 frames from there on), a user logged on to process zero will have an entry with an item-id "000600" (1536), and so on.

Attribute 1 of an active user item contains the name of the user (i.e., the item-id of the user identification item), Attribute 2 the date logged on, and Attribute 3 the time logged on. Active user items are created when a user logs on, and deleted when he logs off. (Attributes of an active user item are summarized in the topic titled ACCOUNTING HISTORY FILE: SUMMARY AND EXAMPLES in this section.)

ACCOUNTING HISTORY ITEMS

The item-id of an accounting history item is the name of the user (i.e., the item-id of the user identification item), followed by '#', followed by the process number at which that user has logged-on (process number is in decimal without leading zeros). Attributes 1, 2 and 3 are not used. The remainder of the attributes are described below:

- Attribute 4: Date(s) logged on. Each unique date is stored. Value marks are tagged on to the value in this attribute if multiple logoffs occur on the same date (for LIST alignment purposes). Date is stored in system date format.
- Attribute 5: Time(s) logged on. An entry is made for each logoff, representing the time at which you logged on. Time is represented in seconds past midnight (24-hour clock).
- Attribute 6: Connect time(s). This entry represents the time in seconds between the logon and logoff.
- Attribute 7: Charge-units. A number representing the CPU usage (CPU milliseconds) is added for each logoff.
- Attribute 8: Disk-reads. This entry reflects the number of frame faults encountered between logon and logoff.

Your System Dictionary item must include a 'U' in Attribute Nine if you are to have your account file history items updated. The entries in the account file contain the history of each session (logon to logoff). If the System Dictionary data has been changed since logon or the history item to be updated is too large for your work space, the following message is printed:

- [338] ACCOUNT FILE STATISTICS WERE NOT UPDATED DUE TO EITHER:
 1. INSUFFICIENT WORKSPACE TO CONTAIN THE ACCOUNT FILE ITEM, OR
 2. SYSTEM DICTIONARY CHANGED WHILE YOU WERE LOGGED ON

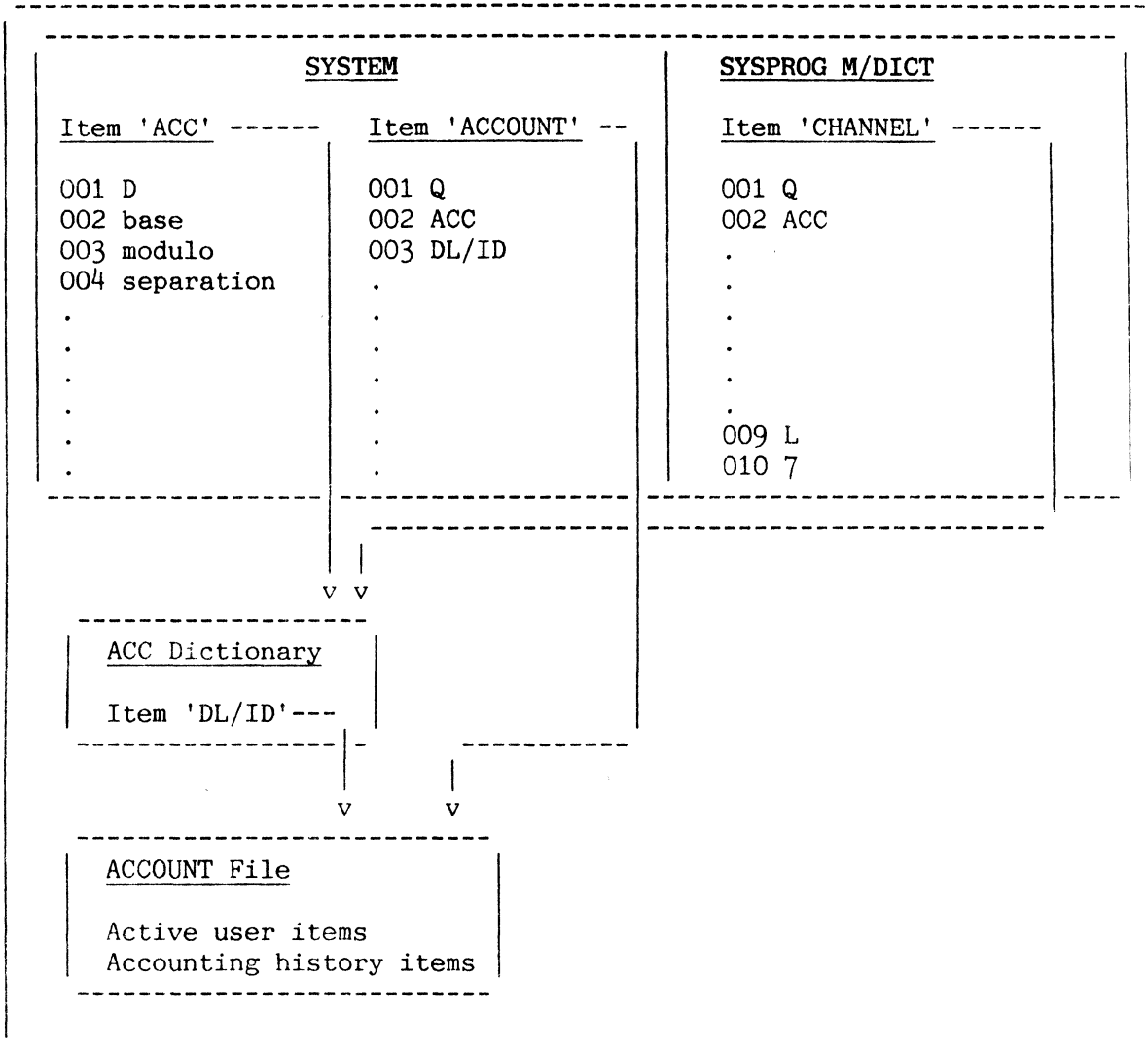


Figure A: Accounting History File Hierarchy

ACCOUNTING HISTORY FILE: SUMMARY AND EXAMPLES

 This topic summarizes formats of the active user items and accounting history items in the Accounting History File. Also presented are sample entries for the Accounting History File.

Figure A summarizes attributes for the active user items and accounting history items. Figure B shows a sample sorted listing of the active users (users with a value for Attribute A1) via an ENGLISH SORT statement. Figure C shows a sample listing of the accounting history item for user FRANK on process 9 via an ENGLISH LIST statement. This same item is "dumped" in Figure D to show its internal storage format.

<u>ATTRIBUTE NUMBER</u>	<u>'ACC' DICTIONARY NAME</u>	<u>ACTIVE USER ITEM</u>	<u>ACCOUNTING HISTORY ITEM</u>
0	(Item-id)	Six-character hexadecimal PCB-FID	User#line
1	A1	User name	Not used
2	A2	Date logged on	Not used
3	A3	Time logged on	Not used
4*	DATE		Dates logged on
5*	TIME		Times logged on
6*	CONN		Connect times (seconds)
7*	UNITS		Charge units (CPU milli-seconds)
8*	DISC-READS		Disc reads
	TOT-CONN		Total connect times
	TOT-UNITS		Total of CPU milliseconds
	TOT-DISC-READS		Total of disc-reads

* Multivalued attribute: one value for each LOGON/LOGOFF sequence

Figure A: Summary of Active User Items and Accounting History Items

```

:--SORT ACC WITH A1 A1 A2 A3 HDR-SUPP <cr>

ACC..... NAME.....DATE .....TIME.....
          LOGGED ON          LOGGED ON

->000640   SYSPROG          10 FEB 86          09:15:16
000680    SP              10 FEB 86          09:53:16
000780    SP              10 FEB 86          09:33:36
000840    FRANK           10 FEB 86          08:38:43

4 ITEMS LISTED.
          |
          ----- date logged on
          ----- PCB-FID=000640
                   (process 1)
    
```

Figure B: Sorted Listing of Active Users

```

:--LIST ACC 'FRANK#9' HDR-SUPP <cr>

ACC.....DATE..... TIME.....CONNECT..CHRG DISC...TOTAL.....TOTAL..TOTAL
          LOGGED ON LOGGED ON TIME      UNITS READS  CONNECT  UNITS  DISC
                   TIME
FRANK#9  10 FEB 86 08:38:43 00:32:12  302  2107   02:01:57  1217  7189
                   09:11:51 00:05:00   58   307
                   09:17:03 00:32:33  321  2009
11 FEB 86 10:50:00 00:03:20   30   121
                   11:06:40 00:08:20   83   413
08 MAR 86 11:08:31 00:40:32  423  2232
    
```

Figure C: Listing of Accounting History Item for User FRANK on Process 9

```

:I-DUMP ACC 'FRANK#9' <cr>

          dates          times          attribute connect
          logged on      logged on      mark      times
          |              |              |      |
          v              v              v      v
FRANK#9^4059]]]4060]]]4061^31123]33111]33423]39000]40000]40111^1932]300]
1953]200]500]2432^302]48]321]30]83]423^2107]307]2009]121]413]2232
          |              |              |
          charge      disc-reads      value
          units
    
```

Figure D: Dump of Accounting History Item for User FRANK on Process 9

ACCOUNTING HISTORY FILE: PERIODIC CLEARING

| To avoid overflowing the accounting history item in the Accounting |
| History File for a specific user using a specific process, the item |
should be periodically cleared.

The point at which an accounting history item overflows depends, for smaller values of workspace, on the activity on the account and the amount of workspace allocated to the account. Accounts with greater than 58 additional workspace frames, however, will not overflow their associated accounting history item until it reaches the normal maximum item size of 32, 267 bytes.

Note: An account does not necessarily acquire just the additional workspace specified in attribute 8 or the corresponding user definition item at logon; if no permanent block of this is available, any available block, of the nearest size larger than this, is attached. The blocks of permanent logon workspace may be inspected using the WORKSPACE verb; the USER verb displays the actual additional workspace currently available on an account. Use the TCL statement 'CT SYSTEM user-name' to inspect line 8 for the additional workspace defined.

The overflow point is the smaller of the following number of bytes:

{(additional workspace + 6) * 500 - 100}

and

32,267

The size of the accounting history item for a user with 'user-name' at process 'process-number' may be determined by entering:

STAT ACC 'user-name#process-number' <cr>

which displays the following output:

STATISTICS OF ACC:
TOTAL = xxx AVERAGE = yyy COUNT = 1

where TOTAL is the total number of bytes in the item.

If this is anywhere near the overflow point, the item should be printed out and then cleared by entering:

:LISTACC user-name#process-number (P) <cr>

followed by:

:DELETE ACC user-name#process-number <cr>

Figure A illustrates these procedures for user MORRIS on process 9.

Note: Logging-off an account which would cause an accounting history item to overflow does not update the item, but displays one of the following messages:

```
'user-name' TRUNC.BASE:nnnn
```

or

```
[338] ACCOUNT FILE STATISTICS WERE NOT UPDATED DUE TO EITHER:
```

1. INSUFFICIENT WORKSPACE TO CONTAIN THE ACCOUNT FILE ITEM,
OR
2. SYSTEM DICTIONARY CHANGED WHILE YOU WERE LOGGED ON.

```
-----
:USER <cr>
0 SYSTEM PRIVILEGE LEVEL
50 ADDITIONAL FRAMES OF WORKSPACE

:STAT ACC 'MORRIS#9'

STATISTICS OF ACC:
TOTAL = 25031  AVERAGE = 25031.0  COUNT = 1

{(50+6) * 500 - 100} = 27,900 bytes

:LISTACC MORRIS#9 (P)<cr>

:DELETE ACC MORRIS#9 <cr>
MORRIS#9 deleted
Item (<cr> to exit) ? <cr>
Done

:
```

Figure A : Monitoring Accounting History Item (for user MORRIS at process 9)

6. FILE MANAGEMENT PROCESSORS

=====

OVERVIEW

| This section describes the data base management processors CREATE-FILE, |
CLEAR-FILE, DELETE-FILE and COPY.

The file management processors provide capabilities for creating, managing, and manipulating files and items within the system.

CREATE-FILE PROCESSOR

The CREATE-FILE processor generates new dictionaries and/or data files. The processor establishes file dictionaries by creating "D" entries in your Master Dictionary (M/DICT) and reserving and linking primary file space. You need only specify values for the desired modulo(s).

CLEAR-FILE PROCESSOR

The CLEAR-FILE processor eliminates the data from a file (i.e., it sets the file to the "empty" state by placing an attribute mark in the first data position of each group of the file). Any overflow frames linked to the primary file space are released to the system's overflow space pool. Either the data section or the dictionary section of a file may be cleared.

DELETE-FILE PROCESSOR

The DELETE-FILE processor deletes a file. Either the data section or both the dictionary and data sections (if present) of the file may be deleted. Frames are returned to available space.

COPY PROCESSOR

The COPY processor copies an entire file or selected items from the file to the terminal, to the printer, to another file or to items with different item-ids within the same file. It may also be used to create a new file.

FILE MANAGEMENT PROCESSORS

EXAMPLES

As a general introduction to this section, Figure A presents a number of examples illustrating the use of file management processors.

ADDITIONAL FILE MANAGEMENT PROCEDURES

Additional file management procedures (such as the creation of new user-accounts, saving and restoring of files, etc.) are detailed in Section 10, SYSTEM MAINTENANCE PROCEDURES.

<u>EXAMPLE</u>	<u>EXPLANATION</u>
:CREATE-FILE DICT TEST 1 <cr> <-----	Creates a file dictionary for the TEST file, with a modulo of 1 and a default separation of 1.
:CREATE-FILE DATA TEST 7,2 <cr> <-----	Reserves disk space for the data area of the TEST file, with a modulo of 7 and a separation of 2.
:CREATE-FILE FNA 1, 3,13 <cr> <-----	Creates a file dictionary for the FNA file, with a modulo of 1 and a separation of 1. Also creates the data area of the FNA file, with a modulo of 3 and a separation of 13.
:CLEAR-FILE DATA XYZ <cr> <-----	Clears data section of XYZ files.
:DELETE-FILE DICT INV <cr> <-----	Deletes dictionary section of INV file (data section must not exist).
:DELETE-FILE FAE <cr> <-----	Deletes data and dictionary sections of FAE file.
:COPY TEST I1 I2 I3 <cr> <----- TO:X1 X2 X3 <cr>	Copies data items I1 I2 and I3 back into the same file (TEST) giving duplicate items with item-ids of X1, X2 and X3.
:COPY DICT SAMPLE * <cr> <----- TO:(DICT FLAVORS) <cr>	Copies all dictionary items from file SAMPLE to dictionary of file FLAVORS.
:COPY TEST * <cr> <----- TO: <cr>	Copies all items in the TEST file to your terminal.

Figure A: Sample Usage of File Management Processors

FILE MANAGEMENT PROCESSORS

CREATING NEW FILES: CREATE-FILE PROCESSOR

The CREATE-FILE processor generates new files and dictionaries.

The CREATE-FILE processor establishes file dictionaries by creating "D" entries in your Master Dictionary (M/DICT). The CREATE-FILE processor is also used to reserve disk space for the data portion of the new file by automatically placing the DL/ID (Data-Level Identifier) entry in the file dictionary, which then points to the data area. CREATE-FILE will automatically locate and reserve a contiguous block of disk frames from the available space pool. You need only specify values for the modulo and (optionally) the separation of both the file dictionary and data area (Figure A). (For a discussion of the values to use for modulo and separation, refer to the topic SELECTING MODULO AND SEPARATION in this section.)

The most common form of the CREATE-FILE command, which creates both the dictionary and data areas at once, has the general form:

```
CREATE-FILE file-name m1{,s1} m2{,s2}
```

where "file-name" is the name of the file, "m1" and "s1" are the modulo and separation of the DICT (respectively), and "m2" and "s2" are the modulo and separation of the data area. The separation parameters each default to one if omitted. The name given to the new file must not already exist in the user's M/DICT. Figure B shows the creation of the dictionary and data sections of the file SAMPLE.

The dictionary portion (DICT) of the file must be created first if the above format is not used. The general form for creating a file dictionary is:

```
CREATE-FILE DICT file-name m,{s}
```

where "file-name" is the name of the file, "m" is the desired modulo, and "s" is the desired separation (which defaults to one if omitted). Again, the file-name must not already exist in the user's M/DICT. An example is shown in Figure C, where the DICT for a file named SAMPLE2 is being created.

Once the DICT has been created, the primary file space for the data area of the file can be reserved. The general form is:

```
CREATE-FILE DATA file-name m,{s}
```

where "file-name" is the name given to the file, "m" is the desired modulo, and "s" is the desired separation (which defaults to one if omitted). An example is shown in Figure D. (Note that a data area need not be reserved for a single-level file where data is to be stored in the DICT.)

The BREAK function is inhibited during a CREATE-FILE.

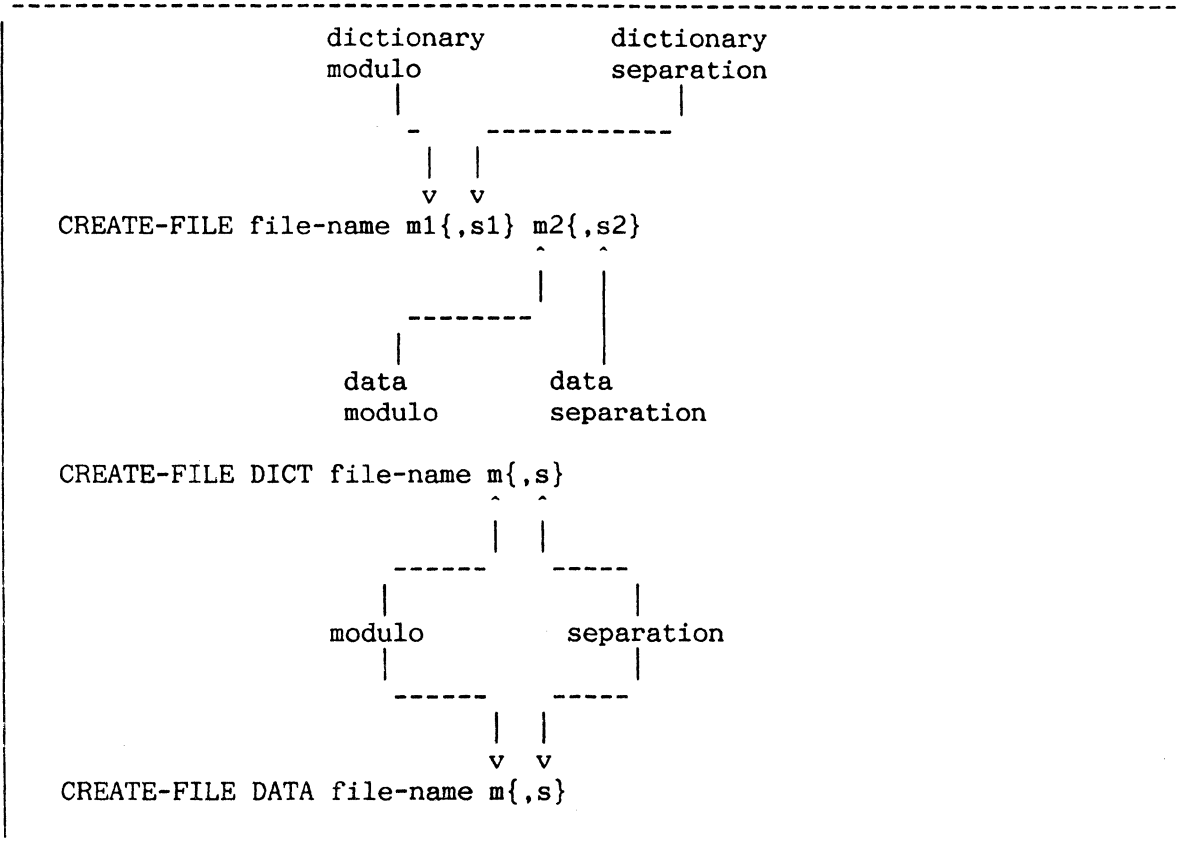


Figure A: General Forms of CREATE-FILE Command

```
:CREATE-FILE SAMPLE 1 11,7 <cr>

[417] FILE 'SAMPLE' CREATED; BASE = 34980, MODULO = 1, SEPAR = 1.
[417] FILE 'DL/ID' CREATED; BASE = 35132, MODULO = 11, SEPAR = 7.
```

Figure B: Sample Creation of Both Dictionary and Data Portions

```
:CREATE-FILE DICT SAMPLE2 1 <cr>

[417] FILE 'SAMPLE2' CREATED; BASE = 34946, MODULO = 1, SEPAR = 1.
```

Figure C: Sample Creation of Dictionary Portion

```
:CREATE-FILE DATA SAMPLE2 11,7 <cr>

[417] FILE 'DL/ID' CREATED; BASE = 34947, MODULO = 11, SEPAR = 7.
```

Figure D: Sample Creation of Data Portion

SELECTING MODULO AND SEPARATION

| Efficient file accessing and disk use depends on proper selection of
modulo and separation.

"Modulo" is the number of groups in a file; "separation" is the number of contiguous frames per group. When a file is created its modulo and separation parameters must be specified; the frames then allocated by the system (modulo*separation) are referred to as the "primary" file-space. As data is placed into the file, any group may overflow by attaching frames from the available system space pool; this space is referred to as the "overflow" file-space. To locate an item, given its item-id, the item-id is "hashed" using the modulo of the file, which results in a unique group number in which it may exist. The item-ids in that group are then linearly searched for the required item. A proper selection of the "modulo" parameter is essential to minimize this search time. The current maximum modulo and separation are 8,388,607 and 127 respectively.

In the current file structure, the SEPARATION parameter is of practically no value. The disk-head will almost definitely have moved between the moment that a process requests one frame of the group and the next; therefore, whether the next linked frame of the group is contiguous (that is, if the separation is one) or not makes a marginal difference. What this amounts to is that there is little difference in accessing a frame in "primary" file-space and in "overflow" space, particularly since the file-restore process "compacts" all files and leaves the "primary" and "overflow" frames close to each other.

Selecting a proper MODULO is extremely important, since the number of groups directly affects the search and update time for an item in the group. What is proposed here is a modulo selection process that would make the average GROUP length between one and two frames (i.e., at most two frame faults to find a particular item). Obviously, if the item-size is of the order of 250 bytes or greater, this rule of thumb must be modified.

The number of frame faults - the factor that causes the most degradation of overall system response - increases dramatically as the number of frames per group increases. This is because, on the average, half of the frames in a group have to be written back to the disk after an item update. Thus, to update an item in a group, we have to read every frame in the group and write half of them (see Figure A).

A good rule of thumb is to select the MODULO to be the smaller of:

- (1) $\frac{\text{The number of bytes of data}}{500}$ or (2) The number of items

and use a separation of one.

The DATA/BASIC program 'SET-ALLOCATION' in SYSPROG-PL uses these formulas to set the reallocation parameters (contained in attribute 13 of the file definition item) for existing files.

Using the table, one can select an appropriate ITEMS/GROUP value; knowing the expected number of items in the file then gives the approximate modulo. Modulo must not be a multiple of 2 or 5 since this can produce a very uneven distribution of items within groups, and thereby cause inefficiency in storage and access.

Type of Activity	Frames per Group					
	1	2	3	4	5	6
READ	1.0	1.5	2.0	2.5	3.0	3.5
UPDATE	3.0	4.5	6.0	7.5	9.0	10.5

Figure A: Disk Operations Versus Frames/Group

If Avg Item-Size Is	Then Item/Group Should Be	And Bytes/Group Will Be
20	22.0	440
35	13.0	455
50	9.0	450
75	12.0	900
100	9.0	900
125	7.5	937
150	6.0	900
175	8.0	1400
200	7.0	1400
250	5.8	1450
300	6.4	1920
350	5.5	1925
400	4.8	1920
500	3.8	1900
1000	3.0	3000
5000	0.8	4000

Figure B: Selecting Items/Group

Avg Item Size	Approximate No. of Items	Items/Group (From Figure B)	Approximate Modulo
20	800	/	= 36
40	5000	/	= 383
210	1800	/	= 257
4000	230	/	= 287

Figure C: Examples of Computing Modulo

FILE MANAGEMENT PROCESSORS

CLEARING AND DELETING FILES: CLEAR-FILE & DELETE-FILE PROCESSORS

| The CLEAR-FILE processor clears (i.e., empties) files. The DELETE-FILE |
processor deletes files.

The CLEAR-FILE processor clears the data from a file (i.e., it sets the file to the "empty" state by placing an attribute mark in the first data position of each group of the file). "Overflow" frames that may be linked to the primary file space will be released to the system's additional space pool. Either the data section or dictionary (DICT) section of a file may be cleared using the CLEAR-FILE verb. If the dictionary section is cleared, and a corresponding data section exists (as implied by the presence of the DL/ID item in the dictionary), then the DL/ID will be maintained in the dictionary. The BREAK function is inhibited during the DELETE process, but not the CLEAR process.

To clear the data section of a file, the following verb is used:

CLEAR-FILE DATA file-name

To clear the dictionary section of a file, the following verb is used:

CLEAR-FILE DICT file-name

The DELETE-FILE processor erases a file. Either the data section, dictionary section, or both of these file sections may be deleted. The dictionary section of a file which has a data section cannot be deleted until the data section is deleted. All frames no longer required are returned to the available space pool.

To delete the data section of a file, the following verb is used:

DELETE-FILE DATA file-name

which will delete the data section pointed to by the DL/ID and will also delete the DL/ID from the dictionary section.

To delete the dictionary section of a file, the following verb is used:

DELETE-FILE DICT file-name

To delete both the data and dictionary sections, the following verb is used:

DELETE-FILE file-name

In the latter two cases, the file-definition item in the M/DICT (the file-name) is also deleted. Files defined by file-synonym in the user's M/DICT cannot be specified in a DELETE-FILE verb. These definitions should be deleted with the DELETE PROC or the EDITOR.

CLEAR-FILE and DELETE-FILE should not be used on files containing group format errors. If the cause of the group format errors is due to incorrect frame linkages, use of these verbs could destroy data in other files. Either repair the error (refer to the topic GFE's: FIX-FILE-ERRORS VERB) or delete the file definition item in the M/DICT (all associated space will be lost until a restore is performed).

The CLEAR-FILE and DELETE-FILE verbs are summarized in Figures A and B respectively. Figure C presents examples.

```

-----
CLEAR-FILE DATA file-name <----- Clears data sections.
CLEAR-FILE DICT file-name <----- Clears dictionary section.
-----
    
```

Figure A: General Forms of CLEAR-FILE Verb

```

-----
DELETE-FILE DATA file-name <----- Deletes data section.
DELETE-FILE DICT file-name <----- Deletes dictionary section.
DELETE-FILE file-name <----- Deletes data and dictionary
                               section.
-----
    
```

Figure B: General Forms of DELETE-FILE Verb

<u>COMMAND</u>	<u>EXPLANATION</u>
:CLEAR-FILE DATA TESTFILE<cr> <-----	Clears data section of file TESTFILE.
:CLEAR-FILE DICT EMPL<cr> <-----	Clears dictionary section of file EMPL.
:DELETE-FILE DATA A/P<cr> <-----	Deletes data section of file A/P.
:DELETE-FILE DICT INV<cr> <-----	Deletes dictionary section of file INV.
:DELETE-FILE PAYROLL<cr> <-----	Deletes data and dictionary sections of file PAYROLL.

Figure C: Sample Usage of CLEAR-FILE and DELETE-FILE Verbs

FILE MANAGEMENT PROCESSORS

COPYING ITEMS: COPY PROCESSOR

The COPY processor copies items from a file to the terminal, printer, to another file (either in your account or in some other user-account), or to items with different item-ids within the same file.

The COPY processor is evoked via the COPY verb. The general form is:

```
COPY {DICT} file-name item-list* {(options)}
```

where:

file-name specifies the source file (the file to be copied).

item-list* consists of one or more item-IDs separated by blanks, or an asterisk (*) to specify all items. The item-list is omitted when the item-list is provided by an immediately-preceding SELECT, SSELECT, GET-LIST, FORM-LIST or equivalent statement (see the ENGLISH Reference Manual for details).

{(options)} any options used must be enclosed in parentheses. Options for COPY are described in the next topic.

When a valid COPY command has been entered, the COPY processor prompts:

TO:

If the COPY is to the same file, you must then enter the item-id(s) for the destination of the COPY operation. If the COPY is to another file, you must enter the destination file-name enclosed in parentheses, optionally followed by the destination item-ids.

The first COPY operation in Figure B illustrates the creation of a new item WORTH in DICT SAMPLE by copying the existing item COST from the same file. The second operation illustrates the creation of one data item (1242-99) from another (1242-01), again in the same file.

When using the COPY operation, multiple items may be specified as source and destination. Each item-id must be separated by a space and may be repeated within the item-list. There may be different numbers of items within the source and destination lists. If the source file item-list is exhausted first, the COPY terminates. If the destination file item-list is exhausted first, the remainder of the items are copied with no change in item-id. For instance, if the destination item-list is omitted the source items are copied to the destination file with the same item-ids.

The third example in Figure B copies data items RED, WHITE, and BLUE back into the same file (called FLAVORS) but gives them item-ids of ALPHA, BETA, and GAMMA.

(continued)

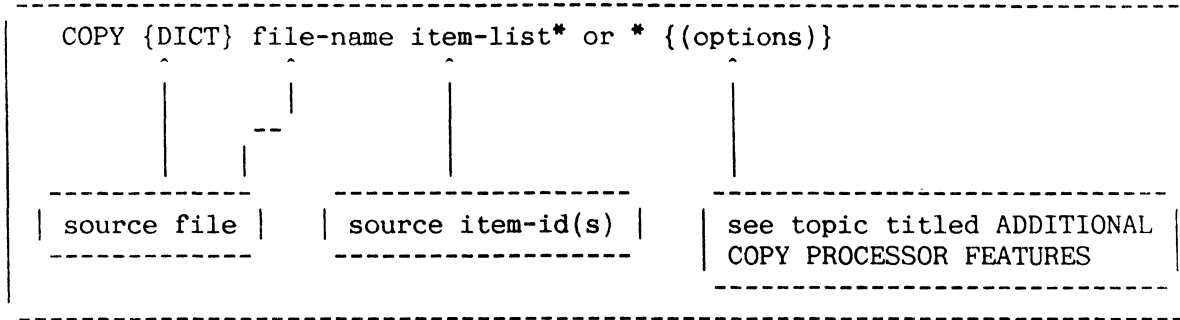


Figure A: General Form of COPY Verb

```

: COPY DICT SAMPLE COST <cr> <----- Single dictionary item copied.
TO: WORTH <cr>
1 ITEMS COPIED

: COPY SAMPLE 1242-01 <cr> <----- Single data item copied.
TO: 1242-99 <cr>
1 ITEMS COPIED

: COPY FLAVORS RED WHITE BLUE <cr> <----- Multiple data items copied
TO: ALPHA BETA GAMMA <cr>
3 ITEMS COPIED

```

Figure B: Copying Items to Same File

FILE MANAGEMENT PROCESSORS

When copying lists, care should be taken to ensure that all of the source items exist. If an item being copied is 'NOT ON FILE', it causes the source-destination correspondence to be out of sequence.

Figure C illustrates a COPY of all dictionary items from file SAMPLE to the dictionary of file FLAVORS. Note that when a destination of other than the source file is desired, the destination file must be enclosed in parentheses. (If a destination file is not specified, the items will be copied into the original file). Note also that if the dictionary is the destination, the word DICT must be included within the parentheses (see Figure C).

When copying from one dictionary to another, the COPY processor does not copy items which have D/CODE=D (such as the DL/ID entry). DL/ID entries must only be created by the CREATE-FILE processor. To recreate both dictionary and data sections of one file in a new file, a command sequence such as that shown in Figure D can be used (but see also the next topic for implicit creation of a new file). Great care should be taken when copying to dictionaries to avoid overlaying D-pointers when using the 'O' option.

Note: to COPY to or from another user's account, a file synonym definition (D/CODE=Q) which points to that specific account and file must already exist.

```

-----
:COPY DICT SAMPLE * <cr> <----- All dictionary items copied.
TO:(DICT FLAVORS) <cr>
DL/ID NOT COPIED
2 ITEMS COPIED

:COPY INV 1007 <cr> <----- Copy one item to a different
TO:(PARTS) A1007-1 <cr>          file with a new item-id.
1 ITEMS COPIED
-----

```

Figure C: Copying Items to Different File

```

-----
:CREATE-FILE NEW-SAMPLE 1 3<cr> <-- New file created
[417] FILE 'NEW-SAMPLE' CREATED; BASE = 25417, MODULO = 1, SEPAR = 1.
[417] FILE 'DL/ID' CREATED: BASE = 25418, MODULO = 3, SEPAR = 1.

:COPY DICT SAMPLE * <cr> <----- All dictionary items (except
TO:(DICT NEW-SAMPLE)<cr>          DL/ID) copied.
DL/ID NOT COPIED
3 ITEMS COPIED

:COPY SAMPLE * <cr> <----- All data items copied.
TO:(NEW-SAMPLE)<cr>
2 ITEMS COPIED
-----

```

Figure D: Creation of Entire Dictionary and Data Sections

ADDITIONAL COPY PROCESSOR FEATURES

| The COPY processor may be used to copy items to the printer or to the |
| terminal, and may be used to create new files. Several options are |
additionally available for use with the COPY processor.

Items may be copied to the terminal by entering a null line (carriage return only) in response to the "TO:" message or by using the 'T' option (Figure B).

Several options are available for use with the COPY processor. Desired options are specified by one or more single characters. The entire options list must be enclosed in parentheses and must follow the item-list in the COPY command (see the topic COPYING ITEMS: COPY PROCESSOR). Valid options are listed in Figure A.

Note: Not all combinations of option characters are meaningful. 'D' and 'O' options only apply on a file-to-file copy. 'X' and 'L' options apply only on a copy to the terminal or printer.

The sample in Figure C copies two items from the FLAVORS file back into the FLAVORS file, changing the item-ids from ALPHA to BLUE and from BETA to YELLOW. Items ALPHA and BETA are deleted. If either item previously existed (BLUE or YELLOW), it is overwritten.

The example in Figure D shows the creation of a new file and copying the entire 'XYZ' file to 'ZZZ'. The modulo and separation are specified as for a CREATE-FILE.

Note: When using the file creation facility of COPY, the data section should be copied before the dictionary.

<u>OPTION</u>	<u>DESCRIPTION</u>
D	Deletes items from source file after they are copied to the destination file. (This option may not be used with the 'P' or 'T' option.)
I	Lists destination item-ids as they are copied.
L	Suppresses line-numbers on a copy to the terminal or printer, when used with the 'T' or 'P' option (respectively).
N	Suppress automatic paging of output when used with the 'T' option.
O	Overwrites destination file items with source file items if item-ids are the same. (This option may not be used with 'P' or 'T' option.)
P	Copies items to the printer (spooler).
S	Suppresses printing of "item-id" NOT COPIED messages.
T	Copies items to your terminal.
X	Specifies a hexadecimal dump to the terminal or printer, when used with the 'T' or 'P' option (respectively).

Figure A: COPY Processor Options.

```

:COPY M/DICT LISTACC <cr>
TO:<cr>

(Item LISTACC will be displayed at the terminal.)

```

Figure B: Copying Items to Terminal

```

:COPY FLAVORS ALPHA BETA (DO) <cr>
TO:BLUE YELLOW <cr>

2 ITEMS COPIED

```

Figure C: Sample Usage of COPY Options

```

:COPY XYZ * <cr>
TO:(ZZZ 1 1)<cr>

[417] FILE 'ZZZ' CREATED, BASE=23633, MODULO=1, SEPAR=1
[417] FILE 'DL/ID' CREATED, BASE=23634, MODULO=1, SEPAR=1

6 ITEMS COPIED

(NOTE: Data items are copied, not dictionary items)

```

Figure D: Creating New File During Copy Process

7. SPOOLER

=====

OVERVIEW

The system enables multiple users to perform processing independently and (apparently) simultaneously. Each system printer and magnetic tape unit can perform just one task at a time, however. A terminal independent process, known as the 'Spooler', is therefore provided to queue jobs to the appropriate devices (including system ports) and output these jobs one at a time. The spooler allows each process to continue processing, whether or not the appropriate device is immediately available, and provides a number of other useful facilities.

A number of commands are provided to control and report the spooler process.

This section describes the use of the spooler and the relevant commands.

SPOOLER

SUMMARY OF SPOOLER FUNCTIONS

PRINT JOBS

Individual reports to be spooled to an output device are called "print jobs". A print job may be created by a PROC, an ENGLISH statement, a DATA/BASIC program or other processor. Before a print job can be output, it must be assigned to a form queue. The spooler will accommodate up to 300 print jobs.

FORM QUEUE

A form queue is a "list" of print jobs waiting to be despoiled. Print jobs are assigned to a form queue via the SP-ASSIGN verb. Once created, the form queue may be assigned to an output device (a system printer, a tape unit, or a port). If the device assignment is successful (i.e., no other form queue is assigned to that device), output may then begin.

QUEUE ASSIGNMENT

When the system is initialised, a form queue STANDARD is created and assigned to printer #1. Any print job created without an SP-ASSIGN statement will be put in the STANDARD form queue. You may modify this assignment at any time by executing the SP-ASSIGN verb. Several options are available (i.e. creation of hold files, instant output, nonspooling, and suppressed output). See the next topic, ASSIGNMENT OF PRINT JOBS AND OPTIONS: SP-ASSIGN VERB.

NOTE: Although the system allows you to change the name of the STANDARD form queue, the default queue will be referred to as STANDARD throughout this manual.

LINE-AT-A-TIME MODE

Using the SP-ASSIGN verb with an 'N' option, you may bypass the spooler. In this case, the process outputs text directly and immediately to the attached output device. This is called line-at-a-time output.

Figure A shows a listing of the major spooler features. Figure B illustrates a procedure that may be used to spool a print job to an output device.

DEVICE ATTACHMENT

The system avoids conflicts by permitting only one process at a time to use each device. System software keeps track of device attachments. Spooler users need not be concerned with attachment/detachment since the spooler automatically attaches the appropriate unit before output begins, and automatically detaches the device when the queue is empty.

The spooler offers the following features:

1. Simultaneous output to the system printer(s), magnetic tape unit(s), and system ports within the system configuration.
2. Ability to restart a print job when the printer malfunctions.
3. Creation of as many form queues as necessary.
4. Movement of one or more print jobs from one form queue to another.
5. Ability to change order of print jobs in queue.
6. Reassignment of devices to form queues by device type and number.
7. Specification of up to 100 print job copies.
8. Examination of closed jobs.
9. SP-STATUS display to show the current status of all devices and spooled reports, open and closed.
10. Reports when spooler has aborted via SP-STATUS.
11. System restart without loss of closed hold files.

Figure A: Major Features of the Spooler.

A procedure that may be used to spool a print job to an output device:

1. Create a form queue
2. Assign the print job to a form queue
3. Assign form queue to an output device

Figure B: Sample Procedure for Spooling a Print Job

SPOOLER

HOLD FILES

The creation of hold files permits print jobs to be spooled to the disk and held for an indefinite length of time for subsequent output. The spooler may accommodate up to 300 jobs including hold files. The actual number of entries your spooler can handle is dependent on the amount of overflow space available. Hold files are normally created for jobs requiring special forms or very lengthy jobs. Hold files may be recovered if an 'X' or 'A' system restore or :RESTART-SPOOLER is performed.

Hold files generated while PROC is in control are recorded in the PROC secondary input buffer, together with error messages. Hold file numbers are recorded in the buffer in the format:

HFNn

where 'n' is the hold file number.

SPOOLER VERBS

Figure A summarises those verbs capable of communicating with the spooler. The "Screen-Orientated Verbs" (SP-STATUS and SP-JOBS) produce a screen display showing current spooler status. Immediately following each display is a "menu" of action codes you may enter to perform the desired spooling function. These action codes are listed below SP-STATUS and SP-JOBS in Figure A.

Verbs listed under "TCL Verbs" perform the same functions as the action codes displayed on the same line (where an action code is shown). However, the "TCL Verbs" do not cause a status display to appear.

If you do not remember the sequence of required parameters needed to execute a spooler verb, press the RETURN key immediately after entering the verb. The system will then display the required parameter(s) in the correct sequence.

<u>Screen-Orientated Verbs</u>	<u>TCL Verbs</u>	<u>Explanation</u>
SP-STATUS		Displays status of spooler and all output devices.
Create Form Queue	SP-CREATE	Creates a form queue and optionally assigns it to a device.
Change Form Queue	SP-FORM	Changes the name of a form queue
Change Form Type	SP-TYPE	Changes type of form used.
Change Device Assignment	SP-DEVICE	Assigns device to a form queue.
Change Page Skip	SP-SKIP	Changes number of pages ejected from printer after exhausting the form queue.
List Print Jobs	SP-JOBS	Displays SP-JOBS screen.
Delete Form Queue	SP-FQDELETE	Causes form queue to be released from table.
Page Eject	SP-EJECT	Causes attached printer to eject specified number of pages
99		Exits screen display.
SP-JOBS		Displays status of print jobs.
Move Form Queue	SP-MOVEQ	Moves all print jobs of form queue to another form queue.
Move Print Job	SP-SWITCH	Moves a print job from current form queue to another.
Change Options	SP-OPTS	Changes options on a print job.
Change #of Copies	SP-COPIES	Specifies number of times a print job will be printed.
Delete Job	SP-DELETE	Deletes one or more print jobs.
Stop Printing	SP-STOP	Stops printer.
Resume Printing	SP-RESUME	Reactivates spooler.
Edit Print Job	SP-EDIT	Allows examination of print jobs
Change Priority	SP-PRIORITY	Changes print job priority within a form queue.
Suspend Printing	SP-SUSPEND	Cancels printing but does not delete job
SP-STATUS	SP-STATUS	Display SP-STATUS screen.
99		Exits screen display.
	SP-ASSIGN	Assigns option codes, a print job code, and/or assigns subsequent print jobs to a queue.
	SP-LOOK	Displays parameters of current statement.
	SP-PORTON	Starts terminal as a spooler port.
	SP-PORTOFF	Logs off an SP-PORTON line.
	SP-KILL	Cancels and deletes currently printing job.
	SP-RETURN	Changes the number of frames the spooler returns to overflow
	SP-TAPEOUT	Spools a job from a spool tape
	:SP-NEWTAB	Reinitialize the spooler
	:RESTART-SPOOLER	Restart the spooler

Figure A: Summary of Spooler Verbs

ASSIGNMENT OF PRINT JOBS AND OPTIONS: SP-ASSIGN VERB

The SP-ASSIGN verb assigns print jobs to a form queue. It may also assign options to print jobs, assign a print report number to a queue, and/or specify number of copies.

The general form of the SP-ASSIGN verb is:

SP-ASSIGN {{n}=form-queue-name {copies}}

or

SP-ASSIGN {{n}=form-queue-name} {options {copies}}

where:

n	corresponds to a print report number in a DATA/BASIC program PRINT ON statement (see the DATA/BASIC Reference Manual). Assignments made without 'n' become the default assignment, and all previous assignments are cleared. Any subsequent assignments which include 'n' apply only to jobs with the corresponding print report numbers (as generated by a DATA/BASIC program). Note that the print report number is not related to the print job number. Note: If 'n' is included in the command, a form queue name must also be included.
=form-queue-name	is the name of the queue to receive print job(s). Omission of this parameter implies an assignment to the form-queue STANDARD.
copies	is a number from 1 to 100 which specifies the number of copies to be printed. If 'copies' is omitted, one copy is implied.
options	may be one or more of the following: H (creation of hold files), I (instant output), N (no spooling) and S (suppress output). Figure A shows incompatible options.

If no SP-ASSIGN statement is used, print jobs are put in the STANDARD form queue.

An SP-ASSIGN statement executed with no parameters or options resets the job assignment to the default queue STANDARD. This also occurs when the operator logs off.

Options may be input in any sequence and need not be separated from each other. Options remain in effect until the next use of the SP-ASSIGN without a specific print report number, when the options (and copies) are reset. Descriptions of these options follow.

H - CREATE HOLD FILE ON DISK

The 'H' option retains a print job on disk after printing is completed. Jobs which require special forms or special manipulations can be handled with this option. When a print job is assigned to a hold file, the message "ENTRY #m" appears on the screen, where 'm' is the print job number of the hold file. The 'H' option cannot be used with the 'N' option.

I - INSTANT OUTPUT (not recommended for use on this release)

Start output as soon as the first frame of output is available (do not wait for report to be closed) and release frames back to the overflow table as soon as they are printed. Should a malfunction occur while outputting a print job with an 'I' option, you may suspend printing; however, you cannot recover any data already printed. SP-RESUME causes output to begin where it left off. The 'I' option is incompatible with multiple copies and the 'N' option.

N - NO SPOOLING (OUTPUT DIRECTLY TO DEVICE)

The 'N' option specifies no spooling. Data is output a line at a time to the assigned device. If the device is not available, you are notified and the 'N' option is removed from the assignment. This option should be used only when the intent is to prohibit other users from accessing the output device. The device(s) will become attached to your line and will require you to detach the device(s) when complete so that others (including the spooler itself) may gain control. The 'N' option is incompatible with the 'I' or 'H' options, multiple copies and with output to a port.

S - SUPPRESS OUTPUT

The 'S' option suppresses spooler output. Normally, the 'S' option is used with the 'H' option. However, if the 'S' option appears alone, an output job is built but not printed because the job is not queued for output.

<u>Spooler Options</u>	<u>Incompatible Option and Device Combinations</u>
H	N
I	N, Copies
N	I, H, Copies, form-queue assigned to port
Copies	I, N

Note: Any of the above combinations give the following error message:
 INCOMPATIBLE OPTIONS: NI, NH, N COPIES, I COPIES, OR N PORT.
 In this case, the assignment is still effective but the last-entered incompatible parameter(s) are ignored.

Figure A: Display of Incompatible Spooler Option and Device Combinations.

GENERAL FORMAT AND EXAMPLES OF SP-ASSIGN VERB

The general format of the SP-ASSIGN verb is shown in Figure A. Note that each parameter is separated by one space. Figure B illustrates the use of the SP-ASSIGN verb.

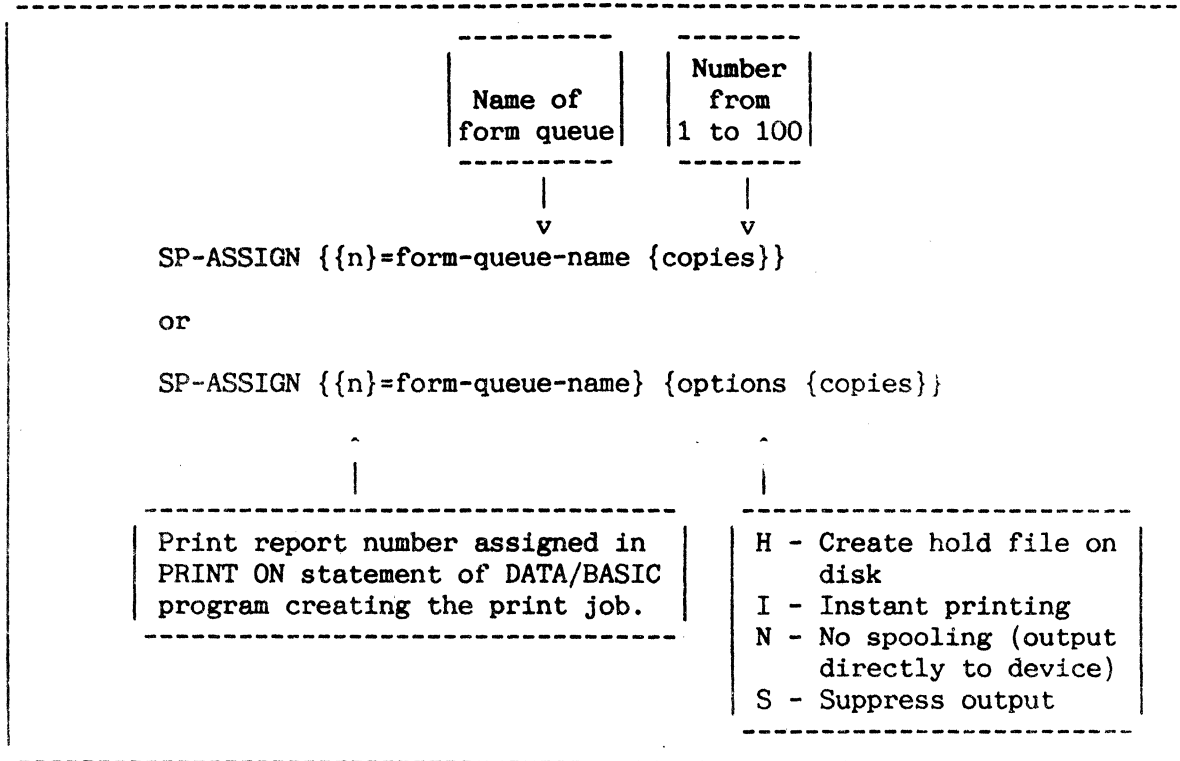


Figure A: General Form of SP-ASSIGN Verb

<u>EXAMPLE</u>	<u>EXPLANATION</u>
:SP-ASSIGN	Spooler device assignments are cleared and all future print jobs are assigned to form queue STANDARD.
:SP-ASSIGN =INVOICE :LIST ENG '750L' (P)	A print job containing item "750L" is assigned to the form queue INVOICE.
:SP-ASSIGN 1 =CHECK I	All future print jobs with a print report number of "1" will be assigned to form queue CHECK with the option of instant printing.
:SP-ASSIGN =STANDARD HS :RUNOFF INV RECEIPTS (P) ENTRY #6	All future print jobs will be assigned to the STANDARD form queue as hold files with suppressed printing. Print job #6, which contains the item "RECEIPTS", is now a hold file.
:SP-ASSIGN =CHECK N :LIST PROD 'PART#' (P)	A print job containing the item "PART#" is assigned to form queue CHECK and is output a line at a time to the assigned device.
:SP-ASSIGN H	All future print jobs assigned to the default form queue will be hold files.
:SP-ASSIGN =INVOICE :SP-ASSIGN 1=CHECK HS	All jobs except those with print report #1 will be assigned to INVOICE.

Figure C: Sample Use of SP-ASSIGN Verb

DISPLAYING SP-ASSIGN PARAMETERS: SP-LOOK VERB

| The SP-LOOK verb displays the parameters used in the last default |
SP-ASSIGN statement executed at your terminal.

The format of the SP-LOOK verb is

SP-LOOK {line #}

To display the parameters of your current default SP-ASSIGN statement, enter:

:SP-LOOK <cr>

If you wish to display the default SP-ASSIGN parameters for another terminal, enter:

:SP-LOOK line#

where 'line#' represents the port number you wish to access. If the port number you enter is greater than or equal to the spooler's line number, the message "LINE # TOO LARGE" is displayed and you are then returned to TCL.

After executing the SP-LOOK verb, you may also see the following message:

DEFAULT ASSIGNMENT IS TO QUEUE "STANDARD".

This message appears:

1. After you log on and before either an SP-ASSIGN statement is entered or before a print job is run.
2. After you enter an SP-ASSIGN statement with no parameters and before a print job is run.

The system maintains an assignment table that logs in your current SP-ASSIGN statement. The above message appears because you do not currently have an entry in the table. If you execute an SP-ASSIGN statement without parameters or options, any previous table entry assigned to your process is removed. Should you execute an output operation without an entry in the assignment table, one will be created and assigned to the STANDARD form queue.

Figure B illustrates the use of SP-LOOK when viewing the SP-ASSIGN parameters for your own terminal. Figure C shows you how to display the default SP-ASSIGN parameters for another port.

```
-----  
| SP-LOOK {line#} |  
-----
```

Figure A: General Format of SP-LOOK Verb

```
-----  
| :SP-LOOK <cr>  
|  
| FORM QUEUE = STANDARD  
| DEVICE TYPE = LINE PRINTER  
| DEVICE NUMBER = 1  
| SPOOLER OPTIONS = HS  
| COPIES = 1  
|  
|-----
```

Figure B: Displaying SP-ASSIGN Parameters For Your Own Terminal

```
-----  
| :SP-LOOK 18 <cr>  
|  
| FORM QUEUE = TAPEOUT  
| DEVICE TYPE = TAPE  
| DEVICE NUMBER = 1  
| SPOOLER OPTIONS =  
| COPIES = 1  
|  
|-----
```

Figure C: Displaying SP-ASSIGN Parameters for Port 18

SPOOLER

DISPLAYING STATUS OF SPOOLER: SP-STATUS VERB

The SP-STATUS verb displays current spooler status.

The general form of the SP-STATUS verb is:

SP-STATUS

The SP-STATUS verb displays:

- 1) the name of each form queue
- 2) the types of forms used
- 3) the device number assigned to each form queue
- 4) the type of output device
- 5) the status of each device
- 6) the number of print jobs waiting to be printed
- 7) the number of page skips specified.

The possible indications for device status are shown in Figure A.

At any given time, more than one of the status conditions listed may exist (e.g. ACTIVE and ASSIGNED or SUSPEND and NOT READY); however, only one status condition is displayed. The display priority is: CABLE OFF, NEXT REEL, PARITY, SUSPEND, NEED DEV, NOT READY, STOP, ACTIVE, ASSIGNED, WAITING.

Immediately following the status display is a list of action codes. You may enter the desired action code number in response to the prompt "ENTER ACTION CODE":.

If the RETURN key is pressed in response to the "ENTER ACTION CODE:" prompt, the SP-STATUS screen is displayed again. To exit from the SP-STATUS display, enter Action Code 99.

Each action code and equivalent spooler verb is described in the following sections.

See Figure B for a sample screen layout which would follow the execution of the SP-STATUS verb.

<u>Status</u>	<u>Explanation</u>
NEED DEV	The desired device is attached to another line.
CABLE OFF	The cable to the line printer is not plugged in.
NOT READY	Device is off-line or, if device is a tape unit, the mounted tape has no write ring.
NEXT REEL	Tape drive needs a new reel.
PARITY	A parity error has occurred on tape.
SUSPEND	A suspend order has been issued.
STOP	A stop order has been issued.
ACTIVE	Device is outputting a job.
ASSIGNED	The queue is assigned to a device but there are no jobs ready for output.
WAITING	Queue is waiting for a device that is currently assigned to another queue.
NO DEVICE	Queue is not assigned to a device.

Figure A: Device Status

```

:SP-STATUS <cr>

                                FORM QUEUE ASSIGNMENT

  QUEUE      FORM      DEVICE      DEVICE      DEVICE      NUMBER      PAGE
  NAME       TYPE      ASSIGNED  TYPE        STATUS      ENTRIES     SKIP

  STANDARD   1-PART     1         LPTR        ACTIVE      7           0
  CHECK      PAYROLL    1         TAPE        NOT READY   1           0
  LIST-T     0             0         0           NO DEVICE   0           0
  LABEL      MAIL       2         PORT        ASSIGNED    0           0
  SAMPLE     3             3         LPTR        ASSIGNED    0           0
  PAYABLES   0             0         LPTR        NO DEVICE   1           0
  .          .         .         .           .           .           .
  .          .         .         .           .           .           .

                                ACTION CODES

  1. CREATE FORM QUEUE  4. CHANGE DEVICE ASSIGNMENT  7. DELETE FORM QUEUE
  2. CHANGE FORM QUEUE  5. CHANGE PAGE SKIP          8. PAGE EJECT
  3. CHANGE FORM TYPE   6. LIST PRINT JOBS          99. EXIT

  ENTER ACTION CODE:

```

Figure B: Sample Screen Display of SP-STATUS Verb

SPOOLER

SPOOLER OUTPUT DEVICES

| The spooler can output print jobs to three types of device: line
printers, tape units, and terminal ports.

LINE PRINTERS

The spooler supports any system printers configured on the system. Each printer is assigned a device number in the range 1 to the number of system printers supported. When the system is initialised, the default form queue STANDARD is assigned to printer #1.

TAPE UNITS

When a job is spooled to tape, the output is written to the magnetic tape unit starting wherever the tape is currently positioned. The tape unit is rewound by the spooler only when the end of a reel is reached; thus, jobs may be spooled to tape one after another. You must mount the tape (with write ring), place the unit on-line, and rewind or position the tape prior to using the spooler.

Caution should be used when spooling to tape while some other user is using the tape drive (a status of "NEED DEV"). The spooler will wait for the tape to become available before attempting to write. If the other user rewinds the tape and detaches it, the spooler will gain control of the tape and start writing the spooled output.

Normally the spooler writes unlabelled tape jobs consisting of 500-byte records. However, if the 'N' option is in effect, the records will be of varying length and each record will terminate at the end of a print line closest to 500 bytes. In other words, the records will be split at line boundaries if the 'N' option is in effect, rather than at the 500th character. This is useful when taking spooled reports to other devices that must have full lines within records.

TERMINAL PORTS

The spooler may spool output to any port if assigned to a queue. Any number of these ports may be attached to terminals or serial printers.

Note: You may not assign Port Zero as a spooler port.

Figure A listed the terms used to represent the three types of spooler output devices and a brief description of each.

<u>Device Type</u>	<u>Brief Explanation</u>	<u>Legal Device# Range *</u>
LPTR	Line printer	1
TAPE	Magnetic Tape Unit; spooler supports a maximum of three.	1 - 3
PORT	Terminal Port; spooler supports as many ports as the system configuration allows.	1 - 63

* actual device # range depends on system configuration.

Figure A: Terms Used to Define Spooler Output Devices

SPOOLER

CREATING A FORM QUEUE

A form queue is created by the execution of either the SP-STATUS Action Code 1 (Create Form Queue) or the SP-CREATE verb.

The general form of the Create Form Queue Action Code is:

```
ENTER ACTION CODE:1 <cr>
FORM-QUEUE DEVICE-TYPE(LPTR,TAPE,PORT) DEVICE#:
```

The following responses are required, each separated from the next by one space:

FORM-QUEUE a form queue name consisting of 20 or fewer alphabetic, numeric, or special characters (not including asterisks or embedded spaces, and not consisting of a single numeric character).

DEVICE-TYPE one of the following:

LPTR	Line Printer
TAPE	Tape Drive
PORT	Terminal Port
0	No Device

DEVICE# the number assigned to the output device. Each system printer is assigned a device number from 1 to 4. The tape unit is device numbered 1,2, or 3. PORT device may have a device number up to the maximum number of ports on the system.

The format of the SP-CREATE verb is:

```
SP-CREATE form-queue device-type device#
```

A device number and device type may both be "0" (zero). In this case, the form queue is created but no print jobs are output until the queue is assigned a device by either the SP-STATUS Action Code 4 (Change Device Assignment) or the SP-DEVICE verb (refer to topic entitled CHANGING DEVICE ASSIGNMENT). Should you create a queue and assign it to a device already assigned to another queue, the following message appears:

```
A QUEUE IS ALREADY ASSIGNED TO THIS DEVICE
OK TO CHANGE?
```

A response of "N" causes your queue to be created but with no device assignment. If you respond with "Y", the print job currently being output on the desired device will first be completed. Until this process is finished, your newly created queue has a status of "WAITING". As soon as the job is completed, the device is reassigned to your new queue; the previously assigned queue is then given a status of "NO DEVICE".

Figure A illustrates the use of SP-STATUS Action Code 1. Figure B illustrates the use of the SP-CREATE verb, and the sample SP-STATUS display which results from either this or the previous example.

```

-----
:SP-STATUS <cr>

                                FORM QUEUE ASSIGNMENT

  QUEUE      FORM      DEVICE  DEVICE  DEVICE  NUMBER  PAGE
  NAME       TYPE      ASSIGNED TYPE   STATUS  ENTRIES SKIP

  STANDARD   1-PART    1      LPTR   ACTIVE   7       0
  CHECK      PAYROLL   1      TAPE   NOT READY 1       0
  LIST-T     0         0      0      NO DEVICE 0       0
  LABEL      MAIL     2      PORT   ASSIGNED 0       0
  SAMPLE     3         3      LPTR   ASSIGNED 0       0
  PAYABLES   0         0      LPTR   NO DEVICE 1       0

                                ACTION CODES

  1. CREATE FORM QUEUE  4. CHANGE DEVICE ASSIGNMENT  7. DELETE FORM QUEUE
  2. CHANGE FORM QUEUE  5. CHANGE PAGE SKIP          8. PAGE EJECT
  3. CHANGE FORM TYPE   6. LIST PRINT JOBS           99. EXIT

  ENTER ACTION CODE:1 <cr>
  FORM-QUEUE DEVICE-TYPE(LPTR,TAPE,PORT)
  DEVICE#:PART LPTR 2 <cr>

                                The Create Form Queue
                                Action Code produces the
                                form queue PART which is
                                assigned to printer #2.
-----

```

Figure A: Sample Usage of Create Form Queue Action Code

```

-----
:SP-CREATE PART LPTR 2 <cr>
:SP-STATUS <cr>

                                FORM QUEUE ASSIGNMENT

  QUEUE      FORM      DEVICE  DEVICE  DEVICE  NUMBER  PAGE
  NAME       TYPE      ASSIGNED TYPE   STATUS  ENTRIES SKIP

  STANDARD   1-PART    1      LPTR   ACTIVE   7       0
  CHECK      PAYROLL   1      TAPE   NOT READY 1       0
  LIST-T     0         0      0      NO DEVICE 0       0
  LABEL      MAIL     2      PORT   ASSIGNED 0       0
  SAMPLE     3         3      LPTR   ASSIGNED 0       0
  PAYABLES   0         0      LPTR   NO DEVICE 1       0
  PART       2         2      LPTR   ASSIGNED 0       0

                                The SP-CREATE verb
                                performs the same
                                function as Action code 1
-----

```

Figure B: Sample Usage of SP-CREATE Verb and SP-STATUS display after execution.

SPOOLER

CHANGING THE NAME OF A FORM QUEUE

| The name of a form queue may be changed by the execution of either the |
SP-STATUS Action Code 2 (CHANGE FORM QUEUE) or the SP-FORM verb.

The format of the Change Form Queue Action Code is:

ENTER ACTION CODE: 2 <cr>
OLD-NAME NEW NAME:

where "OLD-NAME" and "NEW-NAME" are prompts specifying the order of the required response.

The format of the SP-FORM verb is:

SP-FORM old-name new-name

Action Code 2 and the SP-FORM verb both change the name of an already established form queue. The new queue name remains in effect until another name is entered by one of these two methods.

You may change the name of STANDARD; however, the queue itself remains as the default form queue.

Figures A and B illustrate the use of the Change Print Action Code and the SP-FORM verb. Figure C displays the SP-STATUS screen which would follow the execution of the example in Figure A or B.

```

:SP-STATUS <cr>
                                FORM QUEUE ASSIGNMENT

  QUEUE   FORM   DEVICE   DEVICE   DEVICE   NUMBER   PAGE
  NAME    TYPE   ASSIGNED  TYPE     STATUS   ENTRIES  SKIP

  STANDARD 1-PART   1       LPTR     ACTIVE   7        0
  CHECK    PAYROLL 1       TAPE     NOT READY 1        0
  LIST-T   0       0       NO DEVICE 0        0
  LABEL    MAIL    2       PORT     ASSIGNED 0        0
  SAMPLE   3       LPTR     ASSIGNED 0        0
  PAYABLES 0       LPTR     NO DEVICE 1        0
  PART     2       LPTR     ASSIGNED 0        0

                                ACTION CODES
  1. CREATE FORM QUEUE  4. CHANGE DEVICE ASSIGNMENT  7. DELETE FORM QUEUE
  2. CHANGE FORM QUEUE  5. CHANGE PAGE SKIP          8. PAGE EJECT
  3. CHANGE FORM TYPE   6. LIST PRINT JOBS          99. EXIT

  ENTER ACTION CODE:2 <cr>
  OLD-NAME NEW-NAME:STANDARD INVOICES <cr>
                                The Change Form Queue
                                Action Code alters the
                                name of form queue
                                STANDARD to INVOICES.

```

Figure A: Sample Usage of Change Form Queue Action Code

```

:SP-FORM STANDARD INVOICES <cr>
                                The SP-FORM verb is performing
                                the same function as Action Code
                                2 above. Notice that you must
                                enter "SP-STATUS" to get screen
                                display.

:SP-STATUS <cr>

```

Figure B: Sample Usage of SP-FORM Verb

```

                                FORM QUEUE ASSIGNMENT

  QUEUE   FORM   DEVICE   DEVICE   DEVICE   NUMBER   PAGE
  NAME    TYPE   ASSIGNED  TYPE     STATUS   ENTRIES  SKIP

  INVOICES 1-PART   1       LPTR     ACTIVE   7        0
  CHECK    PAYROLL 1       TAPE     NOT READY 1        0
  LIST-T   0       0       NO DEVICE 0        0
  LABEL    MAIL    2       PORT     ASSIGNED 0        0
  SAMPLE   3       LPTR     ASSIGNED 0        0
  PAYABLES 0       LPTR     NO DEVICE 1        0
  PART     2       LPTR     ASSIGNED 0        0

```

Figure C: Sample SP-STATUS Screen Display

SPOOLER

CHANGING TYPE OF FORM USED ON PRINTER

The SP-STATUS Action Code 3 (Change Form Type) and the SP-TYPE verb allow you to specify a form type for each queue. This form type will be displayed on the SP-STATUS screen and serves to remind you of the type of form required on the assigned device.

The format of the Change Form Type Action Code is:

```
ENTER ACTION CODE:3 <cr>  
FORM-QUEUE FORM-TYPE:
```

where:

FORM-QUEUE is prompting for the name of the form queue assigned to the printer.

FORM-TYPE is prompting for the name of the form to be placed on the printer.

The format of the SP-TYPE verb is:

```
SP-TYPE form-queue form-type
```

Note that the terms "FORM-QUEUE" and "FORM-TYPE" refer to two different elements of a form queue. The form queue is associated with devices, not necessarily with the type of forms. There may be a large number of uniquely defined form queues. It is possible to change the form type without changing the name of the form queue because the "form-type" is not used by the spooler. It is intended only as a reminder to you of the type of form that should be on the specified device.

Figures A and B illustrate the use of the Change Form Type Action Code and the SP-TYPE verb. Figure C displays the SP-STATUS screen which would follow the execution of the example in Figure A or B.

```

:SP-STATUS <cr>
                                FORM QUEUE ASSIGNMENT

QUEUE      FORM      DEVICE      DEVICE      DEVICE      NUMBER      PAGE
NAME       TYPE       ASSIGNED   TYPE        STATUS      ENTRIES     SKIP

INVOICES   1-PART      1          LPTR        ACTIVE      7           0
CHECK      PAYROLL      1          TAPE        NOT READY   1           0
LIST-T     0              0          0           NO DEVICE   0           0
LABEL      MAIL         2          PORT        ASSIGNED    0           0
SAMPLE     3              3          LPTR        ASSIGNED    0           0
PAYABLES   0              0          LPTR        NO DEVICE   1           0
PART       2              2          LPTR        ASSIGNED    0           0

                                ACTION CODES
1. CREATE FORM QUEUE  4. CHANGE DEVICE ASSIGNMENT  7. DELETE FORM QUEUE
2. CHANGE FORM QUEUE  5. CHANGE PAGE SKIP          8. PAGE EJECT
3. CHANGE FORM TYPE   6. LIST PRINT JOBS          99. EXIT

ENTER ACTION CODE:3 <cr>
FORM-QUEUE FORM-TYPE:INVOICES INVOICE <cr>
                                The Change Form Type
                                Action Code changes the
                                form type of form queue
                                INVOICES to INVOICE.

```

Figure A: Sample Usage of Change Form Type Action Code

```

:SP-TYPE INVOICES INVOICE <cr>
                                This SP-TYPE verb performs the
                                same function as the action
                                code in Figure A.
:SP-STATUS <cr>

```

Figure B: Sample Usage of SP-TYPE Verb

```

                                FORM QUEUE ASSIGNMENT

QUEUE      FORM      DEVICE      DEVICE      DEVICE      NUMBER      PAGE
NAME       TYPE       ASSIGNED   TYPE        STATUS      ENTRIES     SKIP

INVOICES   INVOICE      1          LPTR        ACTIVE      7           0
CHECK      PAYROLL      1          TAPE        NOT READY   1           0
LIST-T     0              0          0           NO DEVICE   0           0
LABEL      MAIL         2          PORT        ASSIGNED    0           0
SAMPLE     3              3          LPTR        ASSIGNED    0           0
PAYABLES   0              0          LPTR        NO DEVICE   1           0
PART       2              2          LPTR        ASSIGNED    0           0

```

Figure C: SP-STATUS Display Screen

SPOOLER

CHANGING DEVICE ASSIGNMENTS

| A device assignment may be changed by either the SP-STATUS Action Code 4 |
(Change Device Assignment) or the SP-DEVICE verb.

The general form of the Change Device Assignment Action Code is:

```
ENTER ACTION CODE:4 <cr>  
FORM-QUEUE DEVICE-TYPE(LPTR,TAPE,PORT) DEVICE#:
```

The format of the SP-DEVICE verb is:

```
SP-DEVICE form-queue device-type device#
```

Changing a device assignment is especially useful when a printer is idle but there are several form queues waiting to be printed. In this case, you may assign the idle printer to a form queue waiting to be printed.

In general, there may be only one queue assigned to each device at any time. For example, should you attempt to assign Form Queue B to LPTR 2 and this device is already assigned to Form Queue A, you are asked if such a device reassignment is permissible. If permission is granted to change the device assignment, a stop order is issued to Form Queue A and its assignment to LPTR 2 will be cancelled as soon as the current job has been printed. Form Queue B will acquire a "WAITING" status until Form Queue A is finished printing; Form Queue B will then be assigned to LPTR 2.

The range of legal device numbers is given in the topic titled "SPOOLER OUTPUT DEVICES". If the device number specified is not within range, the following message will appear:

```
INVALID DEVICE# RANGE
```

Note:

The spooler has no way of knowing which devices are physically present. It is your responsibility to ensure that the device number specified is actually present on the system.

Figures A and B illustrate the use of the Change Device Assignment Action Code and the SP-DEVICE verb. Figure C displays the SP-STATUS screen which would follow the execution of the example in Figure A or B.

```

:SP-STATUS <cr>
                                FORM QUEUE ASSIGNMENT

  QUEUE   FORM   DEVICE   DEVICE   DEVICE   NUMBER   PAGE
  NAME    TYPE   ASSIGNED  TYPE     STATUS   ENTRIES  SKIP

  INVOICES  INVOICE   1     LPTR    ACTIVE    7        0
  CHECK     PAYROLL   1     TAPE    NOT READY 1        0
  LIST-T    0         0     0       NO DEVICE 0        0
  LABEL     MAIL      2     PORT    ASSIGNED  0        0
  SAMPLE    3         3     LPTR    ASSIGNED  0        0
  PAYABLES  0         0     LPTR    NO DEVICE 1        0
  PART      2         2     LPTR    ASSIGNED  0        0

                                ACTION CODES
  1. CREATE FORM QUEUE  4. CHANGE DEVICE ASSIGNMENT  7. DELETE FORM QUEUE
  2. CHANGE FORM QUEUE  5. CHANGE PAGE SKIP          8. PAGE EJECT
  3. CHANGE FORM TYPE   6. LIST PRINT JOBS           99. EXIT

  ENTER ACTION CODE:4 <cr>
  FORM-QUEUE DEVICE-TYPE(LPTR,TAPE,PORT)
  DEVICE#: PAYABLES LPTR 1 <cr>

  A QUEUE IS ALREADY ASSIGNED TO THIS DEVICE
  OK TO CHANGE?Y <cr>

  The Change Device Assign-
  ment Action Code assigns
  form queue PAYABLES to
  printer #1.
  A response of 'Y' permits
  the device reassignment.
  
```

Figure A: Sample Usage of Change Device Assignment Action Code

```

:SP-DEVICE PAYABLES LPTR 1 <cr>
:SP-STATUS <cr>

  The SP-DEVICE verb performs
  the same function as Action
  Code 7 in Figure A.
  
```

Figure B: Sample Usage of SP-DEVICE Verb

```

                                FORM QUEUE ASSIGNMENT

  QUEUE   FORM   DEVICE   DEVICE   DEVICE   NUMBER   PAGE
  NAME    TYPE   ASSIGNED  TYPE     STATUS   ENTRIES  SKIP

  INVOICES  INVOICE   0     0       NO DEVICE 6        0
  CHECK     PAYROLL   1     TAPE    NOT READY 1        0
  LIST-T    0         0     0       NO DEVICE 0        0
  LABEL     MAIL      2     PORT    ASSIGNED  0        0
  SAMPLE    3         3     LPTR    ASSIGNED  0        0
  PAYABLES  1         1     LPTR    ACTIVE    1        0
  PART      2         2     LPTR    ASSIGNED  0        0
  
```

Figure C: SP-STATUS Screen Display

SPOOLER

DELETING A FORM QUEUE

A form queue may be deleted by use of either the SP-STATUS Action Code 7 (Delete Form Queue) or the SP-FQDELETE verb.

The format of the Delete Form Queue Action Code is:

```
ENTER ACTION CODE:7 <cr>  
FORM-QUEUE:
```

where "FORM-QUEUE" is prompting for the name of the queue to be deleted.

The format of the SP-FQDELETE verb is:

```
SP-FQDELETE form-queue
```

A form queue may only be deleted if it has no jobs queued for output and there are no lines assigned to it. The default form queue (STANDARD) may not be deleted.

Figures A and B illustrate the usage of the Delete Form Queue Action Code and the SP-FQDELETE verb. Figure C displays the SP-STATUS screen which would follow the execution of the example in Figure A or B.

```

:SP-STATUS <cr>
                                FORM QUEUE ASSIGNMENT

  QUEUE   FORM   DEVICE   DEVICE   DEVICE   NUMBER   PAGE
  NAME    TYPE   ASSIGNED  TYPE     STATUS   ENTRIES  SKIP

  INVOICES INVOICE   1     LPTR    ACTIVE   6        0
  CHECK    PAYROLL  1     TAPE    NOT READY 1        0
  LIST-T   0         0     0       NO DEVICE 0        0
  LABEL    MAIL     2     PORT    ASSIGNED 0        0
  SAMPLE   3         3     LPTR    ASSIGNED 0        0
  PAYABLES 0         0     LPTR    NO DEVICE 0        0
  PART     2         2     LPTR    ASSIGNED 0        0

                                ACTION CODES
  1. CREATE FORM QUEUE  4. CHANGE DEVICE ASSIGNMENT  7. DELETE FORM QUEUE
  2. CHANGE FORM QUEUE  5. CHANGE PAGE SKIP          8. PAGE EJECT
  3. CHANGE FORM TYPE   6. LIST PRINT JOBS           99. EXIT

  ENTER ACTION CODE:7 <cr>      The Delete Form Queue Action Code removes
  FORM-QUEUE-NAME: LIST-T <cr>  the LIST-T queue from the queue table.

```

Figure A: Sample Usage of Delete Form Queue

```

:SP-FQDELETE LIST-T <cr>      The SP-FQDELETE verb performs the same
:SP-STATUS <cr>              function as the action code in Figure A.

```

Figure B: Sample Usage of SP-FQDELETE Verb

```

                                FORM QUEUE ASSIGNMENT

  QUEUE   FORM   DEVICE   DEVICE   DEVICE   NUMBER   PAGE
  NAME    TYPE   ASSIGNED  TYPE     STATUS   ENTRIES  SKIP

  INVOICES INVOICE   1     LPTR    ACTIVE   6        0
  CHECK    PAYROLL  1     TAPE    NOT READY 1        0
  LABEL    MAIL     2     PORT    ASSIGNED 0        0
  SAMPLE   3         3     LPTR    ASSIGNED 0        0
  PAYABLES 0         0     LPTR    NO DEVICE 0        0
  PART     2         2     LPTR    ASSIGNED 0        0

```

Figure C: Sample SP-STATUS Screen Display

EJECTING PAGES FROM PRINTER

To alter the number of pages ejected from the printer after exhausting each form queue, use the SP-STATUS Action Code 5 (Change Page Skip) or the SP-SKIP verb. To tell an attached printer to eject a specified number of pages at your command, use SP-STATUS Action Code 8 (Page Eject) or the SP-EJECT verb.

The format of the Change Page Skip Action Code is:

```
ENTER ACTION CODE:5 <cr>  
FORM-QUEUE PAGES:
```

where:

FORM-QUEUE is prompting the name of the queue assigned to the printer.
PAGES is prompting for the number of pages to be ejected.

The format of the SP-SKIP verb is:

```
SP-SKIP form-queue pages
```

The maximum number of pages the printer may eject after exhausting a print queue is 10. Answering the "PAGES" prompt with a number greater than 10 results in the following message: TOO MANY PAGES.

The format of the Page Eject Action Code is:

```
ENTER ACTION CODE:8 <cr>  
FORM-QUEUE PAGES:
```

The format of the SP-EJECT verb is:

```
SP-EJECT form-queue pages
```

Action code 8 and the SP-EJECT verb attach the assigned printer to your line and issue a number of page ejects to that printer. (Note that page ejects are not shown on the SP-STATUS display.) The printer remains attached to the terminal until you issue a P-DET or SP-ASSIGN verb or you log off. See topic entitled PRINTER CONTROL: P-ATT and P-DET for a detailed explanation of the P-DET command.

If you attempt to eject more than 10 pages, the following message appears:

```
TOO MANY PAGES.
```

Figures A and B illustrate the usage of Change Page Skip Action Code and the SP-SKIP verb, and the SP-STATUS screen that would follow the execution of either example. Sample usage of the Page Eject Action Code and the SP-EJECT verb is shown in Figure C.

```

:SP-STATUS <cr>
                                FORM QUEUE ASSIGNMENT

  QUEUE      FORM      DEVICE      DEVICE      DEVICE      NUMBER      PAGE
  NAME       TYPE      ASSIGNED   TYPE        STATUS      ENTRIES     SKIP

  INVOICES   INVOICE    1         LPTR        ACTIVE      6           0
  CHECK      PAYROLL    1         TAPE        NOT READY   1           0

                                ACTION CODES
  1. CREATE FORM QUEUE  4. CHANGE DEVICE ASSIGNMENT  7. DELETE FORM QUEUE
  2. CHANGE FORM QUEUE  5. CHANGE PAGE SKIP          8. PAGE EJECT
  3. CHANGE FORM TYPE   6. LIST PRINT JOBS           99. EXIT

  ENTER ACTION CODE:5 <cr>      The Change Page Skip Action Code alters
  FORM-QUEUE PAGES:INVOICES 2<cr> the Page Skip function from 0 to 2.

```

Figure A: Sample Usage of Change Page Skip Action Code

```

:SP-SKIP INVOICES 2 <cr>      The SP-SKIP verb performs the same
:SP-STATUS <cr>              function as Action Code 5

                                FORM QUEUE ASSIGNMENT

  QUEUE      FORM      DEVICE      DEVICE      DEVICE      NUMBER      PAGE
  NAME       TYPE      ASSIGNED   TYPE        STATUS      ENTRIES     SKIP

  INVOICES   INVOICE    1         LPTR        ACTIVE      6           2
  CHECK      PAYROLL    1         TAPE        NOT READY   1           0

```

Figure B: Sample Usage of SP-SKIP Verb and SP-STATUS display after execution

```

  ENTER ACTION CODE:8 <cr>      The Page Eject Action Code causes the
  FORM-QUEUE PAGES:SAMPLE 4 <cr> printer assigned to SAMPLE to eject 4
                                  pages.

  :SP-EJECT SAMPLE 4 <cr>      The SP-EJECT verb performs the same
                                  function as Action Code 8

```

Figure C: Sample Usage of Page Eject Action Code and SP-EJECT Verb

SPOOLER

DISPLAYING PRINT JOB STATUS: SP-JOBS

| The SP-STATUS Action Code 6 (List Print Jobs) and the SP-JOBS verb |
display the current status of print jobs within each form queue.

The format of the List Print Jobs Action Code is:

ENTER ACTION CODE:6 <cr>

This action code evokes the same screen display as the SP-JOBS verb which is described below.

The format of the SP-JOBS verb is:

SP-JOBS

The SP-JOBS verb displays:

1. name of the form queue,
2. the print job number assigned to spooled jobs,
3. name of account from which print job was created,
4. the terminal line number from which the print job originated,
5. the date the print job was created,
6. status of the print job,
7. the options chosen,
8. the number of frames the print job occupies, and
9. number of copies to be printed.

The job status will be one of the following:

OPEN	A user process is writing to the job.
WAIT	The job is closed and waiting to be printed.
PRINT	The job is being printed.
FINISH	The job has been printed.

At any given time, more than one of the status conditions listed above may exist; however, only one status condition will be displayed. The display priority is: FINISH, PRINT, WAIT, OPEN.

If there are no print jobs assigned to any form queues and the SP-JOBS verb is executed, the message "THE QUEUE IS EMPTY" will be displayed on the screen.

The action codes immediately follow the screen display of the SP-JOBS verb as shown in Figure A.

If you wish to execute an action code, enter its corresponding number in response to the prompt "ENTER ACTION CODE / PAGE NUMBER (P#)?" . As soon as an action code is executed, an updated SP-JOBS screen is displayed.

To page through the SP-JOBS screen(s), press the RETURN key. However, if you wish to skip to a particular page of the SP-JOBS screen, enter a "P", followed by a page number. For example, if you enter a "P22" to the "ENTER ACTION CODE / PAGE NUMBER (P#)? prompt, all jobs listed on page 22 appear on the screen. Should you enter a number larger than the number of SP-JOBS pages, an error message appears and you are reprompted.

To display the SP-STATUS screen, enter Action Code 11. To exit the display screen, enter Action Code 99.

The SP-JOBS action codes are explained in the following sections. Figure A is a sample screen display which follows the execution of the SP-JOBS verb.

```

-----
:SP-JOBS <cr>
                                PRINT JOBS          PAGE 1 OF 1 PAGES
                                DATE
                                CREATED
QUEUE NAME   PRINT JOB   ACCOUNT  LINE  DATE  STATUS  OPT  SIZE  COPIES
INVOICES     1     ACT      9    12 AUG  PRINT   H   12    1
INVOICES     2     ACT      2    12 AUG  WAIT    H   15    1
INVOICES     3     MANF     3    12 AUG  OPEN    H    7    1
CHECK        4     PERS     5    12 AUG  WAIT    H   27    1
SAMPLE       5     PROD    25    12 AUG  WAIT    HS  30    1
INVOICES     6     MANF     3    12 AUG  WAIT    H   15    1
INVOICES     7     ACT      2    12 AUG  WAIT    H    9    1
INVOICES     8     MANF    10    12 AUG  WAIT    H   27    1
.            .           .       .       .       .       .       .
.            .           .       .       .       .       .       .
.            .           .       .       .       .       .       .

1. MOVE FORM QUEUE      5. DELETE JOB          9. SUSPEND PRINTING
2. MOVE PRINT JOB       6. STOP PRINTING      10. CHANGE PRIORITY
3. CHANGE OPTIONS       7. RESUME PRINT       11. SP-STATUS
4. CHANGE #OF COPIES    8. EDIT PRINT JOB     99. EXIT

ENTER ACTION CODE / PAGE NUMBER (P#)?
-----

```

Figure A: Sample SP-JOBS Screen Display

SPOOLER

MOVING A FORM QUEUE

| To transfer all print jobs in one queue to a different queue, enter the |
SP-JOBS Action Code 1 (Move Form Queue) or the SP-MOVEQ verb.

The general form of the Move Form Queue Action Code is:

```
ENTER ACTION CODE / PAGE NUMBER (P#)?1 <cr>  
FROM-FORM-QUEUE TO-FORM-QUEUE:  
PASSWORD:
```

where:

```
FROM-FORM-QUEUE is prompting for the name of the current form queue.  
TO-FORM-QUEUE  is prompting for the name of the form queue accepting  
                the print jobs.  
PASSWORD:      is prompting for the password of the account which  
                created the form queue. If the account has no  
                password or the password is the same as the current  
                account, you are not prompted for the password.
```

The format of the SP-MOVEQ verb is:

```
SP-MOVEQ from-form-queue to-form-queue
```

If the printing of a form queue has been "killed" by the SP-KILL verb before the execution of the Move Form Queue Action Code or the SP-MOVEQ verb, the interrupted job and all subsequent jobs in the queue will be moved.

NOTE: Print jobs having a status of "FINISH" are not moved. Use Action Code 2 (Move Print Job) or SP-SWITCH to move such a job to another form queue (see the topic "MOVING PRINT JOBS" for more information).

Figures A and B illustrate the use of the Move Form Queue Action Code and SP-MOVEQ verb. Figure C displays the SP-JOBS screen which would follow the execution of the example in Figure A or Figure B.

```

:SP-JOBS <cr>
                                PRINT JOBS          PAGE 1 OF 1 PAGES

QUEUE NAME      PRINT          ACCOUNT  LINE  DATE          STATUS  OPT  SIZE  COPIES
JOB

INVOICES        1      ACT       9    12 AUG  PRINT    H   12   1
INVOICES        2      ACT       2    12 AUG  WAIT     H   15   1
INVOICES        3      MANF      3    12 AUG  OPEN     H    7   1
CHECK           4      PERS      5    12 AUG  WAIT     H   27   1
SAMPLE          5      PROD     25   12 AUG  WAIT     HS  30   1
INVOICES        6      MANF      3    12 AUG  WAIT     H   15   1
INVOICES        7      ACT       2    12 AUG  WAIT     H    9   1
INVOICES        8      MANF     10   12 AUG  WAIT     H   27   1

1. MOVE FORM QUEUE      5. DELETE JOB          9. SUSPEND PRINTING
2. MOVE PRINT JOB       6. STOP PRINTING      10. CHANGE PRIORITY
3. CHANGE OPTIONS       7. RESUME PRINT       11. SP-STATUS
4. CHANGE #OF COPIES   8. EDIT PRINT JOB     99. EXIT

ENTER ACTION CODE / PAGE NUMBER (P#)?1<cr> The Move Form Queue Action
FROM-FORM-QUEUE TO-FORM-QUEUE:              Code moves all print jobs
SAMPLE CHECK <cr>                            from form queue SAMPLE to
PASSWORD:TAB <cr>                            CHECK.

```

Figure A: Sample Usage of Move Form Queue Action Code

```

:SP-MOVEQ SAMPLE CHECK <cr>      This SP-MOVEQ verb performs the same
PASSWORD:TAB <cr>                 function as Action Code 1.
:SP-JOBS <cr>

```

Figure B: Sample Usage of SP-MOVEQ Verb

```

                                PRINT JOBS          PAGE 1 OF 1 PAGES

QUEUE NAME      PRINT          ACCOUNT  LINE  DATE          STATUS  OPT  SIZE  COPIES
JOB

INVOICES        1      ACT       9    12 AUG  PRINT    H   12   1
INVOICES        2      ACT       2    12 AUG  WAIT     H   15   1
INVOICES        3      MANF      3    12 AUG  OPEN     H    7   1
CHECK           4      PERS      5    12 AUG  WAIT     H   27   1
CHECK           5      PROD     25   12 AUG  WAIT     HS  30   1
INVOICES        6      MANF      3    12 AUG  WAIT     H   15   1
INVOICES        7      ACT       2    12 AUG  WAIT     H    9   1
INVOICES        8      MANF     10   12 AUG  WAIT     H   27   1

```

Figure C: SP-JOBS Screen Display

SPOOLER

MOVING PRINT JOBS

| The SP-JOBS Action Code 2 (Move Print Job) and the SP-SWITCH verb |
transfer one print job or a list of print jobs to another queue.

The general form of the Move Print Job Action Code is:

```
ENTER ACTION CODE / PAGE NUMBER (P#)?2 <cr>  
NEW-FORM-QUEUE PRINT-JOB-LIST:  
PASSWORD:
```

where:

```
NEW-FORM-QUEUE    is prompting for the name of the form queue accepting  
                  the print job(s).  
PRINT-JOB-LIST   is prompting for the number of each print job to be  
                  moved. Each print job number is separated by one  
                  space.
```

The format of the SP-SWITCH verb is:

```
SP-SWITCH new-form-queue print-job-list
```

A job currently being printed cannot be moved to a new queue unless the Suspend Printing Action Code or the SP-SUSPEND verb has been issued to the current form queue. When the job is moved, it will be reprinted from the beginning of the job.

Figures A and B illustrate the use of the Move Print Job Action Code and the SP-SWITCH verb. Figure C displays the SP-JOBS screen which would follow the execution of the example in Figure A or B.

```

:SP-JOBS <cr>
                                PRINT JOBS          PAGE 1 OF 1 PAGES

QUEUE NAME      PRINT          ACCOUNT  LINE  DATE          STATUS  OPT  SIZE  COPIES
JOB

INVOICES        1          ACT       9    12 AUG        PRINT   H   12   1
INVOICES        2          ACT       2    12 AUG        WAIT    H   15   1
INVOICES        3          MANF      3    12 AUG        OPEN    H    7   1
CHECK           4          PERS      9    12 AUG        WAIT                    27   1
CHECK           5          PROD     25   12 AUG        WAIT    HS  30   1
INVOICES        6          MANF      3    12 AUG        WAIT    H   15   1
INVOICES        7          ACT       2    12 AUG        WAIT    H    9   1
INVOICES        8          MANF     10   12 AUG        WAIT    H   27   1

1. MOVE FORM QUEUE          5. DELETE JOB              9. SUSPEND PRINTING
2. MOVE PRINT JOB           6. STOP PRINTING          10. CHANGE PRIORITY
3. CHANGE OPTIONS           7. RESUME PRINT           11. SP-STATUS
4. CHANGE #OF COPIES       8. EDIT PRINT JOB        99. EXIT

ENTER ACTION CODE / PAGE NUMBER (P#)?2<cr> The Move Print Job Action Code
NEW-FORM-QUEUE PRINT-JOB-LIST:PART 2<cr> moves print job #2 in
                                           INVOICES form queue to PART
                                           form queue (previously
                                           created by SP-STATUS Action
                                           Code 1).

```

Figure A: Sample Usage of Move Print Job Action Code

```

:SP-SWITCH PART 2 <cr>          This SP-SWITCH verb performs the same
:SP-JOBS <cr>                  function as Action Code 2.

```

Figure B: Sample Usage of SP-SWITCH Verb

```

                                PRINT JOBS          PAGE 1 OF 1 PAGES

QUEUE NAME      PRINT          ACCOUNT  LINE  DATE          STATUS  OPT  SIZE  COPIES
JOB

INVOICES        1          ACT       9    12 AUG        PRINT   H   12   1
PART            2          ACT       2    12 AUG        PRINT   H   15   1
INVOICES        3          MANF      3    12 AUG        OPEN    H    7   1
CHECK           4          PERS      9    12 AUG        WAIT                    27   1
CHECK           5          PROD     25   12 AUG        WAIT    HS  30   1
INVOICES        6          MANF      3    12 AUG        WAIT    H   15   1
INVOICES        7          ACT       2    12 AUG        WAIT    H    9   1
INVOICES        8          MANF     10   12 AUG        WAIT    H   27   1

```

Figure C: SP-JOBS Screen Display

SPOOLER

CHANGING PRINT JOB OPTIONS

| To change print job options, use the SP-JOBS Action Code 3 (Change
Options) or the SP-OPTS verb.

The general form of the Change Options Action Code is:

```
ENTER ACTION CODE / PAGE NUMBER (P#)?3 <cr>  
PRINT-JOB#  OPTIONS:
```

where:

```
PRINT-JOB#      is prompting for the number of the print job.  
OPTIONS         is prompting for one or both of the symbols used to  
                indicate those options that can be changed.
```

The format of the SP-OPTS verb is:

```
SP-OPTS print-job# options
```

By using this verb, previous options are deleted and new options are assigned to the print job. The option that may be changed are 'H' (hold print job until deleted) and 'S' (suppress output).

The options of a currently outputting job cannot be changed. If the "OPTIONS" prompt is answered with a <cr> only, the previous options are cancelled and the print job is left with no options.

Figure A and B illustrate the use of the Change Options Action Code and the SP-OPTS verb. Figure C displays the SP-JOBS screen that would follow the execution of the example in Figure A or B.

```

:SP-JOBS <cr>
                                PRINT JOBS          PAGE 1 OF 1 PAGES

QUEUE NAME      PRINT          ACCOUNT  LINE  DATE          STATUS  OPT  SIZE  COPIES
JOB

INVOICES        1      ACT       9    12 AUG    PRINT   H   12   1
PART            2      ACT       2    12 AUG    PRINT   H   15   1
INVOICES        3      MANF      3    12 AUG    OPEN    H    7   1
CHECK           4      PERS      9    12 AUG    WAIT                   27   1
CHECK           5      PROD     25    12 AUG    WAIT    HS   30   1
INVOICES        6      MANF      3    12 AUG    WAIT    H   15   1
INVOICES        7      ACT       2    12 AUG    WAIT    H    9   1
INVOICES        8      MANF     10    12 AUG    WAIT    H   27   1

1. MOVE FORM QUEUE      5. DELETE JOB          9. SUSPEND PRINTING
2. MOVE PRINT JOB       6. STOP PRINTING      10. CHANGE PRIORITY
3. CHANGE OPTIONS       7. RESUME PRINT       11. SP-STATUS
4. CHANGE #OF COPIES   8. EDIT PRINT JOB     99. EXIT

ENTER ACTION CODE / PAGE NUMBER (P#)?3<cr> The Change Options Action
PRINT-JOB#  OPTIONS:4 HS <cr>              Code adds the 'H' and 'S'
                                           options to print job #4.

```

Figure A: Sample Usage of Change Options Action Code

```

:SP-OPTS 4 HS <cr>          This SP-OPTS verb performs the same
:SP-JOBS <cr>              function as Action Code 3 in Figure A.

```

Figure B: Sample Usage of SP-OPTS Verb

```

                                PRINT JOBS          PAGE 1 OF 1 PAGES

QUEUE NAME      PRINT          ACCOUNT  LINE  DATE          STATUS  OPT  SIZE  COPIES
JOB

INVOICES        1      ACT       9    12 AUG    PRINT   H   12   1
PART            2      ACT       2    12 AUG    PRINT   H   15   1
INVOICES        3      MANF      3    12 AUG    OPEN    H    7   1
CHECK           4      PERS      9    12 AUG    WAIT                   27   1
CHECK           5      PROD     25    12 AUG    WAIT    HS   30   1
INVOICES        6      MANF      3    12 AUG    WAIT    H   15   1
INVOICES        7      ACT       2    12 AUG    WAIT    H    9   1
INVOICES        8      MANF     10    12 AUG    WAIT    H   27   1

```

Figure C: SP-JOBS Screen Display

CHANGING NUMBER OF PRINT JOB COPIES

| Use the SP-JOBS Action Code 4 (Change Number of Copies) or the SP-COPIES |
verb to alter the number of times a print job is printed.

The general form of the Change Number of Copies Action Code is:

```
ENTER ACTION CODE / PAGE NUMBER (P#)?4 <cr>  
PRINT-JOB# COPIES:
```

where:

```
PRINT-JOB#      is prompting for the number of the print job whose  
                "copies" parameter is to be changed.  
COPIES          is prompting for the number of copies (1 - 100) to be  
                printed.
```

The format of the SP-COPIES verb is:

```
SP-COPIES print-job-# copies
```

Execution of either of these two methods replaces the current "copies" parameter of a job waiting to be printed or a hold file that has been printed. The "copies" parameter of a currently printing job cannot be changed.

Figures A and B illustrate the use of the Change Number of Copies Action Code and the SP-COPIES verb. Figure C displays the SP-JOBS screen which would follow the execution of the example in Figure A or B.

```

-----
:SP-JOBS <cr>
                                PRINT JOBS          PAGE 1 OF 1 PAGES

QUEUE NAME      PRINT      ACCOUNT  LINE  DATE      STATUS  OPT  SIZE  COPIES
JOB

INVOICES        1      ACT      9    12 AUG    PRINT   H   12   1
PART            2      ACT      2    12 AUG    PRINT   H   15   1
INVOICES        3      MANF     3    12 AUG    OPEN    H    7   1
CHECK           4      PERS     9    12 AUG    WAIT    HS   27   1
CHECK           5      PROD    25    12 AUG    WAIT    HS   30   1
INVOICES        6      MANF     3    12 AUG    WAIT    H   15   1
INVOICES        7      ACT      2    12 AUG    WAIT    H    9   1
INVOICES        8      MANF    10    12 AUG    WAIT    H   27   1

1. MOVE FORM QUEUE      5. DELETE JOB          9. SUSPEND PRINTING
2. MOVE PRINT JOB       6. STOP PRINTING      10. CHANGE PRIORITY
3. CHANGE OPTIONS       7. RESUME PRINT       11. SP-STATUS
4. CHANGE #OF COPIES   8. EDIT PRINT JOB    99. EXIT

ENTER ACTION CODE / PAGE NUMBER (P#)?4<cr> This Change #of Copies Action
PRINT-JOB# COPIES:4 3 <cr>                Code deletes the "1" copy
                                           designation for print job #4
                                           and replaces it with "3".
-----

```

Figure A: Sample Usage of Change Number of Copies Action Code

```

-----
:SP-COPIES 4 3 <cr>          This SP-COPIES verb performs the same
:SP-JOBS <cr>                function as Action Code 4.
-----

```

Figure B: Sample Usage of SP-COPIES Verb

```

-----
                                PRINT JOBS          PAGE 1 OF 1 PAGES

QUEUE NAME      PRINT      ACCOUNT  LINE  DATE      STATUS  OPT  SIZE  COPIES
JOB

INVOICES        1      ACT      9    12 AUG    PRINT   H   12   1
PART            2      ACT      2    12 AUG    PRINT   H   15   1
INVOICES        3      MANF     3    12 AUG    OPEN    H    7   1
CHECK           4      PERS     9    12 AUG    WAIT    HS   27   3
CHECK           5      PROD    25    12 AUG    WAIT    HS   30   1
INVOICES        6      MANF     3    12 AUG    WAIT    H   15   1
INVOICES        7      ACT      2    12 AUG    WAIT    H    9   1
INVOICES        8      MANF    10    12 AUG    WAIT    H   27   1
-----

```

Figure C: SP-JOBS Screen Display

DELETING PRINT JOBS FROM A QUEUE

| The SP-JOBS Action Code 5 (Delete Job) or the SP-DELETE verb removes |
print jobs from a form queue.

The general form of the Delete Job Action Code is:

```
ENTER ACTION CODE / PAGE NUMBER (P#)?5 <cr>  
PRINT-JOB-LIST:  
PASSWORD:
```

where:

```
PRINT-JOB-LIST:  Prompts for the print job number(s) to be deleted.  
                  Each number is separated by one space.  
  
PASSWORD:       Prompts for the password of the account from  
                  which the print job was created (this prompt  
                  is only displayed if a password is required).
```

The format of the SP-DELETE verb is:

```
SP-DELETE print-job-list  
PASSWORD:
```

After deleting a job, the overflow frames are returned to the system. A currently outputting print job may be deleted only if all queue output is killed and the queue is inactive. An open print job cannot be deleted until the job is closed and the job is placed in a "WAIT" or "FINISH" state.

If all print jobs do not have the same password, you will be prompted when a new password is needed.

Figures A and B illustrate the use of the Delete Job Action Code and the SP-DELETE verb. Figure C displays the SP-JOBS screen which would follow the execution of the example in Figure A or B.

```

:SP-JOBS <cr>
                                PRINT JOBS          PAGE 1 OF 1 PAGES

QUEUE NAME      PRINT          ACCOUNT  LINE  DATE      STATUS  OPT  SIZE  COPIES
JOB

INVOICES        1      ACT       9    12 AUG   PRINT   H   12   1
PART            2      ACT       2    12 AUG   PRINT   H   15   1
INVOICES        3      MANF      3    12 AUG   OPEN    H   7    1
CHECK           4      PERS      9    12 AUG   WAIT    HS  27   1
CHECK           5      PROD     25   12 AUG   WAIT    HS  30   1
INVOICES        6      MANF      3    12 AUG   WAIT    H   15   1
INVOICES        7      ACT       2    12 AUG   WAIT    H   9    1
INVOICES        8      MANF     10   12 AUG   WAIT    H   27   1

1. MOVE FORM QUEUE      5. DELETE JOB          9. SUSPEND PRINTING
2. MOVE PRINT JOB       6. STOP PRINTING      10. CHANGE PRIORITY
3. CHANGE OPTIONS       7. RESUME PRINT       11. SP-STATUS
4. CHANGE #OF COPIES   8. EDIT PRINT JOB     99. EXIT

ENTER ACTION CODE / PAGE NUMBER (P#)?5<cr> This Delete Job Action Code
PRINT-JOB-LIST:5 <cr>                       eliminates print job #5
PASSWORD:TPH <cr>                            found in the CHECK form queue
                                                and returns the 30 frames the
                                                job used to overflow.

```

Figure A: Sample Usage of Delete Job Action Code

```

:SP-DELETE 5 <cr>           This SP-DELETE verb performs the same
:SP-JOBS <cr>              function as Action Code 5.

```

Figure B: Sample Usage of SP-DELETE Verb

```

                                PRINT JOBS          PAGE 1 OF 1 PAGES

QUEUE NAME      PRINT          ACCOUNT  LINE  DATE      STATUS  OPT  SIZE  COPIES
JOB

INVOICES        1      ACT       9    12 AUG   PRINT   H   12   1
PART            2      ACT       2    12 AUG   PRINT   H   15   1
INVOICES        3      MANF      3    12 AUG   OPEN    H   7    1
CHECK           4      PERS      9    12 AUG   WAIT    HS  27   1
INVOICES        6      MANF      3    12 AUG   WAIT    H   15   1
INVOICES        7      ACT       2    12 AUG   WAIT    H   9    1
INVOICES        8      MANF     10   12 AUG   WAIT    H   27   1

```

Figure C: SP-JOBS Screen Display

SPOOLER

STOPPING THE PRINTER BEFORE OR AFTER PRINTING A JOB

The SP-JOBS Action Code 6 (Stop Printing) and the SP-STOP verb may be used to stop the printer just after the printing of the current print job or just before the printing of a specified print job.

The general form of the Stop Printing Action Code is:

```
ENTER ACTION CODE / PAGE NUMBER (P#)?6 <cr>  
FORM-QUEUE OR PRINT-JOB#:
```

where:

FORM-QUEUE is prompting for the name of the queue whose printing will be halted.

PRINT-JOB# is prompting for the number of the print job whose printing will be delayed.

The format of the SP-STOP verb is:

```
SP-STOP form-queue or print-job#
```

If you wish to stop the printer just after the current job is printed, enter the name of the active form queue. Should you decide to stop the printer before printing a specified job, enter the number of that print job.

When you stop the printer prior to printing a specific job, all jobs in the queue will be printed until the specified job is encountered.

Figures A and B illustrate the use of the Stop Printing Action Code and the SP-STOP verb. Figure C displays the SP-STATUS display that reflects the interruption to the INVOICES form queue.

```

:SP-JOBS <cr>
                                PRINT JOBS          PAGE 1 OF 1 PAGES

QUEUE NAME      PRINT      ACCOUNT  LINE  DATE      STATUS  OPT  SIZE  COPIES
JOB

INVOICES        1      ACT      9    12 AUG    PRINT   H   12   1
PART            2      ACT      2    12 AUG    PRINT   H   15   1
INVOICES        3      MANF     3    12 AUG    OPEN    H    7   1
CHECK           4      PERS     9    12 AUG    WAIT    HS  27   1
INVOICES        6      MANF     3    12 AUG    WAIT    H   15   1
INVOICES        7      ACT      2    12 AUG    WAIT    H    9   1
INVOICES        8      MANF    10    12 AUG    WAIT    H   27   1

1. MOVE FORM QUEUE      5. DELETE JOB          9. SUSPEND PRINTING
2. MOVE PRINT JOB       6. STOP PRINTING      10. CHANGE PRIORITY
3. CHANGE OPTIONS       7. RESUME PRINT       11. SP-STATUS
4. CHANGE #OF COPIES   8. EDIT PRINT JOB    99. EXIT

ENTER ACTION CODE / PAGE NUMBER (P#)?6<cr> This Stop Printing Action
FORM QUEUE OR PRINT-JOB#:INVOICES <cr>      Code stops the printer
                                              prior to printing job #6.

```

Figure A: Sample Usage of Stop Printing Action Code

```

:SP-STOP INVOICES <cr>          This SP-STOP verb performs the same
                                function as Action Code 6.

```

Figure B: Sample Usage of SP-STOP Verb

```

                                FORM QUEUE ASSIGNMENT

QUEUE      FORM      DEVICE      DEVICE      DEVICE      NUMBER      PAGE
NAME       TYPE      ASSIGNED   TYPE        STATUS      ENTRIES     SKIP

INVOICES   INVOICE   1          LPTR        STOP        4           2
CHECK      PAYROLL   1          TAPE        NOT READY   0           0
LABEL      MAIL      2          PORT        ASSIGNED    0           0
SAMPLE     3          LPTR        ASSIGNED    0           0
PAYABLES   0          LPTR        NO DEVICE   0           0
PART       2          LPTR        ACTIVE      1           0

```

Figure C: SP-STATUS Screen Display

RESTARTING A PRINTER AFTER A FORCED WAIT

Restarting a printer after a forced wait is accomplished by the SP-JOBS
Action Code 7 (Resume Printing) or the SP-RESUME verb.

The general form of the Resume Printing Action Code is:

ENTER ACTION CODE / PAGE NUMBER (P#)?7 <cr>
FORM-QUEUE:

where:

FORM-QUEUE: is prompting for the name of the interrupted form
queue.

The format of the SP-RESUME verb is:

SP-RESUME form-queue

One of these two methods must be used after correcting the problem responsible for a device status of NEXT REEL, PARITY or CABLE OFF, and may be used after the execution of Suspend Printing and Stop Printing Action Codes or SP-SUSPEND and SP-STOP verbs.

When the SP-RESUME or the Resume Printing Action Code is executed, printing will begin where the spooler left off.

Figures A and B illustrate the use of the Stop Printing Action Code and the SP-RESUME verb. Figure C displays the SP-STATUS screen which reflects the "ACTIVE" status of form queue INVOICES.

```

:SP-JOBS <cr>
                                PRINT JOBS          PAGE 1 OF 1 PAGES

QUEUE NAME      PRINT      ACCOUNT  LINE  DATE      STATUS  OPT  SIZE  COPIES
INVOICES        1         ACT      9    12 AUG    FINISH   H   12    1
PART            2         ACT      2    12 AUG    PRINT    H   15    1
INVOICES        3         MANF     3    12 AUG    OPEN     H    7    1
CHECK           4         PERS     9    12 AUG    WAIT     HS   27    1
INVOICES        6         MANF     3    12 AUG    WAIT     H   15    1
INVOICES        7         ACT      2    12 AUG    WAIT     H    9    1
INVOICES        8         MANF    10    12 AUG    WAIT     H   27    1

1. MOVE FORM QUEUE      5. DELETE JOB          9. SUSPEND PRINTING
2. MOVE PRINT JOB       6. STOP PRINTING      10. CHANGE PRIORITY
3. CHANGE OPTIONS       7. RESUME PRINT       11. SP-STATUS
4. CHANGE #OF COPIES   8. EDIT PRINT JOB    99. EXIT

ENTER ACTION CODE / PAGE NUMBER (P#)?7<cr> This Resume Printing Action
FORM-QUEUE:INVOICES <cr>                  Code starts the printing of
                                           job #6 which was interrupted
                                           by Action Code 6 in previous
                                           section.

```

Figure A: Sample Usage of Resume Printing Action Code.

```

:SP-RESUME INVOICES <cr>          This SP-RESUME verb performs the same
                                function as Action Code 7.

```

Figure B: Sample Usage of SP-RESUME Verb

```

:SP-STATUS <cr>
                                FORM QUEUE ASSIGNMENT

QUEUE      FORM      DEVICE      DEVICE      DEVICE      NUMBER      PAGE
NAME       TYPE      ASSIGNED    TYPE        STATUS      ENTRIES     SKIP
INVOICES   INVOICE   1          LPTR        ACTIVE      4           2
CHECK      PAYROLL   1          TAPE        NOT READY   0           0
LABEL      MAIL      2          PORT        ASSIGNED    0           0
SAMPLE                                3          LPTR        ASSIGNED    0           0
PAYABLES   0         LPTR        NO DEVICE   0           0
PART       2         LPTR        ASSIGNED    1           0

```

Figure C: SP-STATUS Screen Display

SPOOLER

INTERRUPTING A PRINTING JOB

The SP-JOBS Action Code 9 (Suspend Printing) and the SP-SUSPEND verb interrupt the printing of the current print job but do not delete the job from the queue. The SP-KILL verb also halts the output of the current print job and deletes that job if it is not a hold file.

The format of the Suspend Printing Action Code is:

```
ENTER ACTION CODE / PAGE NUMBER (P#)?9 <cr>  
FORM-QUEUE --OR-- DEVICE# TYPE:
```

where the following responses are required:

FORM-QUEUE	the name of the interrupted form queue. If you enter the form queue name, do not enter device# and type.
DEVICE# TYPE	the device number and type of assigned device. The form queue name is not necessary if you enter the device number and type.

The format of the SP-SUSPEND verb is:

```
SP-SUSPEND form-queue --or-- device# and type
```

When either of these two methods is initiated, the interruption to the printing job is immediate. To restart printing where the spooler left off, use Action Code 7 (Resume Printing) or the SP-RESUME verb. To resume printing at any location you wish, issue the 'P' command of the Edit Print Job Action Code and the SP-EDIT verb.

The format of the SP-KILL verb is:

```
SP-KILL form-queue --or-- device# and type
```

When the SP-KILL verb is executed, the currently printing job is halted and the next job in the queue begins printing immediately. If the interrupted job is a hold file, it will remain in the job list; if the job is not a hold file, it will be deleted.

Figures A and B illustrate the use of the Suspend Printing Action Code and the SP-SUSPEND verb. Figure C displays the SP-STATUS screen which would follow the execution of the example in Figure A or B. An SP-JOBS screen is not shown here because the actual queue status can only be seen on the SP-STATUS display.

```

:SP-JOBS <cr>
                                PRINT JOBS          PAGE 1 OF 1 PAGES

QUEUE NAME      PRINT      ACCOUNT  LINE  DATE      STATUS  OPT  SIZE  COPIES
JOB

INVOICES        1      ACT      9    12 AUG    PRINT   H   12   1
PART            2      ACT      2    12 AUG    PRINT   H   15   1
INVOICES        3      MANF     3    12 AUG    OPEN    H    7   1
CHECK           4      PERS     9    12 AUG    WAIT    HS  27   1
INVOICES        6      MANF     3    12 AUG    WAIT    H   15   1
INVOICES        7      ACT      2    12 AUG    WAIT    H    9   1
INVOICES        8      MANF    10    12 AUG    WAIT    H   27   1

1. MOVE FORM QUEUE      5. DELETE JOB          9. SUSPEND PRINTING
2. MOVE PRINT JOB       6. STOP PRINTING      10. CHANGE PRIORITY
3. CHANGE OPTIONS       7. RESUME PRINT       11. SP-STATUS
4. CHANGE #OF COPIES   8. EDIT PRINT JOB    99. EXIT

ENTER ACTION CODE / PAGE NUMBER (P#)?9<cr>      This Suspend Printing
FORM QUEUE --OR-- DEVICE# TYPE: INVOICES          Action Code interrupts
                                                    the printing of print
                                                    job #6.

```

Figure A: Sample Usage of Suspend Printing Action Code

```

:SP-SUSPEND 1 LPTR <cr>      This SP-SUSPEND verb performs the same
                                function as Action Code 9 in Figure A.

```

Figure B: Sample Usage of SP-COPIES Verb

```

                                FORM QUEUE ASSIGNMENT

QUEUE      FORM      DEVICE      DEVICE      DEVICE      NUMBER      PAGE
NAME       TYPE      ASSIGNED    TYPE        STATUS      ENTRIES     SKIP

INVOICES   INVOICE   1          LPTR        SUSPEND     4           2
CHECK      PAYROLL   1          TAPE        NOT READY   0           0
LABEL      MAIL      2          PORT        ASSIGNED    0           0
SAMPLE     3          LPTR        ASSIGNED    0           0
PAYABLES   0          LPTR        NO DEVICE   0           0
PART       2          LPTR        ACTIVE      1           0

```

Figure C: SP-STATUS Screen Display

SPOOLER

EXAMINING JOBS WAITING TO BE PRINTED

The SP-JOBS Action Code 8 (Edit Print Job) and the SP-EDIT verb manipulate a print job for display, forms alignment and/or printing a previously interrupted print job. The SP-EDIT verb may also be used to copy a spool job to an ordinary file item.

The general form of the Edit Print Job Action Code is:

```
ENTER ACTION CODE / PAGE NUMBER (P#)?8 <cr>  
PRINT-JOB#:  
PASSWORD:  
*ENTER COMMAND:
```

where the following responses are required:

PRINT-JOB#: the number of the print job to be edited.

PASSWORD: the password of the account from which the job was created. If you do not know the password, press RETURN until you get back to TCL.

*ENTER COMMAND: the spooler editor command to be executed, as described in the following topics.

The format of the SP-EDIT verb is:

```
SP-EDIT {print-job#} {(options)}
```

where options may include one or more of:

- I - Copy print job to ordinary file item. In the absence of other options, all CRs (carriage return: X'0D') and pad characters (X'00') are deleted, LFs (line feed : X'0A') are converted to AMs (attribute mark : X'FE'), and the item is truncated at 32K bytes.
Other options are only significant in conjunction with the 'I' option
- O - Overwrite existing file item.
- P - Printable: all characters outside range X'7F' to X'20' converted to '~' (tilde : X'7E') followed by two hex character representation of converted character, except LFs converted to AMs.
- U - Unconverted: spooler item copied without change, except SMs (segment mark: X'FF') to '_' (X'5F').
- A - All: a spooler job exceeding 32K bytes is divided into items of 32K bytes or fewer, copied to sequential item-ids starting from id.001 (where 'id' is item-id specified).

n - limit size of item(s) generated to 'n' bytes ('n' must be in range 4 to 32211 (which is the default = 32267-2 (AMs) - 4 (countfield) - 50 (item-id)).

After entry of the command, the print job number is prompted for if omitted. A password is then prompted for if the spool job was generated from another account, which requires a password.

If the 'I' option was specified, a prompt is displayed for the destination of the copied spool job:

File and Item ID:

and the spool job is then copied to the item specified (if the item already exists, an error message is displayed and the item is not copied, unless the 'O' option was also specified) in the format as defined by the options specified.

If the 'I' option was not specified the following prompt is displayed:

*ENTER COMMAND:

One of the spooler editor commands, detailed in the following topics, should be entered to this prompt. Figures A and B illustrate the use of the Edit Print Job Action Code and the SP-EDIT verb.

```

-----
ENTER ACTION CODE / PAGE NUMBER (P#)?8 <cr>   The Edit Print Job Action
PRINT-JOB#:4 <cr>                               Code displays the first
PASSWORD:TECS <cr>                             22 lines of print job #4.
*ENTER COMMAND:L22 <cr>
-----

```

Figure A: Sample Usage of Edit Print Job Action Code

```

-----
:SP-EDIT (IOUA200<cr>
PRINT-JOB#:4<cr>
File and Item ID:TEMP ITEM 1
'ITEM 1.008' FILED
-----

```

Figure B: Sample Usage of SP-EDIT Verb

SPOOLER

EDIT PRINT JOB COMMANDS: SUMMARY

| The spooler editor accepts a limited range of commands which allow the |
print job to be inspected and printed but not changed.

Use of Action Code 8 of the SP-JOBS menu, or the SP-EDIT verb without options, displays the prompt:

*ENTER COMMAND:

One of the spooler editor commands, detailed in the following pages, must be entered to this prompt. These commands are:

L{n}	List "n" lines.
L{n}"string"	Locate "string" in next "n" lines.
P	Print to the end of the job.
B{m}	Backward position "m" pages.
F{m}	Forward position "m" pages.
A{n}	Alignment of "n" lines in masked form.
T	Top of report.
EX	Exit.

Figure A shows some examples of the use of these commands.

EDIT PRINT JOB COMMANDS: LIST AND LOCATE

| The following two commands list and locate specified lines within a |
print job.

LIST "n" LINES

The general form of this command is:

L{n}

where "n" specifies number of lines.

This command lists a specified number of lines beginning from the current line position displayed on the terminal. If "n" is not specified, only one line is displayed on the screen, and the current print position is set at the next line.

Should a listing display more than one page, each "TOP OF FORM" will be seen on the screen.

Note: You should be aware of the current print position because subsequent EDIT commands are based on this position.

LOCATE A STRING

The general form of this command is:

L{n}"string"

where:

n specifies the number of lines to be searched.
string specifies the string to be located.

This command changes the current print position to the line that contains the character string specified in the command. If "n" is omitted, the entire print job is searched. If "n" is entered, the search is conducted within the specified number of lines. Any non-numeric delimiter may be used (where double quotes are shown above) to specify the character string as long as the delimiter itself is not part of the string.

```

-----
ENTER ACTION CODE / PAGE NUMBER (P#)?8 <cr>
PRINT-JOB-#:4 <cr>
*ENTER COMMAND:L10 <cr> <-----

```

Print Job #4 is about to be edited.
The first 10 lines will be listed on the screen.

```

PAGE 1 07/04/85
H.S. Smith
78129
F.D. Brown
59217
D.D. Silver
61298
R.M. Henderson
71298
G.F. Jones
.L5"SMITH" <cr> <-----
H.E. SMITH
R.M. SMITH
>L5 <cr> <-----
43710
P.S. MENDELSON
51367
H.T. WELLS
71300
.L10"CORTEZ" <cr> <-----
W.S. CORTEZ

```

The next 5 lines will be searched for the string "SMITH".
The next 5 lines which follow "R.M. SMITH" will be displayed.

The next 10 lines will be searched for the string "CORTEZ". The current print position is now set at the 31st line.

```

-----

```

Figure A: Sample Usage of the List and Locate Commands.

EDIT PRINT JOB COMMANDS: BACKWARD, FORWARD POSITIONING AND TOP

| The Backward Positioning command moves the print position back a
| specified number of pages; the Forward Positioning command moves the
| print position forward a specified number of pages; the Top of Print Job
command sets the print position at the top of the current job.

BACKWARD POSITIONING

The general form of this command is:

B{m}

where "m" specifies the number of pages.

This command controls the current print position. Positioning is set "m" pages backward from the current print position. If "m" is omitted, positioning is set at the top of the current print page. If "m" is greater than the number of previous pages, the position is set to the top of the print job.

FORWARD POSITIONING

The general form if this command is:

F{m}

where "m" specifies the number of pages.

This command moves the current print position forward "m" pages. If "m" is omitted, the position is moved to the top of the next print page. If "m" is greater than the number of remaining pages, the print position is set at the end of the job (EOF).

TOP OF PRINT JOB

The general form of this command is:

T

This command positions the current print position to the top of the print job. A "TOP" message is displayed following the execution of this command.

<pre> ENTER ACTION CODE / PAGE NUMBER (P#)?8 <cr> PRINT-JOB-#:7 <cr> *ENTER COMMAND:L10 <cr> PAGE 1 07/04/85 H.S. SMITH 78129 F.D. BROWN 59217 D.D. SILVER 61298 R.M. HENDERSON 71298 G.O. JONES .F2 <cr> .L1 <cr> PAGE 3 07/04/85 .B2 <cr> .L1 <cr> PAGE 2 07/04/85 .L"PAGE 3" <cr> PAGE 3 07/04/85 .B <cr> .L1 <cr> PAGE 3 07/04/85 .F3 <cr> .L1 <cr> PAGE 6 07/04/85 .T <cr> TOP </pre>	<pre> Print Job #7 is about to be edited. The first 10 lines of print job #7 are listed. Current line position is moved to top of Page 3. The first line of Page 3 is listed. Current line position is set at the top of Page 2. First line of Page 2 is listed. Current line position is set at the line following the string "PAGE 3". Current line position is rolled back to the top of Page 3. Current line position is rolled forward 3 pages. First line of Page 6 is listed. Current line position is set to top of job. </pre>
---	--

Figure A: Sample Use of Backward and Forward Positioning and Top Commands.

SPOOLER

EDIT PRINT JOB COMMANDS: ALIGNMENT, PRINT AND EXIT

The Alignment command outputs to a printer the specified number of lines in masked form; the Print command begins output at the current line position and printing continues until the entire job is printed; the Exit command leaves the print job entirely.

ALIGNMENT

The general form of this command is:

A{n}

where "n" specifies the number of lines to be aligned on a printer. If "n" is not specified, the entire page is printed in masked form. Note that this command cannot be sent to a terminal port.

Masking is performed as follows:

X represents any alphanumeric character
9 represents any numeric character
 All punctuation characters remain the same.

The current line position is set at the next line.

PRINT LINES

The format of this command is:

P

This command enables you to print a job. Printing begins at the current line position; therefore, you must first establish the current line position by using a List, Locate, Top, Forward or Backward command. Printing will then continue until the entire job is printed.

This command may also be used to restart a "suspended" job. The Editor is automatically exited by the Print command and the suspended status of the queue is cleared.

EXIT

The EXIT command format is:

EX

This command allows you to exit from the print job.

Figure A illustrates the use of Alignment, Print and Exit Commands. Figure B displays the SP-STATUS screen that would appear after restarting print job #6.

```

-----
ENTER ACTION CODE / PAGE NUMBER (P#)?8 <cr>   Print Job #6 is about to
PRINT-JOB#:6 <cr>                               be edited.
*ENTER COMMAND:L3 <cr> <----->             The first 3 lines of print
                                           job #6 are listed.

PAGE 1    01/12/84
ABCE 1256 CD
12123
.A4 <cr> <----->                             The next 4 lines are
                                           output to the assigned
                                           device in masked form for
                                           alignment.  In this
                                           example,

                                           XXXX 9999 XX
                                           99999
                                           XXXX 9999 XX
                                           99999

                                           would be printed.
.T <cr> <----->                             Current line position is
                                           set to top of job.

TOP
.P <cr> <----->                             The entire print job is
                                           output to assigned
                                           device.
-----

```

Figure A: Sample Usage of Alignment, Print and Exit Commands

```

-----
:SP-STATUS <cr>

```

QUEUE NAME	FORM TYPE	DEVICE ASSIGNED	DEVICE TYPE	DEVICE STATUS	NUMBER ENTRIES	PAGE SKIP
INVOICES	INVOICE	1	LPTR	SUSPEND	4	2
CHECK	PAYROLL	1	TAPE	NOT READY	0	0
LABEL	MAIL	2	PORT	ASSIGNED	0	0
SAMPLE		3	LPTR	ASSIGNED	0	0
PAYABLES		0	LPTR	NO DEVICE	0	0
PART		2	LPTR	ACTIVE	1	0

```

-----

```

Figure B: SP-STATUS Screen Display

CHANGING PRIORITY OF PRINT JOBS

| The position of a print job within a queue may be changed by either the |
SP-JOBS Action Code 10 (Change Priority) or the SP-PRIORITY verb.

The format of the Change Priority Action Code is:

```
ENTER ACTION CODE/PAGE NUMBER (P#)?10 <cr>  
PRINT-JOB#:
```

where:

PRINT-JOB# is prompting for the number of the print job whose placement in the queue is to be changed.

The format of the SP-PRIORITY verb is:

```
SP-PRIORITY print-job#
```

Both of the above-mentioned methods alter the printing priority of jobs within a form queue. The job priority is changed by moving the specified print job to the top of the queue. The printing of this job commences immediately after the current job is printed.

Figures A and B illustrate the usage of the Change Priority Action Code and the SP-PRIORITY verb. Figure C displays the SP-JOBS screen which would follow the execution of the example in Figure A or B.

```

:SP-JOBS <cr>
                                PRINT JOBS

QUEUE          PRINT          DATE
NAME           JOB           ACCOUNT  LINE  CREATED  STATUS  OPT  SIZE  COPIES

INVOICES       1            ACT        9    12 AUG   FINISH   H   12    1
PART           2            ACT        2    12 AUG   PRINT    H   15    1
INVOICES       3            MANF       3    12 AUG   OPEN     H    7    1
CHECK          4            PERS       9    12 AUG   WAIT     HS   27    3
INVOICES       6            MANF       3    12 AUG   PRINT    H   15    1
INVOICES       7            ACT        2    12 AUG   WAIT     H    9    1
INVOICES       8            MANF      10    12 AUG   WAIT     H   27    1

1. MOVE FORM QUEUE      5. DELETE JOB          9. SUSPEND PRINTING
2. MOVE PRINT JOB       6. STOP PRINTING      10. CHANGE PRIORITY
3. CHANGE OPTIONS       7. RESUME PRINTING   11. SP-STATUS
4. CHANGE #OF COPIES   8. EDIT PRINT JOB    99. EXIT

ENTER ACTION CODE / PAGE NUMBER (P#)?10 <cr>
PRINT-JOB#:8 <cr>
                                The Change Priority Action
                                Code moves job #8 to the
                                top of the form queue and
                                its printing will begin
                                immediately after print
                                job #6 is printed.

```

Figure A: Sample Usage of Change Priority Action Code

```

:SP-PRIORITY 8 <cr>           This SP-PRIORITY verb performs the same
:SP-JOBS <cr>                function as Action Code 10.

```

Figure B: Sample Usage of SP-PRIORITY Verb

```

                                PRINT JOBS

QUEUE          PRINT          DATE
NAME           JOB           ACCOUNT  LINE  CREATED  STATUS  OPT  SIZE  COPIES

INVOICES       1            ACT        9    12 AUG   FINISH   H   12    1
PART           2            ACT        3    12 AUG   PRINT    H   15    1
INVOICES       3            MANF       3    12 AUG   OPEN     H    7    1
CHECK          4            PERS       9    12 AUG   WAIT     HS   27    1
INVOICES       6            MANF       3    12 AUG   FINISH   H   15    1
INVOICES       7            ACT        2    12 AUG   WAIT     H    9    1
INVOICES       8            MANF      10    12 AUG   PRINT    H   27    1

```

Figure C: SP-JOBS Screen Display

TCL CONTROL OF PORTS: SP-PORTON PROC AND PORTOUT VERB

The SP-PORTON PROC starts the process which designates a port as available for spooler output. The PORTOUT verb is similar in function, but is restricted to enabling spooler output at the port at which it is executed.

The general form of the SP-PORTON PROC is:

```
SP-PORTON line# {form-queue} {(options)}
```

If you enter SP-PORTON <cr>, the system responds with a prompt for the required parameters, which are:

line#	specifies the line to be used as a spooler port
form-queue	specifies the name of the queue the print job(s) are assigned to. If the form queue name is entered, you must also enter the line number.
(options)	L, output to printer attached to video display terminal (otherwise, output direct to terminal). X, terminal to return to TCL (or controlling PROC) when despooling complete (and assigned form queue empty).

The SP-PORTON PROC logs on a port as an available device for spooler output. A PORT device cannot output print jobs until the SP-PORTON PROC is executed. Once SP-PORTON is executed, the attached device (which can be a video display terminal, serial character printer or other device) can be controlled by spooler verbs.

The line# may be any port number (the maximum number of ports is determined by the current system configuration), or it may be an asterisk (*), which specifies that print jobs will be sent to the terminal which originated the SP-PORTON PROC. Port zero cannot be used as a spooler port.

If a port process is initiated on the line on which it is going to run, it is considered a "normal" process, and can be interrupted by the BREAK function key. If SP-PORTON is initiated from another line with or without the optional queue name, it is a TIPH phantom process. If a queue name is specified, it will be used as the TASK ID of the TIPH process. See the section titled TERMINAL INDEPENDENT PROCESS HANDLER (TIPH) for a detailed explanation of TIPH.

Examples of various conditions under which SP-PORTON may be used are given in the next topic.

Note: Do not log off an SP-PORTON line. Use the SP-PORTOFF verb to terminate an SP-PORTON assignment.

PORTOUT

The PORTOUT verb may be executed at a terminal to make that terminal available as a spooler device.

The format of this command is:

```
PORTOUT {(options)}
```

The valid options are:

- L direct output to the attached ('local') printer
- X return to TCL (or controlling PROC) when the spooler form queue assigned to the port is empty.

One or both options may be used.

In the absence of the 'L' option, spooler output goes directly to the terminal (i.e. on a video display terminal it is displayed on the screen, on a printing terminal it is printed).

If the 'X' option is omitted, the BREAK function (usually <ctrl>BREAK) followed by END must be used to return to TCL.

Before this verb is executed at a terminal, a form queue should be assigned to the PORT device corresponding to that terminal.

Notes:

1. Terminals normally require a Terminal Executive to be loaded before they can be assigned to the spooler.
2. PORTOUT executes a catalogued DATA/BASIC program, and may therefore be run from a PROC.

TCL CONTROL OF PORTS: SP-PORTON EXAMPLES

| There are four conditions under which the SP-PORTON PROC may be used, |
corresponding to the examples listed here (illustrated by Figure A).

Condition #1

Form queue PAYROLL is created and assigned to a PORT device type (no device number assigned) prior to executing SP-PORTON. Subsequent use of the SP-PORTON PROC with the queue name PAYROLL causes the print jobs to be output to the specified port. The SP-STATUS display reflects a new line number in the "DEVICE ASSIGNED" column.

Condition #2

Form queue PROD is created without a device or device number assigned prior to executing SP-PORTON. The use of SP-PORTON with the queue name PROD automatically assigns that queue to the specified port, and all print jobs assigned to form-queue PROD are sent to that port. The SP-STATUS display reflects a new line number in the "DEVICE ASSIGNED" column and a "PORT" designation in the "DEVICE TYPE" column.

Condition #3

Form queue INVENTORY does not exist. First, the SP-PORTON PROC is executed without a form queue name to activate port process 15, which waits until a form queue is created with the same device number. Next, the form queue INVENTORY is created assigned to PORT 15. The SP-STATUS display reflects the device designation only after the form queue is created.

Condition #4

Form queue RECEIPTS has been assigned to a port that has a line number assigned prior to executing SP-PORTON. The use of SP-PORTON first activates the port process, which will then find the matching device number assigned to the port. The associated print job is output to the device on the specified port. The SP-STATUS display reflects the device assignment.

Condition #1

```

:SP-CREATE PAYROLL PORT 0 <cr> <---- This SP-CREATE verb creates form
                                     queue PAYROLL which is assigned
                                     to a port but no device line
                                     number is designated.
:SP-ASSIGN =PAYROLL H <cr> <----- All future print jobs assigned to
                                     PAYROLL will be hold files.
:LIST ENG 'PROBS' (P) <cr> <----- The two ENGLISH reports become
:LIST ACT 'INV' (P) <cr>             hold files assigned to form
                                     queue PAYROLL.
:SP-PORTON 25 PAYROLL <cr> <----- The SP-PORTON PROC causes both
                                     hold files in PAYROLL to be output
                                     to port line #25.

```

TASK PAYROLL STARTED ON LINE 25

Condition #2

```

:SP-CREATE PROD 0 0 <cr> <----- The SP-CREATE verb produces form
                                     queue PROD without a device
                                     assignment.
:SP-PORTON 4 PROD <cr> <----- The SP-PORTON PROC assigns port
                                     #4 to PROD.

```

TASK PROD STARTED ON LINE 4

```

:SP-ASSIGN =PROD <cr> <----- All future print jobs will be
                                     assigned to PROD and spooled to
                                     port #4.

```

Condition #3

```

:SP-PORTON 15 <cr> <----- The SP-PORTON PROC is executed
                                     before the form queue is created.
TASK STARTED ON LINE 15
:SP-CREATE INVENTORY PORT 15 <cr> <- The SP-CREATE verb produces the
                                     form queue INVENTORY which is
                                     assigned to port line #15.

```

Condition #4

```

:SP-CREATE RECEIPTS PORT 30 <cr> <-- This SP-CREATE verb produces the
                                     RECEIPTS form queue which is
                                     assigned to port line #30.
:SP-PORTON * <cr> <----- The SP-PORTON PROC activates the
                                     port process on line #30. The
                                     asterisk is entered because
                                     the operator is currently using
                                     port #30.

```

Figure A: Sample Usage of SP-PORTON PROC

SPOOLER

TCL CONTROL OF PORTS: SP-PORTOFF VERB

| The SP-PORTOFF verb may log off an SP-PORTON line before or after output |
is finished.

The general form of the SP-PORTOFF verb is:

SP-PORTOFF form-queue-name --or-- line#

or

SP-PORTOFF <cr>
ENTER LINE# --OR-- FQ-NAME:

where:

form-queue-name	specifies the name of the form queue that was assigned to the port
line#	specifies the line used as the spooler port

An asterisk (*) cannot be specified as a "line number". All form queue names must begin with either an alphabetic or a special character (except *). If the first character is numeric, it is treated as a "line number". If SP-PORTOFF is used before output is finished, SP-PORTON may be used to restart the port process. Output will begin at the first line of the job.

Figure A illustrates sample usage of the SP-PORTOFF Verb.

:SP-PORTOFF 30 <cr> <-----	The SP-PORTOFF verb terminates the use of line 30 as a spooler port (see Condition #4).
:SP-PORTON RECEIPTS <cr> <-----	If the previous SP-PORTOFF verb terminated the port process before the job was completed, the SP-PORTON PROC will reactivate the port process. Output begins on the first line of the job.

Figure A: Sample Usage of SP-PORTOFF Verb

SPOOLER

DRIVING A PRINTER ON THE PAn

Parallel printers attached to the Peripheral Local Area Network (Plan) are driven as PORT devices. Any associated form-queues must be assigned and PORT processes activated as described below. Serial printers attached to the PAn are also driven in a similar manner, although they are connected via standard terminal nodes.

Where a printer on the PAn is to be used as the default system printer, the STANDARD form queue should be re-assigned via SP-STATUS action code 4 or as follows:

```
SP-DEVICE STANDARD PORT n
```

where 'n' is the process number corresponding to the line printer to be used.

Any additional line printers on the PAn should be assigned to appropriate form-queues using SP-DEVICE, SP-CREATE, or one of the equivalent SP-STATUS action codes.

To enable line printer activation the associated processes should be logged-on by executing commands of the form:

```
SP-PORTON n
```

for each process number 'n' associated with a printer node.

Assignments remain in force until the spooler process is re-initialised as part of the coldstart procedure or following execution of the :SP-NEWTAB command. The commands outlined above should then be re-executed. Alternatively, these commands may be incorporated into the QUEUE-SETUP PROC (in file SYSPROG-PL), since QUEUE-SETUP is called by SYSTEM-SETUP and therefore should be executed after any type of coldstart.

Figure A shows an example QUEUE-SETUP PROC, tailored in this case to create form-queues NLQ and DRAFT, assign STANDARD to port 12, and start the processes associated with each form-queue.

Note: each of the port processes shown in the example could correspond to a parallel or to a serial printer.

```
-----  
    QUEUE-SETUP  
001 PQN  
002 C  TAILOR THIS PROC TO CREATE THE SPOOLER QUEUES  
003 C  REQUIRED FOR YOUR INSTALLATION  
004 C  EG.  
005 C  ... HSP-CREATE 11INCH LPTR 2  
006 C  ... P  
007 C  ... HSP-CREATE INVOICES PORT 3  
008 C  ... P  
009 C  ... Change next line if required  
010 HSP-SKIP STANDARD 1  
011 P  
012 HSP-CREATE NLQ PORT 2  
013 P  
014 HSP-CREATE DRAFT PORT 3  
015 P  
016 HSP-DEVICE STANDARD PORT 12  
017 P  
018 MV %2 "2","NLQ"  
019 [PROCLIB SP-PORTON]  
020 MV %2 "3","DRAFT"  
021 [PROCLIB SP-PORTON]  
022 MV %2 "12","STANDARD"  
023 [PROCLIB SP-PORTON]  
024 RTN  
-----
```

Figure A: QUEUE-SETUP PROC - Typical Example

SPOOLER

PRINTER CONTROL: P-ATT AND P-DET

| The P-ATT verb attaches your terminal line to the specified printer.
The P-DET verb detaches your terminal line from the specified printer.

The P-ATT verb has the following general form:

P-ATT {form-queue-name or device#}

where:

form-queue-name specifies the name of the queue assigned to the
desired printer.
device# specifies the printer number (which can be a number
from 1 through 4). If no parameters are given,
printer #1 is attached.

If you are already attached to the printer, only the TCL prompt is returned.
If the printer is attached to another line, the following message is
displayed on the screen:

LPTR ATTACHED TO LINE n.

where "n" represents the user's line number (or spooler process) the printer
is attached to. If the attachment is successful, the following message is
displayed:

LPTR ATTACHED.

The format of the P-DET verb is:

P-DET {form-queue-name or device#} {(U)}

The P-DET verb releases printer attachment, but only if your line is
currently attached to the printer. Otherwise, the following message is
displayed:

[655] NOT ATTACHED!

Users with SYS2 privileges may use the unconditional form (i.e., P-DET (U)).
This will detach the printer from any line except the spooler (which
detaches automatically when the queue is empty). An attempt to use this
form without SYS2 privileges, or an attempt to detach the printer from the
spooler, will lead to the following message:

[660] INSUFFICIENT PRIVILEGES OR DEVICE ATTACHED TO SPOOLER.

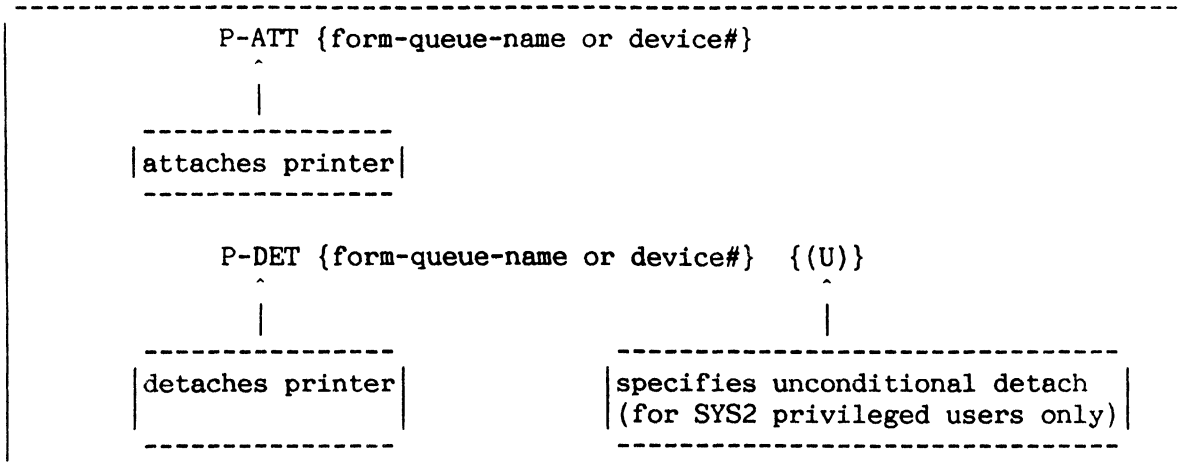


Figure A: General Form of P-ATT and P-DET Verbs

<u>EXAMPLES</u>	<u>EXPLANATION</u>
:P-ATT <cr> <----->	Attaches printer #1 (default printer).
:P-ATT INVOICES <cr> <----->	Attaches printer assigned to form queue INVOICES.
:P-DET <cr> <----->	Detaches printer assigned to user.
:P-DET PAYROLL <cr> <----->	Detaches printer assigned to form queue PAYROLL.
:P-DET 4 (U) <cr> <----->	Detaches printer #4, even if attached to another line (for SYS2 privileged users only).
:SP-ASSIGN <cr> <----->	All user spooler assignments are cleared and all devices attached for 'N' option output are cleared. Subsequent print jobs will be directed to the STANDARD form queue.

Figure B: Sample Usage of P-ATT and P-DET Verbs.

CREATION OF SPOOL JOBS FROM TAPE: T-PRINT PROC AND SP-TAPEOUT VERB

| The T-PRINT PROC and the SP-TAPEOUT verb enable data previously spooled |
tape to be sent to a printer or other assigned spooling device.

The T-PRINT PROC has the following general form:

T-PRINT {"string"}

This is a PROC which uses as input a reel of magnetic tape previously created by spooling to tape (tapes created by another procedure will not have the appropriate format). When T-PRINT is entered, the following steps are performed:

1. The tape unit(s) required are prompted for, and attached if available.
2. The magnetic tape is rewound to the load point.
3. The tape is searched for the "string" (if used).
4. Output of the report proceeds from the tape to the designated printer. Printing begins with the first line which contains the "string" (or the first line in the report if the string is omitted).
5. The tape is still attached to the line after T-PRINT is done.

The general form of the SP-TAPEOUT verb is:

SP-TAPEOUT {"string"}

This verb is used by the T-PRINT PROC. No device assignment is altered and no tape rewind is done. This verb can be used to build custom PROCs requiring tape jobs to be spooled. The next job on the magnetic tape is spooled according to the user's current SP-ASSIGN options.

The verb SP-TAPEOUT and PROC T-PRINT are summarized in Figure A. Examples are shown in Figure B.

```

-----
T-PRINT {"string"} <----- Tape unit is prompted for.  Tape and
                             printer are attached, tape rewound, and
                             the next tape job is spooled according
                             to current SP-ASSIGN options, starting
                             with first occurrence of optional
                             "string" (or first line if "string"
                             omitted).

SP-TAPEOUT {"string"} <----- Next tape job is spooled according to
                             current SP-ASSIGN options, starting
                             with first line containing optional
                             "string" (or first line if "string"
                             omitted).
-----

```

Figure A. Format of SP-TAPEOUT Verb and T-PRINT PROC

```

-----
EXAMPLE

:T-PRINT "XYZ123" <cr>

PRINTS ONE TAPE FILE ON THE PRINTER

Enter tape units desired, in sequence, separated by commas.
Maximum two half inch (1,2) or one quarter inch (3)?1<cr>

:

:SP-TAPEOUT <cr> <----- Next job on tape is spooled.
-----

```

Figure B. Sample Usage of SP-TAPEOUT Verb and T-PRINT PROC

SPOOLER

SPOOLER ACTIVATION: :RESTART-SPOOLER and :SP-NEWTAB Verbs

The spooler is automatically activated during job restore process. For further information regarding this process, refer to Section 10, SYSTEM MAINTENANCE PROCEDURES. The spooler may also be restarted with the :RESTART-SPOOLER verb; should :RESTART-SPOOLER fail, :SP-NEWTAB may be used to reinitialize the spooler.

The spooler may be activated under the following two conditions: 1) print jobs and queue data in use are retained by the spooler, and 2) print jobs and queue data are lost to the system.

The first condition occurs after a system restore with an "X" or "A" option, or after the :RESTART-SPOOLER verb is used. The spooler retains the addresses of its tables and all open print jobs are closed; therefore, no queue data or print jobs are lost.

The :RESTART-SPOOLER verb should only be used when neither the printer nor the tape unit is attached to a user's line and when no line is generating a spooled job. This verb may be used in an emergency situation such as a spooler abort. (The spooler has aborted if the message "SPOOLER ABORTED. RESTART BEING ATTEMPTED" appears on the SP-STATUS display).

The second condition occurs after a system restore with an 'X', 'A' or 'AF' restore and after the verbs :FILES, :ABS/FILES, and :SP-NEWTAB are used. Under such circumstances, all spooler tables and jobs are lost, including hold jobs, and the tables are reinitialized.

8. MAGNETIC TAPE UNITS
=====

MAGNETIC TAPE UNITS

| This section discusses the use of magnetic tape units, and the associated |
verbs. The system can accommodate up to three tape units.

The verbs discussed in this section are:

ASSIGN
CLEAR-ASSIGN
T-ATT
T-DET
T-FWD
T-BCK
T-REW
T-WEOF
T-UNLOAD
T-CHK
T-READ
T-DUMP
ST-DUMP
T-LOAD
MT-LOAD
MT-VERIFY
T-RDLBL
T-COPY
T-ERASE
T-RETENSION

This section also discusses tape unit sequencing for multiple tape units, tape labels, and multiple reel tape operations.

MAGNETIC TAPE UNITS

ASSIGN AND CLEAR-ASSIGN VERBS;TAPE UNIT SEQUENCING

The ASSIGN verb allows you to assign a unit list and a record size to a specified channel, and attaches the unit(s) assigned. The CLEAR-ASSIGN verb allows you to cancel channel assignments, and detaches the tape unit(s) previously assigned.

The general form of the ASSIGN verb is:

ASSIGN specification {;specification}...

where "specification" has the general form of:

{channel}=TAPE unit-list {SIZE=size}

"Channel", if present, is any alphabetic character from A to Z. If a channel is not specified, all existing assignments are destroyed, and the unit-list and record size specified are assigned to the default channel. If a channel is specified and the default channel exists, the default channel is destroyed. The ability to associate a channel name with a logical tape unit allows you to create applications and command streams which refer to channels by name.

The general form of "unit-list" is:

unit{,unit}...

where "unit" is 1, 2 or 3.

"Unit-list" must be present. A unit may not be assigned to two channels at the same time, and may not appear more than once in a list. Different types of unit may not be included in the same unit list. If a unit in the list has already been assigned to another channel, the message:

DEVICE TAPE n ASSIGNED TO THIS LINE, BUT FOR ANOTHER CHANNEL

is displayed, and the channel assignment is not changed.

"Size", if present, is a record size between 20 and 32,256. The default size is 8192.

The general form of the CLEAR-ASSIGN verb is:

CLEAR-ASSIGN {channel}

"Channel", if present, is any alphabetic character from A to Z. If a channel is not specified, all assignments are cancelled and any tape units attached to this line are detached. If a channel is specified, the assignment associated with the specified channel is cancelled and the attached tape units are detached.

CLEAR-ASSIGN operates only on channels associated with the user's own process.

MAGNETIC TAPE UNIT SEQUENCING

The system begins operations with the tape unit that is first in the unit-list of the channel currently assigned. It uses the units associated with the channel in the order they are specified in the unit-list. When the system reaches the end of the list, it returns to the beginning. When a tape is finished, the tape is rewound and the unit is put off-line. If the drive is ready when the system returns to use it again, the system assumes that a new tape has been mounted and continues with the tape operation. If the drive is not ready, the prompt:

```
MOUNT REEL # n ON DRIVE m
TYPE 'C' TO CONTINUE
```

is displayed.

When the system finishes a tape operation, it "remembers" the unit it finished on, and subsequent operations on the channel begin with this unit.

```
-----
:ASSIGN =TAPE 3
[1021] DEVICE(S) TAPE 3 NOW ASSIGNED,
      RECORD SIZE = 8192

:ASSIGN A=TAPE 1,2 SIZE=20
[1120] DEVICE(S) TAPE 1,2 NOW ASSIGNED TO CHANNEL 'A',
      RECORD SIZE = 20

:ASSIGN B=TAPE 3 SIZE=32256
[1120] DEVICE(S) TAPE 3 NOW ASSIGNED TO CHANNEL 'B',
      RECORD SIZE = 32256
-----
```

Figure A: Sample Usage of the ASSIGN Verb

```
-----
:CLEAR-ASSIGN A
-----
```

Cancels the assignment associated with channel A and detaches the associated tape units.

Figure B: Sample Usage of the CLEAR-ASSIGN Verb

MAGNETIC TAPE UNITS

T-ATT VERB

| The T-ATT verb attaches magnetic tape unit 1 to the terminal that issued |
the command and assigns a record size.

Note: The ASSIGN verb (described in the previous topic) attaches any tape unit (or set of units) and is preferred to the use of T-ATT.

The general form of the T-ATT verb is:

T-ATT {n}

where "n" is the record size

The system will automatically attach tape unit 1. To attach other tape units, use the ASSIGN verb. The T-ATT verb must be issued before the tape unit is used. If it is not, the message:

NO SUCH TAPE ASSIGNMENT EXISTS

is displayed. When the tape unit is attached, the message:

DEVICE(S) TAPE 1 NOW ASSIGNED,
RECORD SIZE = n

is displayed, where "n" is the record size that was attached. When the tape is attached, other users are not able to access it. If you attempt to attach the tape when it is already attached to a user, the message:

DEVICE TAPE n IS ASSIGNED TO LINE m

is displayed, where "m" is the line number the tape unit is attached to.

The magnetic tape unit is automatically detached when you log off the system.

Examples of the use of the T-ATT verb are provided in Figure A.

RECORD SIZE

The record size may be from 20 to 32,256 bytes. If a record is not specified with the T-ATT verb, the default size of 8192 is used.

Tape labels generated by the system contain the record size. If the tape is positioned at a label and the record size specified with the T-ATT verb is less than the record size contained in the label, the system automatically uses the size contained in the label. The change in record size is indicated by the message:

TAPE SIZE CHANGED TO n

where "n" is the size obtained from the tape label.

If the tape is not positioned at a label, the system uses the record size specified with the T-ATT verb. If the attached size is smaller than the actual size, the record is truncated at the size specified with the T-ATT verb.

```
-----  
:T-ATT <cr>  
[1021] DEVICE(S) TAPE 1 NOW ASSIGNED      No record size specified --  
RECORD SIZE = 8192                        default size used.  
  
:T-ATT 2048 <cr>  
[1021] DEVICE(S) TAPE 1 NOW ASSIGNED,     Tape was attached with a  
RECORD SIZE = 2048                        record size of 2048 bytes.  
-----
```

Figure A: Sample usage of the T-ATT Verb

MAGNETIC TAPE UNITS

CONTROL: T-DET, T-FWD, T-BCK, T-REW, T-WEOF, T-UNLOAD AND T-CHK VERBS

This topic describes the magnetic tape unit control verbs.

Note: Throughout the remainder of this section, where '{channel}' is shown in the general format, the channel identity must be included if one or more explicit channel assignments have been made. If no explicit channel assignments have been made, the channel identity may not be included.

The general form of the T-DET verb is:

T-DET {(U) n}

where: "(U)" allows you to detach the tape from any line except the spooler, if you have SYS2 privileges. This option should be used with extreme caution, since any process currently using the tape unit specified will abort.

"n" is the tape unit number specified with the "U" option

T-DET detaches tape unit 1. The unconditional form T-DET (U) requires that a unit be specified.

The general form of the T-FWD verb is:

T-FWD {channel} {n}

where: "channel" is any alphabetic character from A to Z

"n" is the number of records the tape will move forward

This verb moves the tape "n" records forward. The maximum value for "n" is 32,256. If "n" is not specified, the tape moves forward to the position immediately beyond the next end-of-file (EOF) mark. To perform multiple T-FWDs, use the SPACE PROC (see Section 9).

The general form of the T-BCK verb is:

T-BCK {channel} {n}

where "n" is the number of records the tape will move backward. This verb moves the tape back "n" records. If "n" is not specified, the tape moves to the position immediately before the previous EOF mark, or to load point. Before reading the next record, a T-FWD must be issued to position the tape after the EOF mark. This verb may not be used with cartridge tape drives.

The T-REW verb performs a "logical" rewind of the specified channel. Only the current drive is rewound. If there are multiple units in the channel's unit-list, the first tape unit in the list is set up as the "current" unit for the channel. Subsequent tape operations will begin with the "current" unit. The general format of this verb is:

T-REW {channel}

The T-WEOF verb writes an EOF mark on the tape; its general form is:

T-WEOF {channel}

The T-UNLOAD verb rewinds and puts off-line the tape unit belonging to the specified (or the default) channel which is currently in use. Its general form is:

T-UNLOAD {channel}

The T-CHK verb checks a tape file for parity errors. Error checking continues until an EOF mark is encountered. The general form of this verb is:

T-CHK {channel}

The T-CHK verb attempts to reattach the tape to the default size of 8190 bytes and all records are then read under this buffer size. This allows all records to be completely checked for parity regardless of the current T-ATT or ASSIGN size. At the end of the T-CHK, the tape buffer is re-attached at its original size. To perform multiple T-CHKs, use the CHECK PROC (see Section 9).

:T-DET <cr>	Detaches tape unit 1.
:T-FWD B 10 <cr>	Moves the tape assigned to channel B forward 10 records.
:T-BCK B 2 <cr>	Moves the tape assigned to channel B back 2 records.
:T-REW B <cr>	Rewinds "logical" tape unit assigned to channel B.
:T-WEOF B <cr>	Writes an EOF mark on the tape assigned to channel B.
:T-UNLOAD <cr>	Unloads the tape unit in use in the default channel.
:T-CHK B <cr>	Checks the tape assigned to channel B for parity errors.

Figure A: Sample Usage of the T-DET, T-FWD, and T-BCK Verbs

MAGNETIC TAPE UNITS

CARTRIDGE TAPE UNIT COMMANDS :T-ERASE AND T-RETENSION

Two additional verbs, T-ERASE and T-RETENSION, are provided for use with the quarter-inch cartridge tape drive (when present on the system). The T-BCK verb may not be used with this type of unit.

These verbs are used with the quarter-inch tape (if present) and may not be used with other tape drives. The T-BCK verb may not be used with this type of drive; all other tape commands may be used with either type.

The quarter inch tape drive is tape unit 3. Before executing T-ERASE or T-RETENSION, this tape unit should be attached via an ASSIGN statement. If attached as the default unit, these verbs are used without parameters. If attached and assigned to a named channel, this channel must be specified in the command.

T-ERASE

The general form of this command is:

T-ERASE {channel}

Execution of this command causes the entire tape to be erased and the tape re-positioned at BOT (beginning of tape). The command should be executed before writing new data to the tape.

T-RETENSION

The general form of this command is:

T-RETENSION {channel}

Execution of this command causes the tape to be retensioned by being wound forward and then rewound, the tape being left at BOT. This should be done before reading data from the tape.

```
-----  
:ASSIGN =TAPE 3<cr>  
[1021] DEVICE(S) TAPE 3 NOW ASSIGNED,  
      RECORD SIZE = 8192  
  
:T-ERASE<cr>  
  
:
```

Figure A: Sample Usage of the T-ERASE verb

```
-----  
:ASSIGN A=TAPE 3 SIZE=16384<cr>  
[1120] DEVICE(S) TAPE 3 NOW ASSIGNED TO CHANNEL 'A',  
      RECORD SIZE = 16384  
  
:T-RETENSION A<cr>  
  
:
```

Figure B: Sample Usage of the T-RETENSION verb

MAGNETIC TAPE I/O: T-READ VERB

The T-READ verb dumps the contents of a tape to the terminal or printer.

The general form of the T-READ verb is:

T-READ {channel} {(options)}

where:

"channel" is any alphabetic character from A to Z

"options" are:

X Dumps in hexadecimal instead of character format

P Dumps to the printer

N Inhibits automatic paging of output to the terminal

n{-m} Dumps records from "n" to "m" as specified, counting
 from the current position of the tape. If "m" is
 omitted, m=n is assumed. If the entire "n-m" option
 is omitted, all records up to the EOF will be dumped

This verb dumps the contents of a tape to the terminal, or optionally to the printer. The T-READ operation terminates when the specified number of records have been dumped, or when an EOF is detected.

Sample usage of the T-READ verb is illustrated in Figure A.

:T-READ (4-6) <cr>	Bypasses the first three records, dumps 4th, 5th and 6th on the terminal, and positions tape at beginning of 7th record.
:T-READ (P,4-6) <cr>	Same as above, but dumps to the printer.
:T-READ B (X) <cr>	Dumps the tape assigned to channel B in hex on the terminal until an EOF mark is detected.

Figure A: Sample Usage of the T-READ Verb

MAGNETIC TAPE I/O: T-DUMP AND ST-DUMP VERBS

The ENGLISH T-DUMP and ST-DUMP verbs cause selected file items to be dumped to the default magnetic tape unit.

T-DUMP is an ENGLISH verb which allows you to dump selected file items to the magnetic tape unit. The ENGLISH verb ST-DUMP works like T-DUMP, except a SORT is performed prior to the dump. T-DUMP and ST-DUMP work only with the default channel. The general form of an ENGLISH input sentence using the T-DUMP verb is:

```
T-DUMP {DICT} file-name {item-list} {selection-criteria}
{HEADER "identification"} {(options)}
```

where:

"file-name" is the name of the file to dump

"item-list" is a list of the item-ids to dump

"selection-criteria" are specified output conditions

"identification" is the tape label information

"options" may be:

I Lists the item-ids as they are dumped

P Prints the item-id list on the printer when used
 with the 'I' option

T Inhibits writing the tape label

The file-name may be preceded by the DICT modifier to dump dictionary data. File definition items (D/CODE = D) are not dumped. The "item-list" and "selection-criteria" (described in the ENGLISH Reference Manual) cause a selected subset of items in the specified file to be dumped to tape. As in other ENGLISH statements, each item-id must be enclosed in single quotes. If the "item-list" and "selection-criteria" are omitted, all items are dumped.

Note: The "item-list" may be supplied by a previous SELECT, SSELECT, GET-LIST, FORM-LIST or equivalent verb, in which case the selection-criteria are omitted.

The "HEADER" option allows you to generate tape label identification. The "identification" material must be enclosed in double quotes. Labels are always written at the beginning of each tape file, unless specifically suppressed. For further information see the Tape Labels topic in this section.

Figure A. illustrates the general form of T-DUMP and ST-DUMP ENGLISH sentences. Sample usage of the T-DUMP verb is illustrated in Figure B.

Files saved by a T-DUMP operation may be used with the ENGLISH 'TAPE' modifier. For more information, refer to the topic USING MODIFIERS AND OPTIONS in the ENGLISH Reference Manual.

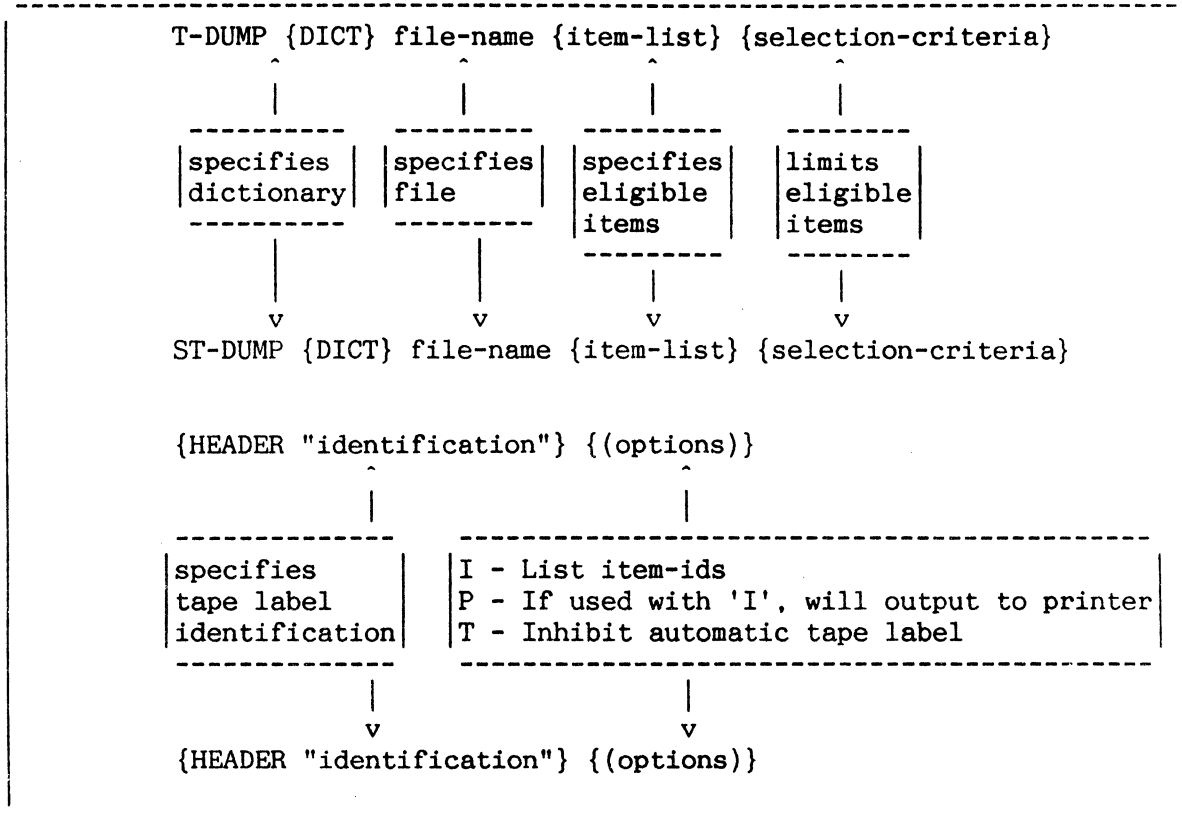


Figure A: General Form of T-DUMP and ST-DUMP ENGLISH Sentences

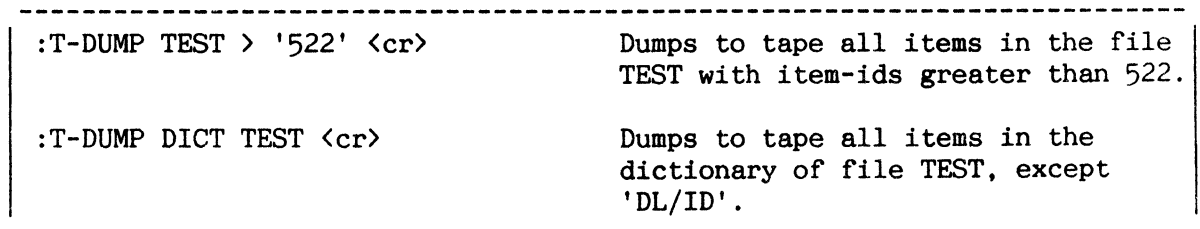


Figure B: Sample Usage of the T-DUMP Verb

MAGNETIC TAPE UNITS

MAGNETIC TAPE I/O: T-LOAD VERB

| The T-LOAD verb allows you to load files previously saved by a T-DUMP |
operation.

The T-LOAD verb works only with the default channel. The general form of an ENGLISH input sentence using the T-LOAD verb is:

```
T-LOAD {DICT} file-name {item-list} {selection-criteria} {(options)}
```

where:

"filename" is the name of the file to be loaded

"item-list" is a list of the item-ids to be loaded

"selection-criteria" are specified output conditions

"options" are:

- O Overwrites all items currently in the file that have
 item-ids corresponding to those on the tape
- P Lists item-ids on the printer as they are loaded
- S Suppresses the item-id listing

This verb writes to disc dictionaries or data files saved by a T-DUMP operation. Data from the tape are loaded to the file "file-name". The file-name may be preceded by the DICT modifier to load dictionary data. The "item-list" and "selection-criteria" (described in the ENGLISH Reference Manual) cause a selected subset of items in the specified file to be loaded.

Note: The "item-list" may be supplied by a previous SELECT, SSELECT, GET-LIST, FORM-LIST or equivalent verb, in which case the selection-criteria are omitted. Item-ids are listed at the terminal as they are loaded, unless the 'P' option is used to specify the printer. The tape is positioned at the EOF mark at the conclusion of the operation.

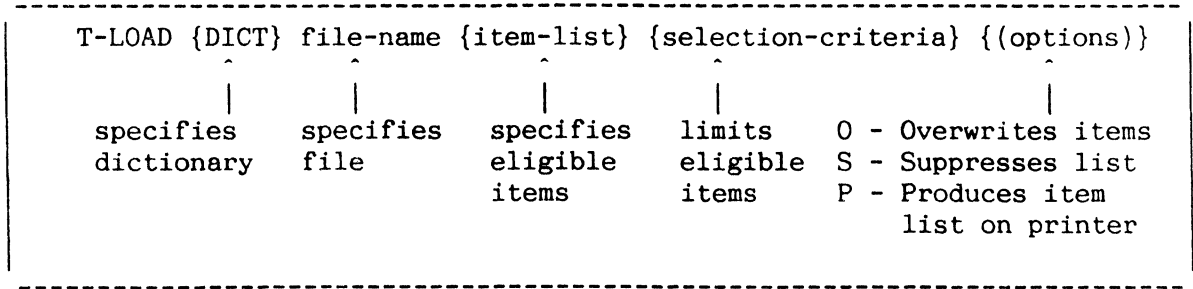


Figure A: General Form of T-LOAD ENGLISH Sentence

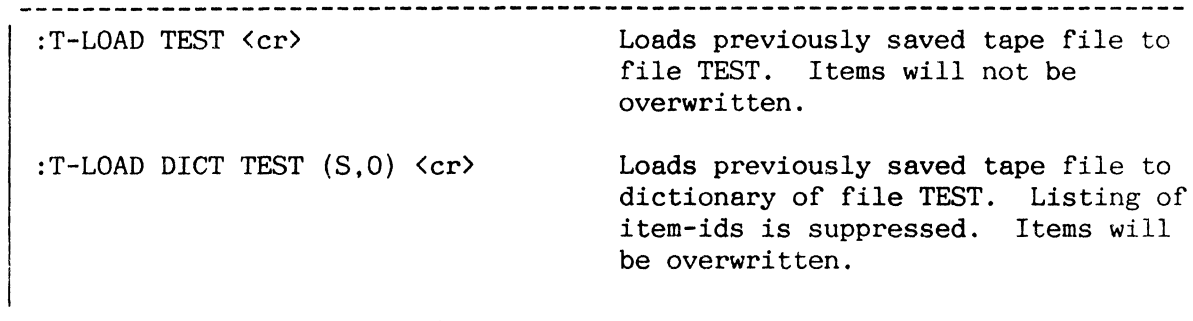


Figure B: Sample Usage of the T-LOAD Verb

MAGNETIC TAPE UNITS

MAGNETIC TAPE I/O: MT-LOAD AND MT-VERIFY VERBS

The MT-LOAD verb loads assembly code from magnetic tape. The MT-VERIFY verb verifies the assembly code. The T-DUMP verb must be used to write the assembly language routines ("modes") to tape prior to using MT-LOAD or MT-VERIFY.

The MT-LOAD and MT-VERIFY verbs work only with the default channel. The general form of the MT-LOAD verb is:

```
MT-LOAD {item-list}
```

where "item-list" specifies the item-ids to load.

This verb loads the item-ids specified in the "item-list" from tape to the ABS area. Items in the "item-list" are separated by blanks. The "item-list" may be supplied by a previous SELECT, SSELECT, GET-LIST, FORM-LIST or equivalent verb. If the "item-list" is omitted, all item-ids are loaded.

The general form of the MT-VERIFY verb is:

```
MT-VERIFY {item-list} {(options)}
```

where:

"item-list" specifies the items-ids to verify

"options" are:

- A Causes a columnar listing of all mismatched bytes. Each byte in the source file which mismatches is listed, followed by the value in the ABS frame. If this option is not specified, only the first mismatched byte in each frame is printed
- E Inhibits printing of verification message lines. When this option is specified, only error messages are printed
- P Sends output to the printer

This verb checks the modes stored on tape against the ABS area. As each mode is verified, or if a mismatch occurs, a message is displayed. The modes which are to be verified are specified in the optional "item-list".

The MT-LOAD and MT-VERIFY routines do not position the tape. This must be done by the user. The routines work with items placed on the tape by the ENGLISH verb T-DUMP. Modes are retrieved from the tape until either an EOF is detected, or the "item-list" is exhausted. If processing has stopped because the "item-list" is exhausted, the tape will probably not be positioned at end-of-file.

Sample usage of the MT-LOAD and MT-VERIFY verbs is illustrated in Figure A.

:MT-LOAD <cr>	Loads all modes from the tape into executable frames.
:MT-LOAD TCL-I <cr>	Loads only the TCL-I mode from tape.
:MT-LOAD TCL-I EDIT-I DB6 <cr>	Loads the modes TCL-I, EDIT-I and DB6 from tape.
:MT-VERIFY <cr>	Verifies that the modes stored in virtual memory are identical to the modes stored on the tape.
:MT-VERIFY (E) <cr>	Verifies that the modes stored in virtual memory are identical to the modes stored on the tape, and suppresses verification message lines.
:MT-VERIFY SORT <cr>	Verifies only the mode SORT.

Figure A: Sample Usage of MT-LOAD and MT-VERIFY Verbs

MAGNETIC TAPE UNITS

TAPE LABELS

This topic includes information on tape label size, placement and use.

Many system processors generate tape labels when writing to tape. These processors also check for a label when reading a tape. Processors that write and/or check for tape labels are DATA/BASIC, ENGLISH T-LOAD and T-DUMP, file-save, and file-restore.

A tape label is 50 characters long and is written at the beginning of the reel. When you generate a tape, you may specify identification information which will be included in the label. This information may be up to 16 characters long. If identification information is specified, the format of the label is:

(SM) L identification (VM) time date (AM) reel# (AM)
record size (AM) (SM) blanks (if necessary)

If the label is not 50 characters long, it will be padded at the end with blanks.

Label creation may be suppressed by using the 'T' option with any verb that causes tape output (T-DUMP, SAVE, RUN, etc.). This option is available to provide compatibility with very early REALITY systems and for data transfer to other types of system only.

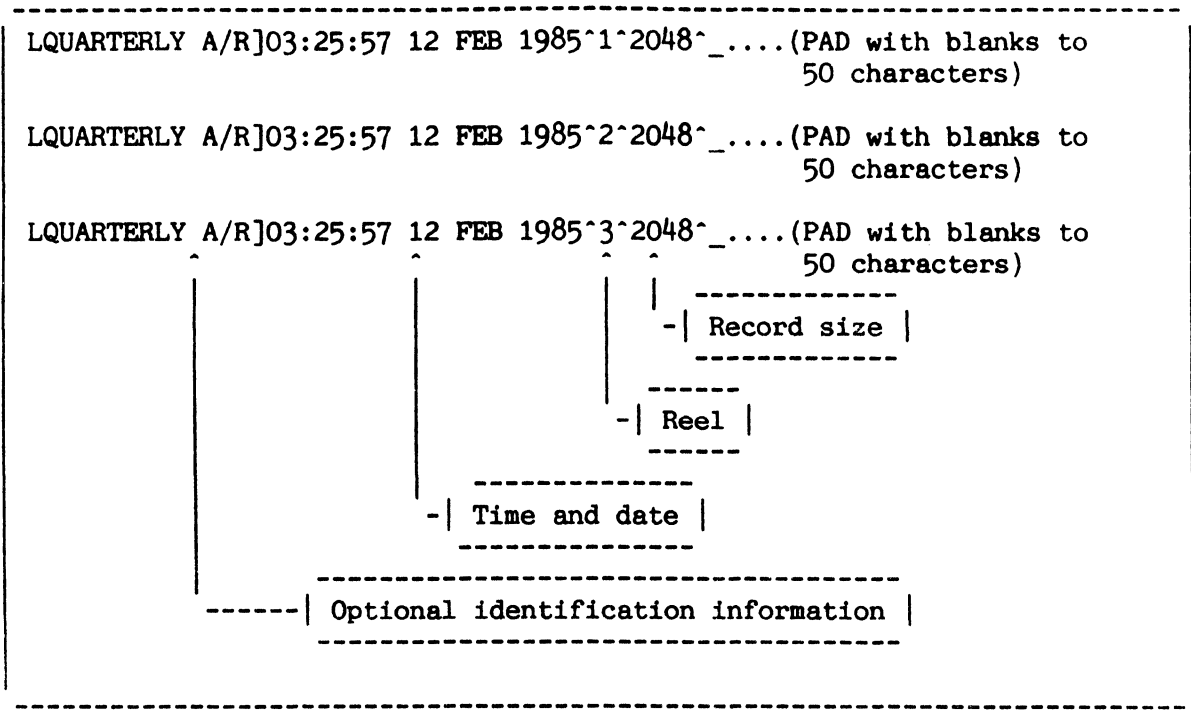


Figure A: Example of Labels for a Multireel Tape Operation

MAGNETIC TAPE UNITS

READING TAPE LABELS: T-RDLBL VERB

| Tape labels are generated by most tape writing operations and may be read |
with the T-RDLBL verb.

The general form of the T-RDLBL verb is:

T-RDLBL {channel} n

where:

"channel" is any alphabetic character from A to Z

"n" is the reel number

This verb reads, displays and stores the label from tape reel number "n" on the specified channel. T-RDLBL automatically changes the record size if it is different from the size previously attached.

T-RDLBL must be used to initialize the internal label storage area when you are working with file-save or account-save tapes, or when beginning tape operations on a reel other than Reel #1.

You only need to use T-RDLBL once when working on a tape set, as the label data is stored during a logon session. An example is a SEL-RESTORE which begins with Reel #2 of a three reel file-save. When Reel #2 reaches end-of-tape (EOT), the system must be able to prompt for reel three. It is able to do so only if, prior to the SEL-RESTORE, Reel #2 was mounted at load point, and the verb "T-RDLBL 2" was used to initialize the label buffer.

NOTE

T-RDLBL must be used when performing an ACCOUNT-RESTORE, SEL-RESTORE or any other operation involving a save tape, even if it is a single reel operation.

:T-RDLBL 1 <cr> LABEL = DOC.ACCT]08:01:04 11 APR 1985 TAPE SIZE CHANGED TO 08190	Tape size changed to size specified in tape label.
:T-RDLBL B 1	Reads, displays and stores the label from the first reel assigned to channel B.
:T-RDLBL 3	Reads, displays and stores the label from the third reel.

Figure A: Sample Usage of the T-RDLBL Verb

MAGNETIC TAPE UNITS

MULTIPLE REEL TAPE OPERATIONS

This section describes multiple reel tape operations.

specified in the unit list. If this drive is offline, the prompt:

```
MOUNT REEL # n ON DRIVE m
TYPE 'C' TO CONTINUE
```

is displayed. When the next reel has been mounted, enter the character "C" to continue the operation. The system will ignore all entries other than "C".

NOTE

The prompt cannot be suppressed, and input to this prompt cannot be "stacked" by a PROC.

Figure A. lists the messages that may appear while reading labeled tapes. These messages will appear if a problem is encountered when reading the first record on any "next reel" operation.

If you enter Yes (Y) to the 'BYPASS LABEL CHECK' question, the following prompt is displayed:

```
TYPE A (ACCEPT), S (SKIP), OR Q (QUIT)
```

Enter 'A' to accept the "label" as a data record and resume read operations. Enter 'S' to skip the "label" and resume read operations with the next record. Enter 'Q' to exit to TCL.

REEL OUT OF SEQUENCE!

PREVIOUS LABEL: |- -50 bytes- -|
RECORD FOUND: |- -50 bytes- -|
BYPASS LABEL CHECK? (Y/N)
TYPE Y (YES), N (NO), Q (QUIT)

This message is displayed when the reel number on the labeled tape does not match the requested reel number, or if the first tape mounted is not reel number one.

WRONG LABEL!

PREVIOUS LABEL: |- -50 bytes- -|
RECORD FOUND: |- -50 bytes- -|
BYPASS LABEL CHECK? (Y/N)
TYPE Y (YES), N (NO), Q (QUIT)

This message is displayed if the tape label on a subsequent reel does not match the tape label on the first reel.

UNLABELED TAPE!

PREVIOUS LABEL: |- -50 bytes- -|
RECORD FOUND: |- -50 bytes- -|
BYPASS LABEL CHECK? (Y/N)
TYPE Y (YES), N (NO), Q (QUIT)

This message is displayed if an unlabelled tape is mounted when the first reel was labelled.

Figure A: Multiple Reel Tape Operation Messages

MAGNETIC TAPE UNITS

T-COPY VERB

| The T-COPY verb allows you to perform a tape-to-tape copy, provided you
have more than one tape unit.

The general form of the T-COPY verb is:

T-COPY channelx channely {n} {(F)}

where:

"channelx" is the source channel name

"channely" is the destination channel name

"n" is the number of files to copy

"(F)" causes consecutive save tape reels to be labeled

This verb allows you to perform a tape-to-tape copy. "Channelx" and "channely" must be specified, and the channels must be assigned before using the T-COPY verb. If "n" is not specified, the system will copy all files up to two consecutive EOFs. If the tape does not contain two consecutive EOFs, an unrecoverable parity error may appear in the first record following the actual data. You may terminate the copy at this point.

The tape copy is a logical copy, as there is no physical reel to reel correspondence between the original tape and the copy. The number of reels in the copy does not necessarily match the number of reels in the original.

If the original tape is labeled, the copy will also be labeled. Note that the time and date in the label of the copy will be the time and date the copy was made, not the time and date the original was made. If the original tape is not labeled, the copy will not be labeled.

:T-COPY A B	Copies all files from Channel A to Channel B.
:T-COPY A B 3	Copies 3 files from Channel A to Channel B.
:T-COPY A B (F)	Copies a FILESAVE or SYSGEN tape from Channel A to Channel B.

Figure A: Sample Usage of the T-COPY Verb

9. SPECIAL SYSTEM UTILITIES

=====

OVERVIEW

This section comprises miscellaneous commands, principally for setting-up and reporting system parameters, but also including general utilities for file management and other purposes.

The commands described include the following:

ADDD etc.	WHERE	LISTACC
PCM	SLEEP	LISTCONN
PORT-SETUP	LIMITS	LISTDICTS
LIST-PORTS	USER	LISTFILES
TYPEAHEAD-ON	DISCIO	LISTPF
TYPEAHEAD-OFF	TIMESLICE	LISTPROCS
TERM	SET-PRIV	LISTU
SET-TERM	POVF	LISTVERBS
TERMINAL	CHECK-SUM	LP80,LP106,LP136
BREAK-KEY-OFF	FIX-FILE-ERRORS	SPACE
BKOFF	SEARCH	COMPARE
BREAK-KEY-ON	ESEARCH	CHECK
BKON	COMP	ECOPY
BLOCK-PRINT	DELETE	EEDIT
BLOCK-TERM	SET-FILE	EBASIC
PRINT-ERR	PRINTRONIX	
I-DUMP	CT	
S-DUMP	CP	
DUMP		
ITEM		
GROUP		
ISTAT		
HASH-TEST		
MSG		
CLEAR-MSG		
WHO		
WHAT		
TIME		
SET-TIME		
SET-DATE		
DATE-FORMAT		

SPECIAL SYSTEM UTILITIES

ARITHMETIC VERBS

This initial topic describes the arithmetic verbs.

Figure A presents general forms of the arithmetic verbs. To terminate the verb format, press the RETURN key; the system will then return the arithmetic result.

Figure B shows sample usage of these verbs. Note that in the system output for the DIVD and DIVX verbs, a space separates the quotient from the remainder.

Also note that negative decimal numbers are specified by appending a minus sign immediately prior to the first digit of the number. Hexadecimal numbers are considered negative when in the range from X'FFFFFFFF' (-1) to X'800000000001' (-140,737,488,355,327).

```

-----
ADDD N m <----- Add decimal "n" to decimal "m"
ADDX n m <----- Add hex "n" to hex "m"
SUBD n m <----- Subtract decimal "m" from decimal "n"
SUBX n m <----- Subtract hex "m" from hex "n"
MULD n m <----- Multiply decimal "n" by decimal "m"
MULX n m <----- Multiply hex "n" by hex "m"
DIVD n m <----- Divide decimal "n" by decimal "m"
DIVX n m <----- Divide hex "n" by hex "m"
DTX n <----- Convert decimal "n" to hex
XTD n <----- Convert hex "n" to decimal
-----

```

Figure A: General Forms of Arithmetic Verbs

```

-----
:ADDD 1 1 <cr>
      2

:ADDX FFFFFFFF 8000000005 <cr>
      80000000004

:DIVD 78645 987 <cr>
      79 672
      ^ ^
      | |
      | |----- Remainder
      | |----- Quotient

:DIVD 1000000 25 <cr>
      40000 0

:DTX -2 <cr>
      FFFFFFFFEE

:ADDD -2 5000 <cr>
      4998

:MULX FFF EEF <cr>
      EEE111

:XTD ABC <cr>
      2748
-----

```

Figure B: Sample Usage of Arithmetic Verbs

SPECIAL SYSTEM UTILITIES

PORT CONFIGURATION MAINTENANCE (PCM)

This catalogued DATA/BASIC program is a utility for definition of line, terminal and user characteristics for the ports of the system. This utility uses the system-level file DEVICES.

This utility replaces commands TERM, SET-TERM and TERMINAL, which are retained for compatibility with earlier systems (and are described later in this section). In addition to the characteristics set up using these commands, PCM enables additional parameters to be set up as described in the remainder of this topic.

The facilities of PCM are restricted according to the privilege level of an account from which it is executed, as explained in the detailed description.

The dictionary of the system-level file DEVICES contains items to provide default characteristics, and characteristics for standard models. The data section of the file contains an item with format PORT*n (where n is a three-digit decimal port number) for each port whose characteristics have been defined. Initially, characteristics for ports 0 to 7 inclusive are set up to the default characteristics contained in dictionary items STD-LINE, STD-TERM and STD-USER. Items in the data section of DEVICES are added and modified via the PCM utility.

The command PORT-SETUP is used to implement the characteristics defined by the DEVICES file after a system restore. The command LIST-PORTS is used to list the ports defined in DEVICES, together with selected characteristics. These utilities are described after the detailed description of PCM which follows.

PCM COMMAND

The general format of this command is:

PCM

On entering this command, if you are logged on to an account with SYS2 privileges, a screen is displayed (as shown in Figure A) which prompts for selection of one of the following options:

- R selects re-definition of line, terminal or user characteristics for a port.
- D selects initial definition of line, terminal and user characteristics for a port.
- L selects listing of the standard definitions available for line, terminal or user definitions.

SPECIAL SYSTEM UTILITIES

The details of each selection are given in the following pages. Entering just <cr> to this prompt causes the PCM utility to be exited.

Note: Users with system privilege level 2 have full access to the facilities of PCM. Users with lower privilege levels have restricted access to these facilities, (and the prompts given are changed accordingly), as follows:

SYS1 - Ability to change and list terminal and user characteristics for current port only. Options to set up initial definitions, to change line characteristics, and to select port are not available and not prompted for.

SYS0 - Ability to change and list user characteristics for current port only. Other options not available and not prompted for.

```
-----  
:PCM<cr>  
  
PORT CONFIGURATION MAINTENANCE  
  
      OPTIONS ARE :-  
  
      R - REDEFINE LINE/TERMINAL/USER CHARACTERISTIC(S)  
      D - DEFINE LINE, TERMINAL & USER CHARACTERISTICS  
      L - LIST LINE/TERMINAL/USER DEFINITION(S)  
  
  
      ENTER OPTION :_
```

Figure A : PCM Command - Initial Menu (Example)

SPECIAL SYSTEM UTILITIES

PCM : RE-DEFINITION

| Selection of the 'R' option from the initial PCM menu allows
| re-definition of selected characteristics for a port which has already
been defined.

On selection of the 'R' option from the initial PCM menu, a further prompt for the port number is displayed (as shown in Figure A). A port number should be entered or just <cr> to indicate the current port.

If the port specified is already defined in the DEVICES file, the REDEFINE menu screen is then displayed as shown in Figure A.

This screen includes a prompt for the option required: re-definition of line, terminal or user characteristics for the specified port. Entering L, T or U causes display of the corresponding screen, as discussed in the following topics. Alternatively, entering just <cr> to this prompt redisplay the initial PCM menu.

If the port specified is not yet defined in the DEVICES file, the message:

PORT NOT YET DEFINED

is displayed; enter a <cr> to re-display the initial PCM menu.

Note: characteristics re-defined via one of the 're-define' screens described on the following pages are put into effect immediately after being filed.

```
-----  
ENTER OPTION   :R <cr>           (Prompts shown on initial  
PORT NUMBER    :<cr>             PCM menu screen)  
  
PORT CONFIGURATION MAINTENANCE - REDEFINE                                PORT 002  
  
      OPTIONS ARE :-  
  
      L - REDEFINE LINE CHARACTERISTIC(S)  
      T - REDEFINE TERMINAL CHARACTERISTIC(S)  
      U - REDEFINE USER CHARACTERISTIC(S)  
  
      ENTER OPTION   :_
```

Figure A : PCM - 'Redefine' Menu (Example)

SPECIAL SYSTEM UTILITIES

PCM : RE-DEFINE LINE CHARACTERISTICS

| Selection of the 'L' option from the re-definition menu displays the line |
| characteristics for the selected port, and allows any or all of these |
characteristics to be changed.

On selection of the 'L' option from the re-definition menu, the current line characteristics for the selected port are displayed as shown in Figure A (the example screen illustrates the default values).

If the characteristics shown are satisfactory, enter EX<cr> to re-display the re-definition menu.

Otherwise, enter a line number to change the characteristic shown on the corresponding line. The cursor is then moved to that line : enter a new value as required, followed by <cr>, or just <cr> to leave the characteristic as it is. Incorrect entries display the message 'INVALID RESPONSE': enter <cr> to return to the relevant line. On selection of a line, the valid responses are displayed on the lower part of the screen.

The cursor then returns to the previous prompt. Enter another line number to change another characteristic; enter FI to file the new characteristic(s), then return to the re-definition menu; enter EX to re-display the re-definition menu without filing any changes which have been made.

The meaning of the line characteristics and their possible values are explained in the following topic.

```
-----  
ENTER OPTION : L<cr> (Prompt on REDEFINE  
menu screen)  
  
PORT CONFIGURATION MAINTENANCE - REDEFINE LINE          PORT 002  
  
1. LOCATION OF DTE :UNKNOWN  
2. PCI SETTING      :78  
3. CTS FLOW CONTROL :OFF  
4. XON/XOFF         :ON  
5. DTR & RTS RAISED :NO  
6. AUTO DISCONNECT :NO  
7. MONITOR DSR/DCD :DSR  
8. BAUD RATE        :9600  
9. HUSHED LOGON     :NO  
10. TIMED INPUT      :NO  
11. TIMEOUT LIMIT    :0  
12. ON TIMEOUT       :CONTINUE  
  
IS THIS CORRECT (FI/EX/LINE NUMBER)?_
```

Figure A : PCM - 'Redefine Line' Screen (Example)

SPECIAL SYSTEM UTILITIES

PCM : LINE CHARACTERISTICS

The line characteristics, accessible only to SYS2-privileged users, are parameters related to the interface between a terminal and its host.

LOCATION OF DTE A user-defined description, in up to 25 characters, of the location of the corresponding terminal.

PCI SETTING The PCI (Programmable Communication Interface) setting for the current terminal (entered as a decimal number in the range 1 to 255). The bits in the PCI setting represented as a binary number are significant as follows:

<u>Bits (MSB=bit 0)</u>	<u>Meaning</u>
0	0 = 1 stop bit 1 = 2 stop bits
1	Not used
2	0 = odd parity 1 = even parity
3	0 = disable parity 1 = enable parity
4-5	00 = 5 bit character length 01 = 6 bit character length 10 = 7 bit character length 11 = 8 bit character length
6-7	Not used

CTS FLOW CONTROL Whether or not the CLEAR TO SEND flow control protocol should be in effect for this port (possible values ON or OFF). Note: this should be set according to the hardware configuration currently in use.

XON/XOFF Whether or not XON/XOFF flow-control is in effect for the specified terminal (i.e. the terminal controller will recognise XON and XOFF characters generated by the terminal : the terminal should be configured correspondingly) (valid settings ON, to implement control, or OFF).

DTR & RTS RAISED Whether or not the port is connected to a dial-up line (possible values YES or NO). (Sets DTR and RTS modem lines). If NO, next two prompts ignored.

AUTO DISCONNECT Whether or not this port should be automatically logged off when the monitored line is disconnected, or in case of power failure, to prevent unauthorised logon (possible values YES or NO). If NO, next prompt ignored.

MONITOR DSR/DCD Specifies line to monitor (possible values DSR or DCD). Auto disconnect effected when DATA SET READY or DATA CARRIER DETECT (as specified) goes 'low'.

SPECIAL SYSTEM UTILITIES

BAUD RATE	The baud rate for the specified terminal (possible values 19200/9600/4800/2400/1200/600/300/110/75). Must be set to correspond to the current terminal configuration.
HUSHED LOGON	When in effect, suppresses transmission to terminal of all logon and logoff messages (including 'LOGON PLEASE:'). This is provided for devices which cannot handle these messages (possible values YES or NO).
TIMED INPUT	When in effect, the line is monitored while awaiting input. If the input is not completed before the timeout limit, the action specified for ON TIMEOUT is taken. (At TCL level, for instance, input is completed when a <cr> is entered or more than 140 characters are entered.) When not in effect, the line is not monitored awaiting input and the next two prompts are not put into effect (possible values YES or NO).
TIMEOUT LIMIT	A timeout limit in the range 0 to 60 minutes.
ON TIMEOUT	The action to be taken on timeout. Entry LOGOFF causes the line to be logged off on timeout; entry CONTINUE causes input to be terminated (with a null input) and the controlling program to continue with the next instruction. At TCL level, CONTINUE produces no visible effect on timeout (if just re-displays the colon prompt).

```
-----  
1. LOCATION OF DTE :UNKNOWN  
2. PCI SETTING :78  
3. CTS FLOW CONTROL :OFF  
4. XON/XOFF :ON  
5. DTR & RTS RAISED :NO  
6. AUTO DISCONNECT :NO  
7. MONITOR DSR/DCD :DSR  
8. BAUD RATE :9600  
9. HUSHED LOGON :NO  
10. TIMED INPUT :NO  
11. TIMEOUT LIMIT :0  
12. ON TIMEOUT :CONTINUE  
-----
```

Figure A: Line Characteristics (listed with their default values)

SPECIAL SYSTEM UTILITIES

PCM : RE-DEFINE TERMINAL CHARACTERISTICS

Selection of the 'T' option from the re-definition menu displays the terminal characteristics for the selected port, and allows any or all of these characteristics to be changed.

On selection of the 'T' option from the re-definition menu, the terminal characteristics for the selected port are displayed as shown in Figure A.

If the characteristics shown are satisfactory, enter EX<cr> to re-display the re-definition menu.

Otherwise, proceed as described for the 'Redefine Line' screen.

The meaning of the terminal characteristics and their possible value are as follows:

PAGE WIDTH	The maximum number of characters to be displayed on one line (valid range 10 to 140).
PAGE DEPTH	The number of lines to be used per complete page/screen (valid range 10 to 65).
LINE SKIP	The number of blank lines per page on the terminal (valid range 0 to 55).
LF DELAY	Number of delay (idle) characters following each carriage return or line feed (valid range 0 to 130).
FF DELAY	Number of delay characters following each top-of-form and form-feed character.

If set to zero, no form feed character is output; if set to one, a form feed character is output to a printer but not to a terminal. If set to two or greater, a form feed character is sent to the terminal before each page (valid range 0 to 10).

BACKSPACE	The character to be used as backspace (destructive or non-destructive : see USER CHARACTERISTICS) (valid range 0 to 255, but normally set to 8).
-----------	--

TERMINAL TYPE	Set to: 3 printing terminal (cursor control sequences replaced by CR/LF). 4 other terminal not using XON/XOFF 68 other terminal using XON/XOFF (recommended default value).
---------------	--

The previous default value (0) may be used provided new 'negative parameter' cursor positioning functions in DATA/BASIC and PROC are not required.

SPECIAL SYSTEM UTILITIES

- UPPER CASE ONLY Whether terminal can accept only upper case characters (valid entries YES or NO).
- LINE 25 Whether line 25 is available for system messages (valid entries YES or NO). If set to 'NO', system messages are sent to the terminal preceded and followed by LF/CR characters.
- FIELD READ Specifies a terminal as configurable for field read: valid for models capable of loading the appropriate terminal executive only (valid entries YES or NO). (The 'Field' read facility, provided for use with terminals working in a network environment or remote from the host, allows input to be processed locally (at the terminal) to improve response time and system efficiency).

```
-----  
PORT CONFIGURATION MAINTENANCE - REDEFINE TERMINAL                                PORT 002  
  
1. PAGE WIDTH          :79  
2. PAGE DEPTH          :24  
3. LINE SKIP           :0  
4. LF DELAY            :0  
5. FF DELAY            :2  
6. BACKSPACE           :8  
7. TERMINAL TYPE       :68  
8. UPPER CASE ONLY     :NO  
9. LINE 25             :YES  
10. FIELD READ         :NO  
  
IS THIS CORRECT (FI/EX/LINE NUMBER)?_
```

Figure A : PCM - 'Redefine Terminal' Screen (Example)

SPECIAL SYSTEM UTILITIES

PCM : REDEFINE USER CHARACTERISTICS

| Selection of the 'U' option from the re-definition menu displays the user |
| characteristics for the selected port, and allows any or all of these |
characteristics to be changed.

On selection of the 'U' option from the re-definition menu, the user characteristics for the selected port are displayed as shown in Figure A.

If the characteristics shown are satisfactory, enter EX<cr> to re-display the re-definition menu.

Otherwise, proceed as described for the 'Redefine Line' screen.

The meaning of the user characteristics and their possible values are as follows:

TYPEAHEAD	Whether or not typeahead (entry of further commands while previous commands are still being processed) is to be allowed (valid entries YES or NO).
CONTROL X	Whether or not <c>X clears the whole typeahead buffer (CLEAR) or just the last command entered (NORMAL). (This is only significant when TYPEAHEAD is enabled).
LPTR PAGE WIDTH	The number of characters per line on the line printer to which spooled jobs from this port are directed (range 10 to 140).
LPTR PAGE DEPTH	The number of lines per page in the line printer to which spooled jobs from this port are directed (range 1 to 64).
LPTR LINE SKIP	Number of linefeeds to be appended to each form feed (range 0 to 63).
ERASE	Whether or not backspace is to be destructive or non-destructive (i.e. 'YES' setting effectively overwrites previous character, 'NO' just moves cursor back one position).
MESSAGES	Whether system messages to the specified terminal are to be accepted or suppressed (YES for acceptance, NO for suppression : see also TERMINAL CHARACTERISTICS description for line 25 allocation).

```
-----  
ENTER OPTION      :U<cr>  (Prompt shown on  
                           REDEFINE menu screen)  
  
PORT CONFIGURATION MAINTENANCE - REDEFINE USER                PORT 002  
  
1. TYPEAHEAD      :YES  
2. CONTROL X      :CLEAR  
3. LPTR PAGE WIDTH :132  
4. LPTR PAGE DEPTH :60  
5. LPTR LINE SKIP  :0  
6. ERASE           :YES  
7. MESSAGES        :YES  
  
IS THIS CORRECT (FI/EX/LINE NUMBER)?_
```

Figure A : PCM - 'Redefine User' Screen (Example)

SPECIAL SYSTEM UTILITIES

PCM : DEFINE LINE, TERMINAL AND USER CHARACTERISTICS

| Selection of the 'D' option from the initial PCM menu allows definition |
| of line, terminal and user characteristics for a port which does not have |
characteristics already defined.

On selection of the 'D' option from the initial PCM menu, a further prompt for the port number is displayed (as shown in Figure A). A port number should be entered or just <cr> to indicate the current port. If the port specified is already defined in the devices file, the message

PORT ALREADY DEFINED

is displayed; enter <cr> to re-display the initial PCM menu.

If the port specified is not already defined in the DEVICES file, the DEFINE LINE screen is then displayed as shown in Figure A.

The DEFINE LINE screen first prompts for an identifier. An identifier may be entered here (a list of identifiers may be obtained via the PCM 'list' option : see subsequent pages of PCM description).

If a valid identifier is entered, the parameters assigned to that identifier are displayed. Alternatively, if just <cr> is entered to the 'IDENTIFIER' prompt, each parameter is prompted for individually : if <cr> is entered to any of these prompts, the default value from identifier STD-LINE is used. Once all of the parameters are displayed, a prompt to file the line characteristics, exit, or change one of the parameters is displayed.

The parameters displayed are as described previously for REDEFINE LINE.

If EX<cr> is entered, the initial PCM menu is re-displayed.

If FI<cr> is entered, the line characteristics shown are saved and the DEFINE TERMINAL screen is displayed.

Entry of a line number allows that parameter to be changed as described previously for REDEFINE LINE.

The DEFINE TERMINAL screen prompts for an identifier similarly to the DEFINE LINE screen. Again, a valid identifier (such as STD-TERM) may be entered or parameters may be entered individually. On filing the parameters set up, the DEFINE USER screen is displayed; this functions similarly to the two previous screens (default identifier is STD-USER). Once user characteristics are filed, the complete set of characteristics now defined is filed in DEVICES and set up in the appropriate locations to immediately put into effect the new characteristics.

The description given previously for 'REDEFINE TERMINAL' and 'REDEFINE USER' describe the parameters which must be established via these screens. If any of the define screens is exited via EX <cr>, no characteristics are filed or set up : a complete set of characteristics must be defined in order to initially define characteristics for a port.

SPECIAL SYSTEM UTILITIES

PCM : LISTING STANDARD DEFINITIONS

| Selection of 'L' option from the initial PCM menu allows standard |
identifiers and their associated characteristics to be listed.

On selection of the 'L' option from the initial PCM menu, the LIST menu is displayed as shown in Figure A.

Selection of one of the options L, T or U causes the corresponding screen to be displayed including a prompt for an identifier.

Entering just <cr> to this prompt causes the standard identifiers for the corresponding set of characteristics - LINE, TERMINAL or USER characteristics according to the option selected - to be listed.

Alternatively, entering a valid identifier causes the characteristics defined for that identifier to be listed. Characteristics are listed as shown in the REDEFINE description, except that there is no option to change the standard characteristics.

In either case, subsequently entering <cr> re-displays the LIST menu.

Line characteristics identifiers include STD-LINE, together with other line characteristics identifiers which may vary and are therefore not described here.

Similarly, terminal and user characteristics identifiers include STD-TERM and STD-USER respectively, together with identifiers for local types of terminal and printer.

```

ENTER OPTION  :L<cr>  (Prompt shown on initial PCM
PORT CONFIGURATION MAINTENANCE - LIST
                                menu screen)

OPTIONS ARE :-

L - LIST LINE DEFINITION(S)

T - LIST TERMINAL DEFINITION(S)

U - LIST USER DEFINITION(S)

ENTER OPTION:

```

Figure A : PCM - 'LIST' Menu

```

ENTER OPTION  :L<cr>  (Prompt shown on LIST menu
PORT CONFIGURATION MAINTENANCE - LIST LINE DEFINITION
                                screen)

IDENTIFIER      :<cr>

STD-LINE        Standard Port characteristics
19200-LINE      19200 baud port

2 DEFINITION(S) LISTED_

```

Figure B : PCM - Sample Identifier Listing

SPECIAL SYSTEM UTILITIES

PCM : LIST-PORTS AND PORT-SETUP

Two additional utilities are associated with PCM. LIST-PORTS is a PROC which lists the ports which are defined in the DEVICES file. PORT-SETUP is a catalogued DATA/BASIC program which may be used to set up characteristics for ports not currently logged on when the characteristics have been defined or altered.

LIST-PORTS produces a report similar to the sample shown in Figure A.

PORT-SETUP prompts for whether ports not defined in DEVICES are to be set up with standard characteristics.

If the reply is Y (for 'yes'), PORT-SETUP sets up the complete set of characteristics for all ports configured on the system which are not currently logged-on. These characteristics are derived from the DEVICES file, if the corresponding item is set up, or else are set to the default characteristics associated with the standard identifiers STD-LINE, STD-TERM and STD-USER. New items are created in the DEVICES file corresponding to any ports set to the default characteristics.

If the reply is N (for 'no'), only ports defined in DEVICES are set up. This has the effect of setting up the line characteristics correctly in the 'nodes', if, for instance, the hardware configuration has changed or a system reset has been done.

In either case, the ports configured on the system are listed in turn with an appropriate message, as illustrated by the sample terminal listing shown in Figure B.

PORT-SETUP is executed from SYSTEM-SETUP, SYS-LOAD and SYS-UPDATE.

```

:LIST-PORTS <cr>
                                     10:19:30 16 APR 1985
PAGE      1

PORT. LOCATION.....DTR & RTS  AUTO....  BAUD.. PCI.....
                RAISED  DISCONNECT RATE  SETTING

000  MACK'S DESK                NO      NO      9600  78
001  PENNY'S DESK              NO      NO      9600  78
002  TIM'S DESK                 NO      NO      9600  78
003  FIONA'S DESK              NO      NO      9600  78
004  PETE'S DESK                NO      NO      9600  78
005  JON'S DESK                 NO      NO      9600  78
006  TED'S DESK                 NO      NO      9600  78
007  JON'S OFFICE              NO      NO      9600  78
033  JAMES'S ROOM              NO      NO      9600  78

9 ITEMS LISTED
:

```

Figure A : Sample LIST-PORTS Report

```

:PORT-SETUP <cr>
Do you wish undefined ports to be automatically defined (Y/N)? Y <cr>

PORT 000 ALREADY LOGGED ON
PORT 001 CHARACTERISTICS SET
PORT 002 ALREADY LOGGED ON
PORT 003 ALREADY LOGGED ON
PORT 004 ALREADY LOGGED ON
PORT 005 CHARACTERISTICS SET
PORT 006 CHARACTERISTICS SET
PORT 007 CHARACTERISTICS SET

:

```

Figure B : Sample PORT-SETUP Listing

SPECIAL SYSTEM UTILITIES

MODIFYING 'TYPEAHEAD' CHARACTERISTIC DURING A SESSION

| The verbs TYPEAHEAD-ON and TYPEAHEAD-OFF may be used to control the |
| typeahead facility during a session, and may be used in a logon PROC to |
control the use of this facility for a specific user.

The facility for entering further commands while a previous command is being executed, known as 'typeahead', permits up to 255 bytes of additional commands (including terminating characters) to be entered while an initial command is being processed.

This facility is normally turned on and off using the PCM utility. However, it may be more convenient to use the TYPEAHEAD-ON and TYPEAHEAD-OFF verbs, which turn the facility on and off (respectively) at the terminal from which they are executed. When the terminal is logged off, then back on again, the facility reverts to its PCM-defined setting, if it exists (otherwise it is not changed).

The general format of these verbs is:

TYPEAHEAD-ON

TYPEAHEAD-OFF

They may be executed, if required, from a logon or other PROC.

```
:TYPEAHEAD-ON<cr>
```

```
:TYPEAHEAD-OFF<cr>
```

```
:
```

Figure A : Use of TYPEAHEAD-ON and TYPEAHEAD-OFF Commands

SPECIAL SYSTEM UTILITIES

SETTING DEVICE CHARACTERISTICS: TERM VERB

Terminal and/or printer characteristics may be displayed or set for the the port you are logged on via the TERM verb. TERM alters these characteristics only for the duration of a session, and may therefore be suitable for inclusion in a logon PROC. When characteristics are to be changed permanently, PCM should be used.

TERM

This statement sets the terminal/printer characteristics for the port you are currently logged on. The general form of the TERM verb is:

```
TERM {a,b,c,d,e,f,g,h,i,j,k,l} {(m)}
```

where:

- a - is a terminal line length (i.e., number of characters per line). The 'a' parameter must be in the range $10 \leq a < 140$.
- b - is the number of print lines per page on the terminal.
- c - is the number of blank lines per page on the terminal (sum of 'b' and 'c' equals page length).
- d - is the number of delay or idle characters following each carriage return or line feed. This is used for terminals that require a pause after a carriage return or line feed. This parameter should be set between 20 and 127 for MATRIX terminals running at 1200 baud.
- e - is the number of delay characters following each top-of-form and form-feed control. If 'e' is two or greater, a form feed character is also output to the terminal before each page. If 'e' is zero, no form feed character is output to either the terminal or printer. If 'e' is one, a form-feed is output to the printer but not to the terminal.
- f - is the backspace character. This parameter should be eight for all terminals. This defines the character which, upon receipt by the system, is treated as a backspace. When this character is entered the system will delete the previous data character entered and echo the 'f' parameter followed by a space, followed by the 'f' parameter again. When this is the first character entered on a line, it is ignored.
- g - is the printer line length. This parameter must be in the range $10 \leq g < 140$.
- h - is the printer page length.
- i - is the terminal type. Set to 3 (printing terminal), 4 (other terminal not using XON/XOFF) or 68 (other terminal using XON/XOFF). Obsolescent value 0 compatible provided terminal-independent functions not in use.
- j - is the baud rate. The values supported are 19200, 9600, 7200, 4800, 3600, 2400, 2000, 1800, 1200, 600, 300, 150, 134, 110, 75 and 50.
- k - is a PCI setting which defines character length, parity and number of stop bits. This parameter may be a decimal number between 0

and 255.

If a terminal is to run at or below 110 baud, two stop bits should be specified. The format of this byte is defined in the PCM description under RE-DEFINE LINE CHARACTERISTICS.

- l - specifies the number of linefeeds to be appended to a formfeed (e.g., allows a WORDMATE header to be printed on a letter-quality printer).
- m - is a numeric option to specify the port number.

If a numeric option is entered ('m' parameter), the display/changes apply to the given port. However, the verb only allows you to change another port's baud rate and PCI setting. Attempted changes to any other TERM parameters on another port are rejected.

Individual parameters may be null (i.e. as specified by two adjacent commas in the TERM verb). If so, the previously defined parameter remains in force. If a parameter is specified that is out of range, an error message is output and that parameter retains its previously defined value. A TERM command without a parameter list displays the current characteristics.

If a coldstart, ABS load or 'AF' restore is performed, the system reverts back to the default terminal characteristics specified by the hardware.

```

-----
:TERM 79,24,1,5,5,8,132,60,4<cr> <----- Alter characteristics.

:TERM<cr> <----- Display characteristics.
          TERMINAL    PRINTER

PAGE WIDTH:      79          132
PAGE DEPTH:      24          60
LINE SKIP :      1          0 <----- Number of linefeeds appended
LF DELAY  :      5          to formfeed.
FF DELAY  :      5
BACKSPACE :      8
TERMINAL  :      4
BAUD RATE :    9600
PCI       :      78

:TERM .....85,56,,,2<cr> <----- Alter printer characteristics
                                     to 85, 56 and 2.
-----

```

Figure A: Sample Use of TERM Verb

SETTING DEVICE CHARACTERISTICS: SET-TERM VERB

Terminal and/or printer default characteristics may be set for all ports via the SET-TERM verb. This verb is retained for compatibility with earlier systems, but it is recommended that the PORT-SETUP verb is used to set characteristics for normal purposes. Ports for which characteristics are defined in the DEVICES file are not affected by SET-TERM settings.

The general format of the SET-TERM verb is:

SET-TERM {a,b,c,d,e,f,g,h,i}

where:

- a - is a terminal line length (i.e., number of characters per line). The 'a' parameter must be in the range $10 \leq a < 140$.
- b - is the number of print lines per page on the terminal.
- c - is the number of blank lines per page on the terminal (sum of 'b' and 'c' equals page length).
- d - is the number of delay or idle characters following each carriage return or line feed. This is used for terminals that require a pause after a carriage return or line feed. This parameter should be set between 20 and 127 for MATRIX terminals running at 1200 baud.
- e - is the number of delay characters following each top-of-form and form-feed control. If 'e' is two or greater, a form feed character is also output to the terminal before each page. If 'e' is zero, no form feed character is output to either the terminal or printer. If 'e' is one, a form-feed is output to the printer but not to the terminal.
- f - is the backspace character. This parameter should be eight for all terminals. This defines the character which, upon receipt by the system, is treated as a backspace. When this character is entered the system deletes the previous data character entered and echoes the 'f' parameter followed by a space, followed by the 'f' parameter again. When this is the first character entered on a line, it is ignored.
- g - is the printer line length. This parameter must be in the range $10 \leq g < 140$.
- h - is the printer page length.
- i - is the terminal type. Set to 3 (printing terminal), 4 (other terminal not using XON/XOFF) or 68 (other terminal using XON/XOFF). Obsolescent value 0 compatible provided terminal-independent functions not in use.

The SET-TERM verb sets the default printer and terminal characteristics except the baud rate and PCI setting for subsequent logons on all terminals. This verb is present only on the SYSPROG account.

If no parameter list is entered, the characteristics are displayed as for the TERM verb. The baud rate and the PCI setting for the port executing the verb are then printed.

Figure A gives examples of the use of the SET-TERM verb.

```

-----
:SET-TERM 79,24,1,2,2,21,80,60,4 <cr> <----- Alter characteristics.

:SET-TERM<cr> <----- Display characteristics.
                TERMINAL      PRINTER

PAGE WIDTH:      79          80          Shows the settings for
PAGE DEPTH:      24          60          standard terminal.
LINE SKIP :      1           0
LF DELAY  :      2
FF DELAY  :      2
BACKSPACE :      21
TERMINAL   :      4
BAUD RATE  :     9600
PCI        :      78

:SET-TERM ,,,,,,85,56 <cr> <----- Alter characteristics.

:SET-TERM<cr> <----- Display characteristics.
                TERMINAL      PRINTER

PAGE WIDTH:      79          85
PAGE DEPTH:      24          56
LINE SKIP :      1           0
LF DELAY  :      5
FF DELAY  :      5
BACKSPACE :      8
TERMINAL   :      4
BAUD RATE  :     9600
PCI        :      78
-----

```

Figure A: Sample Use of SET-TERM Verb

SPECIAL SYSTEM UTILITIES

SETTING DEVICE CHARACTERISTICS: TERMINAL PROC

The TERMINAL PROC allows you to build a parameter and options list for the TERM verb by printing a menu, soliciting values for each parameter, and editing the values entered. This PROC is retained for compatibility with earlier systems, but it is recommended that PCM is used to set characteristics for normal purposes.

To generate such a list, type in the following:

```
:TERMINAL
```

If a menu for another port is displayed, only changes made to the baud rate and PCI characteristics will be accepted. While entering data, you may enter "HELP" to get assistance or "EX" to exit.

The following is a list of messages that may be generated when errors are detected by the verb TERM, SET-TERM or the TERMINAL PROC.

- [1010] THE PARAMETERS BEING CHANGED ARE NOT COMPATIBLE WITH THE PORT OPTION.
- [1011] BAUD RATE AND PCI TRANSMISSION CHARACTERISTICS CANNOT BE MODIFIED WITH THE SET-TERM VERB. USE "TERM" INSTEAD.

```

:TERMINAL <cr>

WHICH PORT NUMBER (RETURN IF OWN PORT)? <cr>

      TERMINAL SETTING CONTROL MENU

          1.  TERMINAL PAGE WIDTH : 79 <cr>
          2.  TERMINAL PAGE DEPTH : 24 <cr>
          3.  LINE SKIP           : 1 <cr>
          4.  LINE FEED DELAY     : 5 <cr>
          5.  FORM FEED DELAY     : 5 <cr>
          6.  BACKSPACE CHARACTER : 8 <cr>
          7.  PRINTER PAGE WIDTH  : 85 <cr>
          8.  PRINTER PAGE DEPTH  : 56 <cr>
          9.  TERMINAL TYPE       : 4 <cr>
         10.  BAUD RATE           : <cr>
         11.  PCI SETTING         : <cr>
         12.  PRINTER LINE SKIP   : 2 <cr>

      IS THIS CORRECT (FI/LINE NUMBER/EX)? FI <cr>

:TERM <cr> <----- Display characteristics.
          TERMINAL    PRINTER
PAGE WIDTH:         79         85
PAGE DEPTH:         24         56
LINE SKIP :          1          2
LF DELAY  :          5
FF DELAY  :          5
BACKSPACE :          8
TERMINAL  :          4
BAUD RATE :
PCI       :

```

Figure A: Sample Use of TERMINAL PROC.

SPECIAL SYSTEM UTILITIES

DISABLING AND RE-ENABLING USE OF 'BREAK' FUNCTION

The BREAK-KEY-OFF (or BKOFF) verb disables the BREAK function on the current, or (optionally) specified, terminal. The BREAK-KEY-ON (or BKON) verb re-enables use of the BREAK function.

The BREAK-KEY-OFF verb (which has the synonym BKOFF) has the general format:

BREAK-KEY-OFF {(n)}

The effect of this command is to disable the BREAK Key (or its equivalent) on the terminal at which it is executed, or at terminal 'n'. Use of the 'n' option requires SYS2 privileges.

The BREAK-KEY-ON verb (which has the synonym BKON) has the general format:

BREAK-KEY-ON {(n)}

Use of this verb requires SYS2 privileges. This command re-enables the BREAK function on the current (or specified) terminal when previously disabled via the BREAK-KEY-OFF command, or via the BREAK KEY OFF statement in a DATA/BASIC program, or via a PROC user exit.

Figure A illustrates the use of these verbs.

```
:BREAK-KEY-OFF (5<cr>
```

```
:BKON (5<cr>
```

```
:
```

Figure A : Use of BREAK-KEY-OFF/BKOFF and BREAK-KEY-ON/BKON Commands.

BLOCKING PRINTING: BLOCK-PRINT AND BLOCK-TERM VERBS

| The BLOCK-PRINT or BLOCK-TERM verb prints characters in a 9-by-n |
block-form on the printer or your terminal, respectively.

The general form of the BLOCK-PRINT verb is:

BLOCK-PRINT character-string

This verb causes the specified character-string to be block-printed on the printer. Multiple character-strings separated by blanks are output on separate lines. Any character-string containing single quotes (') must be enclosed in double quotes ("), and vice versa. The surrounding quotes are not printed. A character-string not containing quotes as part of the string need not be surrounded by quotes, unless embedded blanks are to be preserved.

The general form of the BLOCK-TERM verb is:

BLOCK-TERM character-string

This verb is identical to the BLOCK-PRINT verb, except that the text string is output in block-print form to your terminal.

Character-strings must be such that the total number of characters does not exceed the current line length set by the most recent TERM verb.

If a BLOCK-PRINT or BLOCK-TERM verb is illegally formed, any of the error messages 520 through 525 may be displayed (refer to the list error messages in the Appendix).

BLOCK-PRINT and BLOCK-TERM verbs use the BLOCK-CONVERT file to create the characters. This existing file contains conversion specifications for all printable ASCII characters (no lower case alphas, however). With this file, characters will be printed in 9-by-12 to 9-by-20 blocks.

If you wish to change the way any character is printed, you must change the corresponding item in the BLOCK-CONVERT file. The item-id of the item is the character to be converted. Each item in the file must consist of exactly 10 attributes. The first must specify in decimal the number of horizontal bytes in the blocked character to be output (i.e., "n" of the 9-by-n block mentioned above). The second and subsequent attributes provide the conversion specification. These attributes contain one or more values; each value is separated from the preceding by a value mark (<c>],X'FD'). The first character of the first value in each attribute must be "C" or "B"; these signal that the output matrix line of the blocked character begins with a character or a blank, respectively. Immediately following must be the number of characters or blanks (in decimal). The presence of a value mark indicates a switch from character to blank status (or vice versa) and must be followed by the number of bytes to be output. For example, an "X" might be specified as shown in Figure B.

```

:BLOCK-TERM AB12 <cr>

      AAAA          BBBB BBBB          1111          2222222222
      AAAAAA        BBBB BBBB BBBB        111111        222222222222
      AAAAAAAA      BBBB  BBBB          1111111        2222  2222
      AAAA  AAAA    BBBB BBBB BBBB        1111111        2222
      AAAA  AAAA    BBBB BBBB BBBB          1111          2222222222
      AAAAAAAAAA    BBBB  BBBB          1111          2222222222
      AAAAAAAAAA    BBBB  BBBB          1111          2222
      AAAA  AAAA    BBBB BBBB BBBB        11111111111        22222222222
      AAAA  AAAA    BBBB BBBB BBBB        11111111111        22222222222

:BLOCK-PRINT AB12 <cr>

      Produces identical output as above, but on printer.

:BLOCK-PRINT "X'Y'Z" <cr>

      Produces block-print of "X'Y'Z".

```

Figure A: Sample Usage of BLOCK-PRINT and BLOCK-TERM Verbs

```

Item 'X' in file BLOCK-CONVERT

001 12 <----- Blocked character is 12 bytes wide.
002 C4]4]4 <----- Output 4 characters, 4 blanks, 4 characters
003 B1]4]2]4]1 <----- Output 1 blank, 4 characters, 2 blanks, 4
                        characters, 1 blank.
004 B2]8]2 <----- Outputs 2 blanks, 8 characters, 2 blanks.
005 B3]6]3 <----- Outputs 3 blanks, 6 characters, 3 blanks.
006 B4]4]4 <----- Outputs 4 blanks, 4 characters, 4 blanks.
007 B3]6]3 <----- Outputs 3 blanks, 6 characters, 3 blanks.
008 B2]8]2 <----- Outputs 2 blanks, 8 characters, 2 blanks.
009 B1]4]2]4]1 <----- Outputs 1 blank, 4 characters, 2 blanks, 4
                        characters, 1 blank.
010 C4]4]4 <----- Outputs 4 characters, 4 blanks, 4 characters.

```

Figure B: Sample Item in BLOCK-CONVERT File

DISPLAYING ERRMSG-FORMAT ITEMS: PRINT-ERR VERB

PRINT-ERR displays ERRMSG-format items as they would appear when displayed as error messages.

The general format of the TCL-II verb PRINT-ERR is:

PRINT-ERR filename item-list* {(options)}

where options may be:

- P direct output to printer (or other currently assigned spooler device).
- I causes prompts to be displayed for the parameters to be included in the error message.
- B prints each message on a new screen/page.

An asterisk in place of an item-list displays all items. Alternatively, the item-list may be supplied by an immediately-preceding SELECT, SSELECT, GET-LIST, FORM-LIST etc. statement; if omitted and not supplied previously, an item-list is prompted for.

Each parameter in an ERRMSG-format item is replaced in the output by a tilde (~), unless the 'I' option is specified. In this case, for each item in turn, prompts are displayed to allow entry of one or more parameters, until a <cr> is entered. The parameters input are substituted into the display in the order entered, but any parameters in excess of those entered are replaced by tildes. If both 'P' and 'I' options are specified, each item-id is displayed on the terminal before the relevant parameters are prompted for.

The figure opposite illustrates the use of the PRINT-ERR verb.

```
:PRINT-ERR ERRMSG 1 2 3 (I <cr>
Parameter:<cr>
[1] ILLEGAL USE OF DOUBLE-QUOTE IN ITEM-ID
Parameter:<cr>
[2] UNEVEN NUMBER OF SINGLE OR DOUBLE QUOTE-SIGNS ('").
Parameter:XXXX <cr>
Parameter:<cr>
'XXXX' IS NOT A VERB

:PRINT-ERR ERRMSG (IP <cr>)
ITEM-ID :1 2 3 <cr>
1
Parameter:<cr>
2
Parameter:<cr>
3
Parameter:XXXX <cr>

:
```

Figure A : Examples of Use of PRINT-ERR Verb

SPECIAL SYSTEM UTILITIES

DUMPING TO TERMINAL: I-DUMP AND S-DUMP VERBS

I-DUMP and S-DUMP are ENGLISH verbs that allow you to selectively dump dictionaries and data files to the terminal.

The I-DUMP verb (Figure A) has the following general form:

```
I-DUMP {DICT} file-name {item-list*} {selection-criteria} {(options)}
```

The "item-list" and "selection-criteria" (described in the ENGLISH Reference Manual) cause a selected subset of items in the specified file to be dumped to the terminal. If omitted, all items are dumped. The DICT modifier causes dictionary data to be dumped. File definition items (D/CODE=D) are not dumped. S-DUMP performs a sorted I-DUMP.

Valid options for these two verbs are:

N - Inhibit automatic paging of output to terminal

P - Direct output to printer

- * The item-list can be supplied by a previous SELECT, SSELECT, GET-LIST, FORM-LIST or equivalent command, in which case the selection-criteria are omitted.

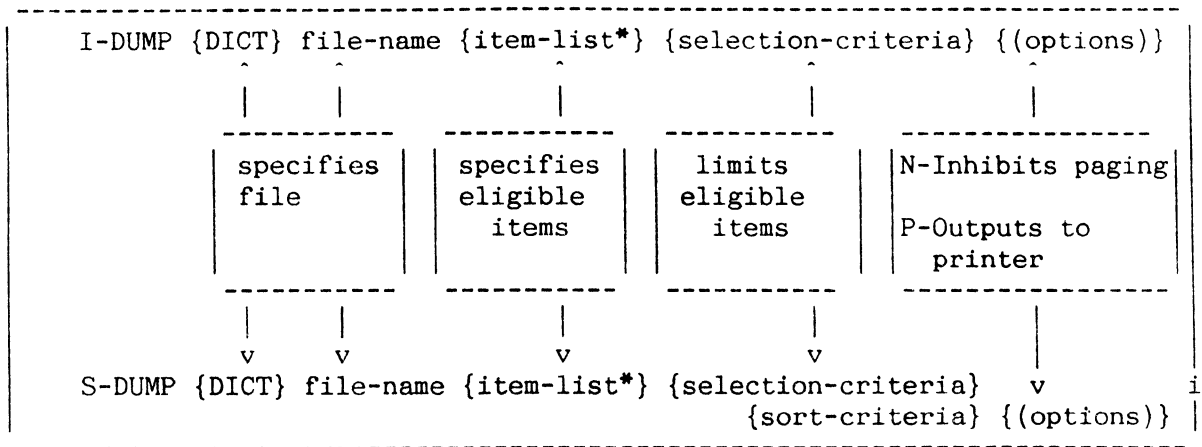


Figure A: General Form of I-DUMP and S-DUMP Verbs

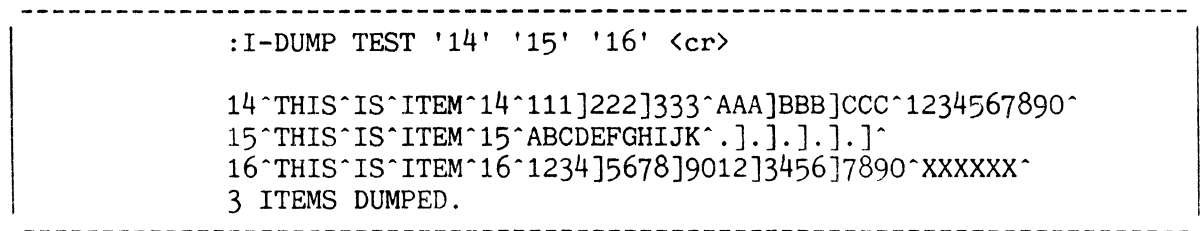


Figure B: Example of I-DUMP Verb

SPECIAL SYSTEM UTILITIES

DUMP VERB

The DUMP verb allows you to examine the contents of virtual memory frames in character or hexadecimal format. Use of this verb requires a system privilege level two.

The general form of the DUMP verb is:

```
DUMP fid1 {-fid2} {(options)}
```

where "fid1" is the beginning FID and "fid2" is the ending FID. If "fid1" is specified alone, only that frame is dumped; "fid1" and "fid2" may be specified as hexadecimal by preceding them with a period. If "fid2" is hexadecimal, it should not be preceded by the hyphen (-).

Valid options are listed below. Multiple options are separated by commas.

- G Dumps group data - all linked frames beginning with "fid1" are displayed; "fid2", if present, is ignored
- L Like 'G', but dumps link fields of frames only (no data dump)
- N Suppresses pagination of terminal display
- P Dumps to the printer instead of the terminal
- U When used with the 'G' or 'L' option, causes dump to follow backward links instead of default forward links; if 'U' is used without 'G' or 'L' option, the 'G' option is assumed.
- X Displays data in hexadecimal as well as character format

Link information is displayed in decimal and hexadecimal. A "+" next to a frame indicates a frame found by tracking forward links. A "-" indicates a link found by tracing backward links ('U' option). Figure B shows examples of the DUMP verb.

```
DUMP fid1{-fid2} {(options)}
```

- G Dumps group data
- L Dumps link fields only
- N Suppresses paging of terminal display
- P Dumps to printer
- U Traces backward links (with 'G' or 'L' options)
- X Dumps in hexadecimal format also

Figure A: General Form of DUMP Verb

```

:DUMP 5400 <cr>

          # of          # of
          Next          Previous
          Contiguous   Forward   Backward   Contiguous   Hex
          Frame        Link      Link       Frame        Equivalents
          ^            ^         ^           ^            ^
          |            |         |             |            |
FID:      5400: 0    10532    0    0 | (1518 : 0 2924 0 0) |

000 .....)S.....0015CHARGE-TO^P^6032^0017C-READ^P^2^94
050 ^^CUN^001AZERO-DISC-ERRS^P^20CF^001B*A11^A^11^.....
100 ^^L^10^1^001CCT^PQ^(DICT PROCLIB CT)^0018T-WEOF^PZ
150 ^6033^B023^002FBLOCK-CONVERT^Q^BLOCK-CONVERT^.....
200 ^L^10^1^002ALISTDICTS^PQ^(DICT PROCLIB LISTDICTS)^
250 0016SAVE-LIST^PL^2064^0013SP-STOP^P^10A6^001D*A3^A
300 ^03^.....^T10^L^10^1^001FJIM^Q^SYSTEM^JIM^.....^L^10
350 ^0014MT-VERIFY^PV^D1^0027PF^Q^SYSTEM^POINTER-FILE^
400 ^.....^R^10^0092BRANCH^ FRAME 400] FRM: 400 001 7FFO
450 0190  ORG 1 001 ^***09MAR77^.....^ BSL LA BX
500 001 19FD^ OR

:DUMP .2EA.2EC (L,U) <cr>      Dump frames X'2EA' through X'2EC'; dump
                                links only; trace backward links.

FID:   746 : 5 747           0 0   ( 2EA : 5 2EB   0 0 )
FID:   747 : 4 748           746 1   ( 2EB : 4 2EC   2EA 1 )
-FID:  746 : 5 747           0 0   ( 2EA : 5 2EB   0 0 )

FID:   748 : 3 749           747 2   ( 2EC : 3 2ED   2EB 2 )
-FID:  747 : 4 748           746 1   ( 2EB : 4 2EC   2EA 1 )
-FID:  746 : 5 747           0 0   ( 2EA : 5 2EB   0 0 )

:DUMP 5400 (X) <cr>           Dump frame 5400 in hex and character
                                format.

FID:   5400 : 0 10532    0 0   ( 1518 : 0 2924   0 0 )

000 00000000 29240000 00000000 30303135 000 :.....)S.....0015:
010 43484152 47452D54 4FFE50FE 36303332 015 :CHARGE-TO^P^6032:
.      .      .      .      .      .
.      .      .      .      .      .
1E0 2AFE2AFE 2AFE2AFE 2AFE2042 534C204C 480 :***** BSL L:
1F0 414258FC 30303120 31394644 FE204F52 496 :ABX 001 19FD^ OR:

```

Figure B: Examples of DUMP Verb

SPECIAL SYSTEM UTILITIES

FID INFORMATION: ITEM AND GROUP VERBS

ITEM and GROUP verbs provide information about the item and group structure of files. Output can be displayed at the terminal or optionally directed to the printer.

ITEM

The ITEM verb has the following general form:

```
ITEM file-name item-id {{options}}
```

This verb displays the base FID of the group into which the specified item-id hashes. If the item is not on file, the message "ITEM NOT FOUND" is displayed. In addition, every item-id in that group is listed in the order in which it occurs in the group along with a character count of the item (in hex). At the end of the list the following message is displayed:

```
n ITEMS m BYTES p/q FRAMES
```

where:

n is the number of items in the group
m is the total number of bytes used in the group
p is the number of full frames in the group
q is the number of bytes used in the last frame of the group

Valid options for this verb are:

N - Defeat automatic paging of terminal output
P - Direct output to printer
S - Suppress item list

GROUP

The GROUP verb has this general form:

```
GROUP file-name {{options}}
```

This verb is similar to the ITEM verb, but GROUP displays the base FID of every group in the specified file. In addition, every item-id in each group is listed along with a character count of the item (in hex). The end of list message for each group and valid options are the same as for the ITEM verb.

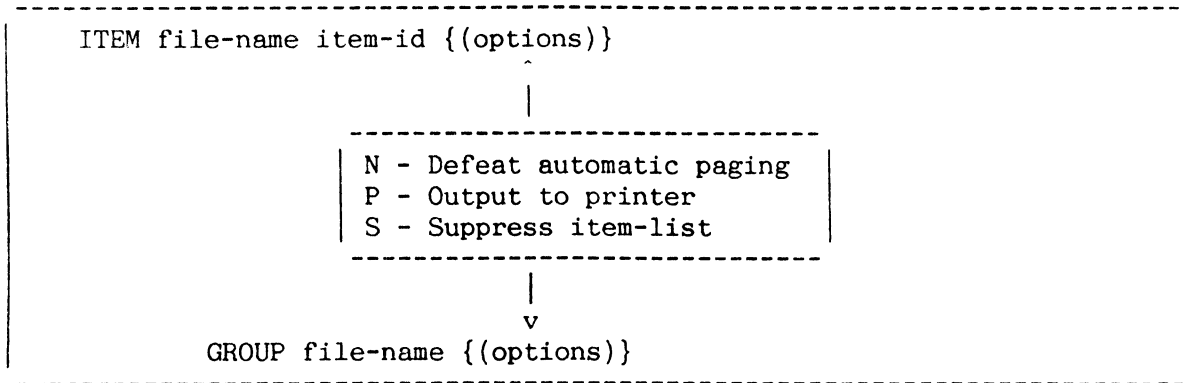


Figure A: General Form of ITEM and GROUP Verbs

```

:ITEM TEST 14 <cr>
1895 <----- Beginning FID of group in which
0007 10          item '14' resides. Order of items
0007 11          is '10', '11', and '14'; each is
0007 14          seven bytes.
3 ITEMS 21 BYTES 0/21 FRAMES

:ITEM TEST 14 (S) <cr>
1895
3 ITEMS 21 BYTES 0/21 FRAMES

:GROUP TEST <cr>
1895 <----- Beginning FID of first group.
0007 10
0007 11
0007 14
3 ITEMS 21 BYTES 0/21 FRAMES
1896 <----- Beginning FID of second group.
0009 12
000A 33
0008 28
0010 92
4 ITEMS 43 BYTES 0/43 FRAMES
1897 <----- Beginning FID of third group.
001F 88
1 ITEM 31 BYTES 0/31 FRAMES

```

Figure B: Sample Usage of ITEM and GROUP Verbs

SPECIAL SYSTEM UTILITIES

HASHING INFORMATION: ISTAT AND HASH-TEST VERBS

ISTAT and HASH-TEST are ENGLISH verbs that produce file hashing statistics and an optional histogram; ISTAT for specified file items and HASH-TEST on the basis of a user-specified test modulo.

ISTAT

An ENGLISH sentence using the ISTAT verb is illustrated in Figure A. The ISTAT verb provides file hashing statistics and, if desired, an optional histogram for selected items in the selected file, shown by examples in Figure B. For further information regarding item hashing, refer to the topic ITEM STORAGE AND THE HASHING ALGORITHM in Section 2.

HASH-TEST

HASH-TEST produces the same output as ISTAT, but uses a user-specified test modulo. The general form of this verb is shown in Figure A. Figure C illustrates a HASH-TEST dialog and printout.

Options for these two verbs are:

- H - Prints histogram
- N - Defeats automatic paging of output to terminal
- P - Directs output to printer

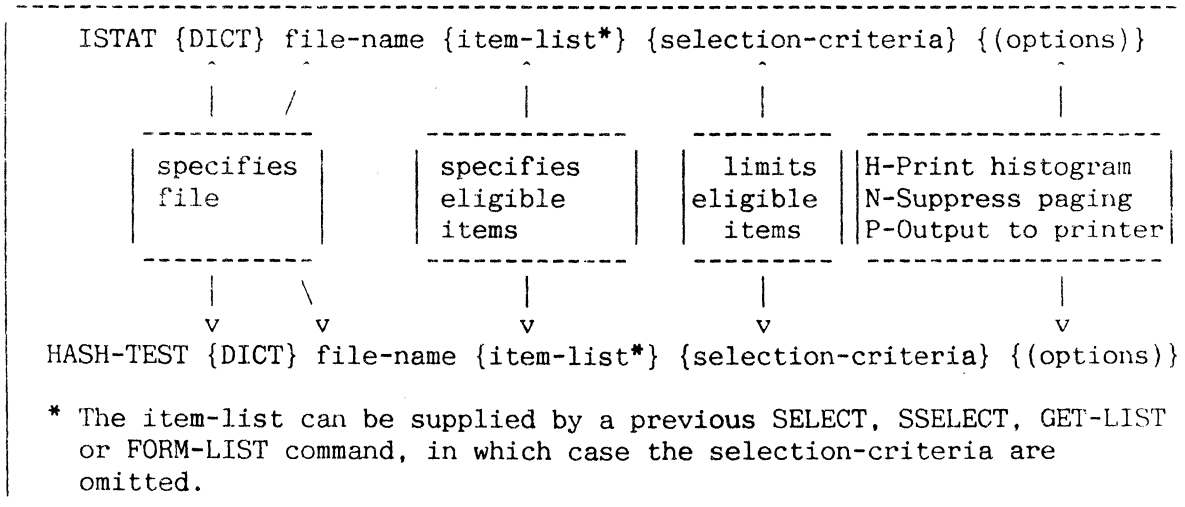


Figure A: General Form of ISTAT and HASH-TEST Verbs

SPECIAL SYSTEM UTILITIES

INFORMATION: MSG, CLEAR-MSG, WHO, AND WHAT VERBS

These verbs provide miscellaneous system information and capabilities.

MESSAGE OR MSG

All users may communicate with other users on the system. To transmit a message to another user, type the verb MESSAGE (or MSG) followed by the user's account name or '!' followed by a destination port#, then the text of the message. The general form is:

```
MESSAGE account-name text-string
```

or

```
MESSAGE !n text-string
```

where: "n" is the destination port#. For example:

```
:MESSAGE USER-2 HELLO THERE <cr>
```

The maximum message size is 70 characters. Anyone currently active on the account referenced, or at the port specified, will receive the message. The message is displayed on line 25 of his terminal in the form:

```
From n: text-string
```

where 'n' is the port from which the message was sent.

For example, the above message might be received as:

```
From 5: HELLO THERE
```

If the specified user is not presently logged on, the system responds with [337]NO MESSAGE SENT. If your account has system privilege level two, you may broadcast to all users by substituting an asterisk (*) for the account-name. For example:

```
:MSG * GOOD MORNING CAMPERS <cr>
```

CLEAR-MSG

This verb clears any messages shown on line 25 of your terminal. Its general form is:

```
:CLEAR-MSG
```

WHO

The WHO verb returns the line number and account-name to which you are logged on. The general form is:

```
WHO {n}
```

where 'n' specifies a line number. If specified, line 'n' and the name of the account logged on to line 'n' are displayed. The account name is obtained by looking up the PCB FID in the ACC file and returning Attribute

1. If the account file item is not found, "UNKNOWN" is returned as the account name.

WHAT

The WHAT verb causes current system configuration parameters to be displayed. This verb is usually only available on the SYSPROG account. The general form is:

```
WHAT
```

Figure A shows examples of these verbs.

```
-----
:MSG JOE ARE YOU FINISHED WITH THE TAPE? <cr>

:WHO <cr> <----- You are on line 22 and
22 SYSPROG                logged on to SYSPROG
                        account.

:WHO 3 <cr> <----- User on line 3 is
3 GARY                logged on to account
                        GARY.

:WHAT <cr>
5037 IS THE SYSTEM SERIAL NUMBER
132  IS THE PROM MICROCODE VERSION NUMBER
6    IS THE WRITABLE CONTROL STORE MICROCODE VERSION NUMBER
1024K OF MEMORY
1535 ABS FRAMES
48   ACTIVE LINES
49   IS THE LINE TO WHICH THE SPOOLER IS ASSIGNED
1536 IS THE FRAME ID OF THE PCB FOR LINE ZERO
21849 IS THE FRAME ID OF THE BASE OF THE SYSTEM DICTIONARY
545591 IS THE MAXIMUM FRAME ID FOR THIS SYSTEM

DISC DRIVE CONFIGURATION:
      CONTROLLER          CHANNEL
      08                  01
      08                  03
-----
```

Figure A: Examples of MSG, WHO, and WHAT Verbs

SPECIAL SYSTEM UTILITIES

INFORMATION: TIME, SET-TIME, SET-DATE AND DATE-FORMAT VERBS

The TIME, SET-TIME, SET-DATE and DATE-FORMAT verbs provide miscellaneous system information and capabilities. Figure A shows examples of these verbs.

TIME

The system time and date can be displayed on the terminal by entering the TIME verb.

SET-TIME

The SET-TIME verb changes the current system time. This verb is usually only available on the SYSPROG account. The general form is:

```
SET-TIME hh:mm{:ss}
```

where "hh:mm" (seconds optional) is the time in 24-hour format.

SET-DATE

The SET-DATE verb changes the current system date. This verb is usually only available on the SYSPROG account. The general form is:

```
SET-DATE ddmmm{yy} or mm/dd{/yy}
```

where "dd" is the numeric day; "mmm" is the three-letter month abbreviation (or two digit month number in the second form); "yy" is the optional year specification. If not used, the current year displayed by the system is used.

DATE-FORMAT

The DATE-FORMAT verb changes the form of dates printed with ENGLISH date conversions. The general form is:

```
DATE-FORMAT {(I)}
```

If the 'I' option is used, all dates normally displayed as mm/dd/yy are printed in European (international) format as dd/mm/yy, and the system assumes that external dates entered in formats such as mm/dd/yy are to be interpreted as dd/mm/yy. The date format may be reset by entering DATE-FORMAT (without an option).

```
:TIME <cr>  
09:42:37 12 FEB 1983  
  
:SET-TIME 14:23 <cr>  
14:23:00 12 FEB 1983  
  
:SET-DATE 13 FEB 1983 <cr>  
14:23:11 13 FEB 1983  
  
:DATE-FORMAT (I) <cr>  
  
:DATE-FORMAT <cr>
```

Figure A: Examples of TIME, SET-TIME, SET-DATE and DATE-FORMAT Verbs

SPECIAL SYSTEM UTILITIES

INFORMATION: WHERE AND SLEEP VERBS

| The WHERE and SLEEP verbs provide miscellaneous system information and |
capabilities. Figure A shows examples of these verbs.

WHERE

The WHERE verb (which is normally only available on the SYSPROG account) causes currently logged-on user assignments to be displayed. The general form is:

```
WHERE {n} {(options)}
```

The display produced without use of the 'S' option shows line number, PCB FID, PIB road block bytes (bytes 0 and 3), followed by register one and the software return stack. An asterisk (*) next to an entry indicates assignments for the line which issued the verb.

If "n" is specified, user assignments for line "n" only are displayed (in the absence of "n" all logged-on users are shown).

The 'S' option displays the number of process activations, CPU milliseconds, number of disk reads, PIB Status bytes zero and three (the roadblock bytes) and LINKS (consisting of PIB bytes 1 (debug byte), 4 (next PIB), 5 (previous PIB) and 1C (lock #)).

The 'A' option displays all lines including those not currently logged on; a question mark (?) is printed next to ports that are logged off but do not have a status 'FBFF' (an error condition).

The 'N' option inhibits automatic paging of output to the terminal.

The 'P' option directs the report to the printer.

SLEEP

The SLEEP verb sets a process to "sleep" for a specified period of time. The mechanism that allows a process to "sleep" is implemented in the MONITOR so that a "sleeping" process creates practically no overhead in the system. The "sleep" mechanism is useful for processes that must remain quiescent for a period of time. The form of the SLEEP verb is:

```
SLEEP n  
or  
SLEEP hh:mm{:ss}
```

where "n" is the number of seconds the process should sleep, or where "hh:mm" (seconds optional) is the time (in 24-hour format) when the process is to regain control. In either case, the terminal's BREAK function may be used to wake the terminal.

```
:WHERE <cr>
```

PORT	PCBFID	PS	RTN	STACK..		
*000	000600	FFFF	121.019	1275.167	1275.085	
001	000620	FBFF	6.085	6.044	5.057	
003	000660	FBFF	6.085	6.044	181.101	
004	000680	FBFF	6.085	6.044	181.101	
008	000700	FBFF	6.085	6.044	181.101	
013	0007A0	FBFF	6.085	274.179	269.185	
049	000C20	BFFF	164.084			

```
:WHERE (AS <cr>
```

PORT	PCBFID	PS	CPU	MSEC	# ACTS	READS	LINKS
*000	000600	FFFF		3362	403	128	00 09 FF FF
001	000620	FBFF		18396	1873	222	00 FF 0C FF
003	000660	FBFF		1439	116	28	00 05 08 FF
004	000680	FBFF		3765	334	21	00 0D 08 FF
012	000780	FBFF		6108	529	44	00 01 02 FF
013	0007A0	FBFF		57155	4790	480	00 31 04 FF
049	000C20	FFFF		1251437	923314	90845	00 08 0D FF

```
:WHERE 3 <cr>
```

PORT	PCBFID	PS	RTN	STACK..		
003	000660	FBFF	6.085	6.044	181.101	

```
:SLEEP 10 <cr>
```

```
:SLEEP 11:32 <cr>
```

Figure A: Examples of WHERE and SLEEP Verbs

SPECIAL SYSTEM UTILITIES

INFORMATION: LIMITS, USER AND DISCIO VERBS

The LIMITS, USER and DISCIO verbs provide miscellaneous system information and capabilities.

LIMITS

The LIMITS verb displays current system configuration parameters, in an identical format to that produced by the WHAT verb (see earlier topic). The general form is:

LIMITS

USER

The USER verb displays the system privilege level and number of additional logon work space frames (per linked work space area) reserved for the line which executed this verb. The general form is:

USER

DISCIO

The DISCIO verb creates a 5-second "snapshot" disk activity report. Information displayed includes:

1. Disk I/Os per second for each unit
2. Total disk I/Os per second
3. Number of disk reads per second
4. Number of disk write/verify operations per second
5. Cumulative disk I/Os per unit
6. Cumulative disk I/Os
7. Cumulative disk reads
8. Cumulative disk write/verify operations

where "cumulative" means the total number processed since the start of DISCIO execution. The general form is:

DISCIO {(options)}

If a numeric option is used, the display is repeated the specified number of times. The 'T' option displays the current total disk I/O counters for each unit (i.e., the number of disk I/O operations since the last coldstart). The numeric option cannot be used with the 'T' option. The 'P' option produces the report on the printer.

Figure A shows usage of the USER and DISCIO verbs.

```

:USER <cr>
2   SYSTEM PRIVILEGE LEVEL
50  ADDITIONAL FRAMES OF WORK SPACE

```

```

:DISCIO <cr>

```

DEVICE	I/O'S PER SECOND	TOTAL I/O'S
D0	6.17	32
D1	7.13	37
D2	7.22	40
D3	5.55	28
TOTAL:	26.1	137
READS:	16.1	77
WRITES:	10.0	60

```

:DISCIO (T) <cr>

```

DEVICE	DISC UNIT I/O'S
D0	351762
D1	327797
D2	342781
D3	338765

Figure A: Sample Use of USER and DISCIO Verbs

SETTING PROCESS TIMESLICES: TIMESLICE VERB

This section describes how the system allocates timeslices among various terminals. The TIMESLICE verb allows each line to be assigned a different timeslice to allow adjustment of task priorities. This verb is normally only available on the SYSPROG account.

Each process (terminal) not waiting for an operator entry is placed on a queue of terminals waiting for activation. Activation means that the process is given a timeslice (or quantum) of CPU time in which to do its work.

To determine which job to activate, the system scans from the top of the queue until it finds a job which can be activated. A terminal cannot be activated if:

1. It is waiting for a frame to come in from disk
2. It is "sleeping"

A process which is activated continues to process until:

1. It uses up its timeslice
2. It references a frame which is not in main memory
3. The program voluntarily deactivates
4. The operator is prompted
5. It references a frame to which I/O is being performed by a controller

Priority is determined by a process's position on the queue or "ladder". The position of processes changes as they use various system resources such as disk access and CPU processing.

Normally, each process is assigned a 10 millisecond timeslice. The timeslice for each line is kept in main memory and current setting for each line stays in effect until a coldstart is performed, at which point the settings for all lines are set to 10 milliseconds.

The form of the TIMESLICE verb is:

TIMESLICE {n} {(options)}

where "n" is a number (1-1023) indicating the timeslice in milliseconds. If "n" is omitted, the timeslice is displayed but remains unchanged.

The timeslice for other lines may be changed or displayed from another terminal by specifying the line number in the option. For example, TIMESLICE 50 (3) sets the timeslice to 50 milliseconds on line 3, whereas TIMESLICE (3) displays the current timeslice setting for line 3.

All of the process timeslices may be set or displayed by using the 'A' option.

A point to remember in setting process timeslices is that setting a large timeslice on one or more lines may cause other processes to degrade because they may have to wait longer for processing time. Setting timeslices too low may also cause degradation because processes may get "bumped off" before they get to do any meaningful work.

You are encouraged to experiment with different settings for your job mix, possibly changing timeslices within PROCs that activate certain jobs.

```
-----
TIMESLICE {n} {(options)}
-----
```

Figure A: General Form of TIMESLICE Verb

```
-----
:TIMESLICE <cr> <----- Displays current timeslice.
TIMESLICE FOR LINE 10 EQUALS 100 MILLISECONDS

:TIMESLICE 25 <cr> <----- Sets timeslice to 25
                                milliseconds.

:TIMESLICE (3) <cr> <----- Displays timeslice for line 3.
TIMESLICE FOR LINE 3 EQUALS 75 MILLISECONDS

:TIMESLICE 50 (3) <cr> <----- Sets timeslice for line 3 at 50
                                milliseconds.

:TIMESLICE 75 (A) <cr> <----- Sets timeslice for all lines at
                                75 milliseconds.
-----
```

Figure B: Sample Usage of TIMESLICE Verb

SET PRIVILEGE LEVEL AT A PORT: SET-PRIV VERB

| SET-PRIV sets the privilege level as required at a specified port.
It is usually only executable from the SYSPROG account.

This verb has the general format:

SET-PRIV m {(n)}

where: m is the privilege level to be set (in the range 0 to 2 inclusive)

n is the port number at which this privilege level is to be set
(omission of the parameter implies the port from which the command
is executed)

SYS2 privileges are required to execute this verb.

:SET-PRIV 0 (38 <cr>

Set SYS0 privilege level at port
(process) 38.

:

Figure A : Example of Use of SET-PRIV Verb

AVAILABLE SYSTEM SPACE: POVF VERB

| The POVF verb displays the number of frames in the four "available
space" tables.

POVF

The POVF verb (Figure A) has the following general form:

POVF {(P)}

POVF displays the number of frames in the four "available space" tables. If there are any frames in the linked chain of file space, the first line of output contains the FID of the beginning frame in the chain. The "TO" field contains the word "LINKED" and the "#FRAMES" field contains the total number of frames in the linked chain. Contiguous file space is displayed with the beginning and ending FIDs of each block and the total frames in the block. Lines of output after the linked chain information are in the form:

m - n p m - n p

where:

"m" is the first frame of a contiguous block

"n" is the last frame of the block

"p" is the number of frames in the block

Following the contiguous frame display is the total number of contiguous file space frames. Also listed are the number of frames in the temporary space block and the temporary space chain.

The 'P' option directs output to the printer.

Figure B illustrates usage of the POVF verb.

Note: CLINK, TCLINK and CLAIM verbs are provided to verify and re-organise available space: see Section 10 for further information.

```

POVF {(options)} <----- Prints overflow list
      |
      |-----| P - Output to Printer |
  
```

Figure A: General Form of POVf Verb

:POVF <cr>

FROM	TO	#FRAMES	FROM	TO	#FRAMES
54872 - LINKED		5			
5622 - 5711		90	6427 - 6440		14
25805 - 25806		2	35700 - 35707		8
35712 - 35714		3	35718 - 35725		8
36566 - 36595		30	36680 - 36694		15
36899 - 36916		18	36931 - 36939		9
36959 - 36965		7	37736 - 37737		2
33740 - 37751		12	38557 - 38570		14
38584 - 38596		13	39290 - 39291		2
53936 - 53993		58	54000 - 54007		8
54016 - 54026		11	54033 - 54042		10
54049 - 54228		180	54373 - 54384		12
54413 - 54793		381	54841 - 54853		13
54914 - 54922		9	54924 - 54927		4
54937 - 54937		1	54960 - 54960		1
54969 - 54978		10	55038 - 55045		8
55399 - 55578		180	55957 - 58463		2507

TOTAL NUMBER OF CONTIGUOUS FRAMES = 3640
NUMBER OF FRAMES IN THE TEMPORARY SPACE BLOCK = 0
NUMBER OF FRAMES IN THE TEMPORARY SPACE CHAIN = 31

Figure B: Sample Usage of POVf Verb

GENERATING CHECKSUMS: CHECK-SUM COMMAND

```
-----
| The CHECK-SUM command generates a checksum for file items; from this |
| you can determine if data in a file has been changed.                |
|-----
```

CHECK-SUM is an ENGLISH verb (Figure A) which has the following general form:

```
CHECK-SUM {DICT} file-name {item-list*} {attribute} {selection-criteria}
                                                {(options)}
```

A checksum is an arithmetic total, disregarding overflow, of all bytes in the selected item(s) and is generated for items in the specified file, or subset of items if the optional "item-list" and/or "selection-criteria" appear. Furthermore, the checksum may be calculated for one specified attribute. If no attribute is specified, the first default attribute is used. If there is no default attribute, or if the AMC is 9999, the entire item is included. The checksum includes the binary value of each character times a positional value. This yields a checksum which has a high probability of being unique for a given character string. The dictionary portion is checksummed if the "DICT" option appears. The 'P' option may be used to direct output to the printer.

A message is output, giving checksum statistics, in the following form:

```
BYTE STATISTICS FOR file-name (or attribute name):
TOTAL = t AVERAGE = a ITEMS = i CKSUM = c BITS = b
```

where:

```
"t" is the total number of bytes in the attribute (or item) included
"a" is the average number of bytes per item
"i" is the number of items
"c" is the checksum
"b" is a bit count
```

The attribute mark trailing the specified attribute (or item) is included in the statistics.

Sample output statistics are shown in Figure B.

To use checksums, you should issue CHECK-SUM commands for all files, or portions of files, to be verified and record the output statistics. Subsequently, the CHECK-SUM commands can be reissued to verify that the checksum statistics have not changed. The checksum for a given file must be recalculated whenever you update that file.

- * The item-list can be supplied by a previous SELECT, SSELECT, GET-LIST, FORM-LIST or equivalent command, in which case the selection-criteria are omitted.

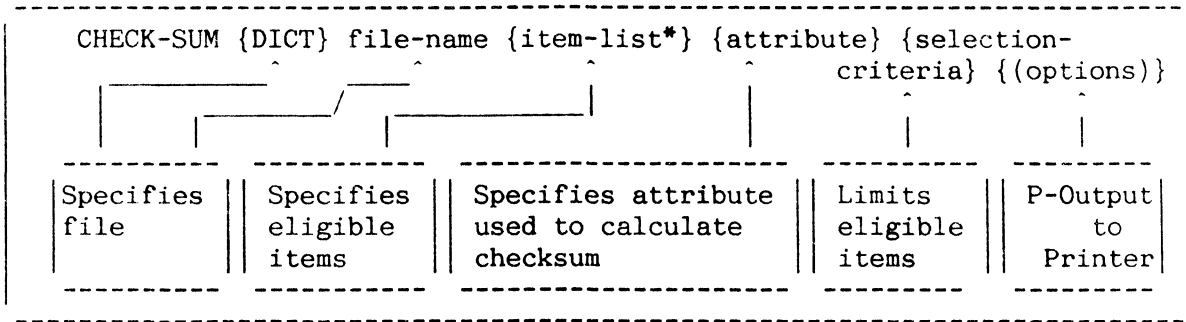


Figure A: General Form of CHECK-SUM Verb

```

:CHECK-SUM TEST <cr>

BYTE STATISTICS OF TEST :
TOTAL = 70 AVERAGE = 7.00 ITEMS = 10 CKSUM = 38444 BITS = 245

:DELETE TEST 13 <cr> <----- Update file
'13' DELETED

:CHECK-SUM TEST <cr>          -----Note changed statistics
                               |
BYTE STATISTICS OF TEST:      v
TOTAL = 63 AVERAGE = 7.00 ITEMS = 9 CKSUM = 35607 BITS = 220

:CHECK-SUM ACCOUNT NEXT-ACCOUNT <cr>

BYTE STATISTICS OF NEXT-ACCOUNT:
TOTAL = 392 AVERAGE = 5.76 ITEMS = 68 CKSUM = 148837 BITS = 1484

:CHECK-SUM DICT ACCOUNT <cr>          (D/Code will be used since
                                     it is a default attribute
                                     -- item '1')
                                     |
                                     v
BYTE STATISTICS OF D/CODE :
TOTAL = 183 AVERAGE = 1.96 ITEMS = 93 CKSUM = 53070 BITS = 888

```

Figure B: Examples of CHECK-SUM Verb

GROUP FORMAT ERRORS: FIX-FILE-ERRORS VERB

THE FIX-FILE-ERRORS verb is designed to assist users in determining the nature of group format errors they have encountered. In many instances the program will recover data which would previously have been lost.

The FIX-FILE-ERRORS verb can be misused: It should be used only by experienced users, and then only after a cautious examination of the situation.

The general form of this verb (Figure A) is:

FIX-FILE-ERRORS {DICT} file-name

You should be logged on to an account with at least 66 as its additional workspace parameter. The TSYM file must be defined on this account.

The FIX-FILE-ERRORS verb operates on either the data or dictionary portion of a file. This operation is performed on each group until all groups have been processed. When possible, the operation will remove data in error and place it in the dictionary of the TSYM file.

Two passes are made through each group. Pass one verifies the frame linkages of the group, and pass two examines the data.

Pass one first checks the links of the primary file space. Errors are reported in the following format:

LINK FIELD ERROR - GROUP n FRAME n

LINKS xx xxxx xxxx xx

If the primary space crosses the maximum FID, the program stops with an appropriate message. After processing the primary space, the operation scans links of secondary frames searching for an incorrect backward link and for the end of group. If an incorrect backward link is found, the program scans (at most) another 66 frames in that group.

Pass two scans the data one item at a time. Items are examined for format errors, described in Figure B. Corrective actions for these errors are also listed in this figure.

If segment marks are encountered, they are replaced with underscore characters (_). A message of the form "SM @ xxxx.yy" is displayed, where xxxx.yy is a frame and displacement. Error items built in the TSYM dictionary are given special item-ids. The item-id is composed of:

Error type, FID of first frame in group, sequential integers

Note that subsequent use of this program or the assembler causes the error items to be lost because TSYM is cleared!

```
FIX-FILE-ERRORS {DICT} file-name
```

Figure A: General Form of FIX-FILE-ERRORS Verb

<u>ERROR TYPE</u>	<u>DESCRIPTION</u>	<u>CORRECTIVE ACTION TAKEN</u>
O	Premature end of data	*
N	Nonhex character in count field	*
C	Count field out of range (was <5 or >32267)	*
A	No attribute mark at end of item	*
I	An item-id exceeds 50 characters	**
H	An item-id does not hash to current group	**
(none)	A segment mark was found in the item	

*A new item is created in the TSYM dictionary and a group-terminating attribute mark is placed after the last good item. Any secondary frames beyond the last item are disconnected from the group.

**Offending items are removed from the file and placed in the TSYM dictionary. The scan continues through the group.

Figure B: Errors Detected and Corrective Action Taken by Pass Two of FIX-FILE-ERRORS

SEARCHING A FILE FOR A CHARACTER STRING: SEARCH AND ESEARCH VERBS

SEARCH and ESEARCH generate a SELECT list of items in a specified file containing (or not containing) one or more specified strings. SEARCH is a TCL-II verb whereas ESEARCH is an ENGLISH verb.

The formats of these commands are:

ESEARCH filename {selection-criteria} {(options)}

SEARCH filename {item-list*} {(options)}

The selection-criteria or item-list may be omitted if supplied by an immediately-preceding SELECT (or equivalent) command.

Selection-criteria are as defined for a standard ENGLISH command: if omitted from the ESEARCH command, all items are implied.

If the item-list (or *, implying all items) is omitted from a SEARCH command, and a SELECT list is not supplied, an item-list is prompted for. Options available are:

- A AND - all strings specified must be present for selection/rejection.
- I Print item-ids as items are selected.
- L Save line numbers - item-id is listed appended with value mark and attribute number, for each attribute containing string(s) (as for exploded SELECT - typically used as input to DATA/BASIC).
- N NOT - list comprises all items in file which would not be selected without this option.
- S Suppress SELECT list ('I' option to list item-ids implied).

For either verb, the character string(s) to search for are repeatedly prompted for as follows:

STRING:
STRING:

until a <cr> is entered. (The total of all strings entered may be up to 500 characters in length.) In the absence of options, the result is a SELECT list of all items in the file containing any of the specified strings.

The illustration opposite shows examples of the use of these commands.

```
:SEARCH BP (AI)<cr>
ITEM-ID:*<cr>
STRING:TEMP<cr>
STRING:<cr>
TEMP1
TEMP2

2 ITEMS SELECTED
>LIST ONLY BP<cr>
TEMP1
TEMP2

:ESEARCH BP (S)<cr>

STRING:TEMP<cr>
STRING:<cr>
TEMP1
TEMP2

2 ITEMS SELECTED

:
```

Figure A : Examples of Use of SEARCH and ESEARCH

SPECIAL SYSTEM UTILITIES

COMPARISON OF FILES: COMP VERB

The COMP verb carries out a detailed comparison of selected (or all) items in one or two specified files, resulting in a report on differences (including mismatched lines).

The syntax of the COMP verb is similar to that of the COPY verb. The general format of the first part of the command is:

```
COMP {DICT} filename item-list* {(options)}
```

where options may be:

- A list all lines in items (not just changed, deleted or inserted lines)
 - B report on each item on a new page
 - C do not compare comment lines (starting with '*')
 - E do not compare end-of-line comments in assembler programs (valid only in conjunction with 'M' option)
 - M compare assembly programs (suppress line counter comparison)
 - N no paging in listing
 - P output to printer (or other currently assigned spooler device)
 - S suppress listing of matching items
 - W suppress wrapping of output lines, and truncate attributes greater than column width.
- n value 'n' overrides default forward search parameter (20) (for no. of lines to search forward in trying to re-align item attributes)

If the item-list is omitted, a prompt is given for an item-list unless supplied by a preceding SELECT, SSELECT etc. command. An asterisk specifies all items.

After this, a prompt is given for the file and/or items to be compared:

```
WITH:{{{DICT} filename}} {item-list}
```

If the filename is not entered to this prompt, an item list must be entered to give a useful result. Each item in the first item list is then compared in turn with the corresponding item, within the same file, from the second item list. Note that if items in the first list are not on file the comparison will go out of sequence.

If a filename is entered without an item list, the items in the first file are compared with items in the second file with the same item-ids. Note that if the second file contains items not in the first file, the files must

be compared a second time in the opposite sequence to detect the additional items.

If both a filename and an item list are entered, the comparison is carried out as for entry of an item list only (except that the second list identifies items in the second file).

The standard comparison report shows each pair of items in turn in two-column format, with the first item in the left-hand column and the compared item in the right-hand column. If all attributes match, the item-ids only are shown. Otherwise, attributes which differ are listed as follows: mismatching attributes are listed for each item, preceded by 'C' (for 'changed'); attributes present in the first item only are listed in the column for the first item, preceded by 'D' (for 'deleted'); attributes present in the second item only are listed in the second column preceded by 'I' (for 'inserted'). Missing items are reported by messages of the form 'item-id' NOT ON FILE, which appear in the left-hand column of the report regardless of the file from which the item is missing.

Note:

Items containing non-printable characters (such as DATA/BASIC object items) should not be compared as the resulting report may produce unwanted effects on the terminal.

```

-----
COMP FILE1 A B<cr>
WITH:(FILE2)<cr>

PAGE 1          2.0 COMPARISON          10:43:41    06 DEC 1985

FILE1 A
FILE1 B
C  1 AN
D  2 ITEM

FILE2 A
FILE2 B
C  1 AN ITEM

COMP FILE1 * (A)<cr>
WITH:(FILE2)<cr>

PAGE 1          2.0 COMPARISON          10:44:27    06 DEC 1985

FILE1 A
  1 AN ITEM
  2 WHICH
  3 IS THE
  4 SAME

FILE2 A
  1 AN ITEM
  2 WHICH
  3 IS THE
  4 SAME

FILE1 B
C  1 AN
D  2 ITEM
  3 WITH

  4 DIFFERENCES

FILE2 B
C  1 AN ITEM

  2 WITH
I  3 SOME
  4 DIFFERENCES
-----

```

Figure A : Examples of Use of COMP Verb

SPECIAL SYSTEM UTILITIES

FILE MANAGEMENT PROCS: DELETE AND SET-FILE

```
-----  
DELETE and SET-FILE are two PROCS available for file management.  
DELETE deletes one or more items from a specified file. SET-FILE  
sets up a Q-pointer, with item-id QFILE, to a specified file.  
-----
```

DELETE PROC

This PROC is evoked by typing:

```
DELETE file-name {item-id}
```

This PROC deletes a specified item. Alternatively, a list of items to be deleted may be supplied by an immediately-preceding SELECT, SSELECT, GET-LIST, FORM-LIST or equivalent command. If no item is specified, the message "NEXT ITEM" prompts you for an item-id. A carriage return ends the input prompting.

SET-FILE PROC

The SET-FILE PROC provides a quick way to set up a file synonym definition item (Q-pointer) to any file in the system. The general form is:

```
SET-FILE {account-name {file-name}}
```

where "account-name" is the name of the logon account the file is defined in, and "file-name" is the name of the file that a Q-pointer is to be set up for. SET-FILE always returns the message:

```
QFILE UPDATED
```

indicating that an item called "QFILE" has been set up in the M/DICT as a Q-pointer to the required file. For example:

```
:SET-FILE SYSPROG XSYM <cr>  
'QFILE UPDATED'
```

This example sets up QFILE to the XSYM file in SYSPROG.

Typing SET-FILE with one or both parameters missing will cause a request for that parameter:

```
:SET-FILE <cr>  
ACCOUNT-NAME?SYSPROG <cr>  
FILE-NAME?XSYM <cr>  
'QFILE UPDATED'
```

This example causes the same action as the previous example. Typing a null line to either the request for "ACCOUNT-NAME?" or "FILE-NAME?" leaves the corresponding name in the 'QFILE' item unchanged from its previous value.

Figure A shows the general form of DELETE and SET-FILE PROCS.

```
-----  
DELETE file-name {item-id} <----- Deletes item(s)  
SET-FILE {account-name {file-name}} <----- Creates Q-pointer  
-----
```

Figure A: General Form of DELETE and SET-FILE PROCs

PRINTRONIX PRINTER ADJUSTMENTS: PRINTRONIX UTILITY

A catalogued DATA/BASIC program called "PRINTRONIX" enables you to change the form length and number of lines per inch on a Printronix printer.

This utility is executed by typing "PRINTRONIX" at TCL. Upon its execution, you are prompted for the form length, number of lines per inch, and channel specifications. After you enter your responses to the prompts, the utility transmits the necessary control to the printer corresponding to the current spooler assignment.

NOTE:

The data is sent to the printer as a "normal" print job (i.e., job is placed in the job queue). Therefore, any changes caused by the PRINTRONIX utility do not take effect until the other print jobs ahead of it are printed. However, you have the option of changing the priority of print jobs via SP-JOBS Action Code 10 or the SP-PRIORITY verb.

The PRINTRONIX utility also allows you to equate IBM "channel" numbers to a particular line number. For example, if Channel Three is equated to Line Six, the printer moves to the sixth line when a "Skip to Channel 3" command is sent to the printer. This feature is used only when receiving print format data from an IBM system via the 5750 Communications Terminal.

On the adjacent page you see two examples of PRINTRONIX usage. In Figure A, the user is specifying a form length of 11 inches and 88 lines per page (11 x 8). Note that the "8 LPI" button must still be pressed if eight lines per inches are desired.

In Figure B, a user is specifying a form length of 11 with 66 lines per page. In addition, two channel specifications are made. When this utility is executed, a "Skip to Channel 3" command moves the paper to the sixth line; a "Skip to Channel 4" command moves the paper to Line 33. Note that Channels 1 and 2 are reserved.

```
-----  
:PRINTRONIX <cr>  
ENTER FORM LENGTH IN INCHES?11 <cr>  
IS THIS 6 OR 8 LINES PER INCH?8 <cr>  
VFU CHANNEL SKIPS (Y/N)? N <cr>  
NOW SET PRINTER TO 6 OR 8 LPI AND ADJUST FORMS IN PRINTER.  
-----
```

Figure A: Sample Usage of PRINTRONIX Utility (Without Channel Specification)

```
-----  
:PRINTRONIX <cr>  
ENTER FORM LENGTH IN INCHES?11 <cr>  
IS THIS 6 OR 8 LINES PER INCH?6 <cr>  
VFU CHANNEL SKIPS (Y/N)? Y <cr>  
CHANNEL 1 IS USED AS TOP-OF-FORM  
CHANNEL 2 IS USED AS DUMMY CHANNEL  
CHANNEL, LINE#  
?3,6  
?4,33  
? <cr>  
NOW SET THE PRINTER TO 6 OR 8 LPI AND ADJUST FORMS IN PRINTER.  
-----
```

Figure B: Sample Usage of PRINTRONIX Utility (With Channel Specification)

SPECIAL SYSTEM UTILITIES

MISCELLANEOUS PROCS/VERBS

This topic describes various utility PROCs and verbs.

CT PROC

This PROC is evoked by typing:

```
CT file-name item-list* or * {(options)}
```

The item(s) specified are copied to the terminal. Options recognized by the COPY verb may be added.

CP PROC

This PROC is evoked by typing:

```
CP {file-name} {item-list* or *} {(options)}
```

The item(s) specified are copied to the printer. If the file-name and/or items are not specified, they are prompted for. Options recognized by the COPY verb may be added (see Section 6).

CHOO-CHOO PROC

The CHOO-CHOO PROC is evoked by typing:

```
CHOO-CHOO
```

This PROC prints a picture of a choo-choo train on your terminal (just for fun).

LISTACC PROC

The LISTACC PROC is evoked by typing:

```
LISTACC {account-names} {LPTR} {NOPAGE} {(options)}
```

This PROC lists accounting data for the account-names specified. If no account-names are specified, accounting data for all users are listed. Options N (output to terminal without paging) or P (output to printer) may be included. Output is paged to terminal unless defeated.

LISTCONN PROC

The LISTCONN PROC is evoked by typing:

```
LISTCONN {file-name} {LPTR} {NOPAGE} {(options)}
```

This PROC sorts all connectives in the M/DICT or optionally specified file (options as for LISTACC).

LISTDICTS PROC

The LISTDICTS PROC sorts all attribute or attribute synonym definition items in the M/DICT or the optionally specified file. The general form is:

```
LISTDICTS {file-name} {LPTR} {NOPAGE} {(options)}
```

LISTFILES PROC

The LISTFILES PROC sorts all file and file synonym definition items in the M/DICT or optionally specified file. The PROC displays the D/CODE of the file entry, the MODULO, SEPARATION, and BASE FID of the dictionary portion of the file, the number of ACCESS and RETRIEVAL codes, any correlatives, the output type (l, r, t, etc.), and the output length. The last three operate on the item-ids of the data items during ENGLISH output. The general form is:

```
LISTFILES {file-name} {LPTR} {NOPAGE} {(options)}
```

LISTPF PROC

The LISTPF PROC sorts the POINTER-FILE and produces a report displaying the account-names, types of catalogued items (C=catalogued DATA/BASIC program; L=save-list of item-ids/data), the names of the programs or lists, the FID of the first frame of the information, the number of frames occupied, the number of item-ids (or attributes of catalogued data) for save-lists, and the catalogue date. The general form is:

```
LISTPF {LPTR} {NOPAGE} {(options)}
```

LISTPROCS PROC

LISTPROCS sorts all PROCs in the PROCLIB file or optionally specified file. The general form is:

```
LISTPROCS {file-name} {LPTR} {NOPAGE} {(options)}
```


SPECIAL SYSTEM UTILITIES

MISCELLANEOUS PROCS/VERBS

This topic describes additional utility PROCs and verbs.

LISTU PROC

The LISTU PROC lists the account names of all users currently active on the system, along with their PCB-FID, location, logon time and channel number.

The location field is stored as attribute 1 in items with item-ids equivalent to the two digit channel number in the dictionary of the ACC file.

LISTVERBS PROC

The LISTVERBS PROC sorts all verbs (not PROCs) in the M/DICT or optionally specified file. The general form is:

```
LISTVERBS {file-name} {LPTR} {NOPAGE} {(options)}
```

LP80, LP106, LP136 PROCs

The LP-type PROCs set the terminal characteristics so printer listings will be formatted for paper of the following sizes:

LP80	8-1/2" wide by 11" long
LP106	11" wide by 8-1/2" long
LP136	14" wide by 8-1/2" long

SPACE PROC

The SPACE PROC may be used to perform multiple T-FWD commands to skip over a specified number of tape files. You must first attach the tape with the ASSIGN or T-ATT verb. To execute this PROC, enter the word SPACE and the system will prompt for the number of files to T-FWD over.

COMPARE VERB

The COMPARE verb, which invokes a catalogued DATA/BASIC program, compares items in two different files. Output includes the number of items in each file, a list of items that do not compare, and a list of items that are in one file but not the other. To execute, enter COMPARE. The program prompts for the two file-names and asks if output is to be to the terminal or printer. For details of the COMP verb, which carries out a detailed comparison of items within one or more files, see previous topic within this section.

CHECK PROC

The CHECK PROC is similar to the SPACE PROC except that multiple T-CHKs are performed, checking a number of sequential tape files for parity errors. You must first attach the tape with the T-ATT or ASSIGN verb.

Figure A illustrates the sample usage of SPACE, COMPARE and CHECK PROCs.

```

:SPACE <cr>
SPECIFY NUMBER OF TAPE FILES TO T-FWD
MUST BE TWO DIGITS=03 <cr>
03
[94] END OF FILE
02
[94] END OF FILE
01
[94]END OF FILE
...YOU ARE SPACED OUT...

:COMPARE <cr>
ENTER FIRST FILE TO COMPARE ([DICT] FILE-NAME):AA <cr>
ENTER SECOND FILE TO COMPARE:BB <cr>
DO YOU WANT OUTPUT TO THE PRINTER (Y/N):N <cr>

FILE COMPARISON BETWEEN 'AA' AND 'BB'
PAGE 1      13:15:32      12 FEB 1983

-----

ITEM COUNTS: 'AA':36      'BB':36

-----

123-AA DOES NOT MATCH

1632-ES IS MISSING FROM BB

8821-GG IS MISSING FROM AA

---FINISHED---

:CHECK <cr>
SPECIFY NUMBER OF TAPE FILES TO T-CHK
MUST BE TWO DIGITS=02 <cr>
02
[94] END OF FILE
01
[94] END OF FILE
...YOU ARE CHECKED OUT...

```

Figure A: Sample Usage of SPACE, COMPARE, and CHECK PROCs.

SOURCE COMPACTION: ECOPY, EEDIT (EED) AND EBASIC VERBS

| This topic describes the system capability to maintain source programs |
in a compacted form.

You may compact source code (for DATA/BASIC programs) so that the source programs take less file space. The compaction is achieved with only a slight decrease in processing speed during editing, compiling, etc. Compaction is achieved by replacing any sequence of three or more blanks or asterisks with a delineator, a count of the number of characters compacted, and the character. The delineator used is the data link escape character (DLE, X'10').

The compaction takes place when an item is written into a file using the 'FI' command of the EDITOR when invoked via the EEDIT verb (or its synonym EED). This processor is the only one which is able to compact data. The EDITOR can also expand the data during the time it is being edited (i.e., while in the IS and OS work areas). In addition to the EDITOR, the DATA/BASIC compiler and COPY evoke the expansion feature. Note: The assembler does not recognize this feature.

To use this feature, you must specify one of these alternative verbs in your TCL statement.

Note that the ECOPY verb will expand items as they are copied. Since this is usually not what is desired, you should, in general, use the normal COPY verb when copying compressed items.

10. SYSTEM MAINTENANCE PROCEDURES: SYSPROG ACCOUNT

=====

SYSTEM PROGRAMMING (SYSPROG) ACCOUNT

Several special facilities are normally used from the System Programmer (SYSPROG) account. Procedures on this account are normally performed by persons more familiar with the overall operation of the system.

To log on to the SYSPROG account, when the message (or similar prompt):

LOGON PLEASE:

is displayed, type:

SYSPROG{,password} <cr>

where "password" is the appropriate password set up for SYSPROG. Alternative logon names, such as SP, which has no additional work space frames, may be used.

Some SYSPROG verbs described in this section are:

:ABSLOAD	LIST-OVF-ERRORS
:ABS/FILES	LIST-SYS-ERRORS
ACCOUNT-RESTORE	LOCK-FRAME
BUFFERS	SAVE
CHARGE-TO	SEL-RESTORE
CLAIM	SET-WRITES
CLEAR-ITEM-LOCKS	SIZE-ITEM-LOCKS
CLINK	STRIP-SOURCE
DE-COPY	TCLINK
DSKAD	UNLOCK-FRAME
:FILES	
:IL-NEWTAB	
LIST-DISK-ERRORS	
LIST-ITEM LOCKS	

Some SYSPROG PROCs described in this section are:

ACCOUNT-SAVE	SYS-GEN
ALL-FILE-STATS	SYS-UPDATE
CREATE-ACCOUNT	SYSTEM-SETUP
FILE-SAVE	UPDATE-ACCOUNT
M-A-R	VERIFY-SYSTEM
M-A-S	
SETUP-ASSY	

MINIMIZING DATA LOSS DURING RESTORES: SET-WRITES VERB

The SET-WRITES verb flushes write-required memory buffers. Use of this verb minimizes data loss during a coldstart and ABS restore.

Because of the system's large main memory size, it is possible that the system is waiting an inordinate amount of time before copying write-required buffers to disc. This may result in a loss of data in the event a coldstart or ABS restore is performed. To avoid such losses, the SET-WRITES verb was designed to minimize the total number of write-required buffers at any one time by forcing a write-required memory buffer to be written to disk after a specified number of port activations. These buffers are flushed in sequential order (i.e., after the first "set" of activations, the first write-required buffer is flushed; after the second "set of activations, the second write-required buffer is flushed, etc.).

The format of the SET-WRITES verb is:

SET-WRITES k

where 'k' is a decimal number between 1 and 2,147,483,646 that specified the number of activations between each disk write. This number is reset to a default value of 2,147,483,646 when a coldstart is performed.

Listed below are some guidelines you should consider before using the SET-WRITES verb:

1. If your system rarely needs to be coldstarted, the SET-WRITES verb is probably not needed.
2. If your system is coldstarted regularly and
 - a. if you run large applications (large in terms of main memory usage) and key in small amounts of data, set 'k' to a high number, or
 - b. if you run "tight" programs (those that use a minimum of main memory) and key in large amounts of data, set 'k' to a low number.

It is not possible to apply a fixed "correct" value to the SET-WRITES verb because you begin to see degradation of response time as 'k' is lowered. The number of users on the system at any given time, the type of processing, and the frequency of coldstarts are all considerations. Thus, a value judgement must be made by the person responsible for the system.

A basic setting used at many system sites is a value of 20,000. At the very least, this can be a starting point. The number can be varied at different times of day, or varied on a daily basis. Use of the BUFFERS verb with the 'T' option gives you the total number of write-required buffers.

SYSTEM MAINTENANCE PROCEDURES: SYSPROG ACCOUNT

The following restrictions apply to the SET-WRITES verb:

1. The SET-WRITES verb must be executed from SYSPROG.
2. You must have SYS2 privileges.

To better understand the way that a given SET-WRITES setting may affect the operation of your system, we suggest you use one of the following formulae: 1) calculate how long it will take to cycle through main memory with a given SET-WRITES setting or 2) calculate the correct SET-WRITES setting to achieve one complete cycle through main memory in a given amount of time. Both formulae are described and illustrated in the following two topics.

Figure A illustrates the format of the SET-WRITES verb.

```
-----  
|                               SET-WRITES k                               |  
-----
```

Figure A: Format of SET-WRITES Verb

USING SET-WRITES FORMULA #1

Formula #1 calculates a SET-WRITES setting ('k'). This topic describes the equation to calculate 'k' and illustrates the use of the formula in Figure A.

Before you can use either Formula #1 or #2, a series of system-dependent variables must be defined:

- ATS = Average of timeslice values of system processes
- KBUF = 2 * total megabytes of main memory
- NDISC = total number of disk drive units
- AVXRT = Average disk I/O transfer rate (i.e., AVXTR represents an approximation of the maximum throughput, in frames/buffers per second, that the disk sub-system can handle), to be calculated by the following formula: $AVXRT = (50 - (NDISC * 5)) * NDISC$
- MINT = Minimum cycle time, in seconds, in which KBUF can be flushed to disc; use this formula: $MINT = (KBUF / AVXRT) * 1000$
- TM = Actual Cycle Time, in seconds. This will either be the time it takes for one complete cycle through main memory, or the amount of time that a complete cycle through main memory will take with a given SET-WRITES value. Note that TM is simply a time value in seconds that cannot be less than MINT.
- k = The value of SET-WRITES.

FORMULA #1: DETERMINING A SET-WRITES SETTING ('k')

This formula calculates the value for the SET-WRITES setting ('k') when it is desired that SET-WRITES make a full cycle through memory in a given time. To ensure that the time given (TM) does not exceed the saturation point of the disk sub-system transfer rate (AVXRT), TM cannot be less than MINT.

To calculate 'k', follow these steps:

1. Solve for AVXRT, then MINT.
2. Convert the desired cycle time to seconds. This is TM. If TM is less than MINT, then round TM upward to MINT (TM cannot be less than MINT).
3. Now solve for 'k': $k = TM / (ATS * KBUF)$

Parameters used in this example:

1 Megabyte of main memory KBUF = 2*1 = 2
 10 Milliseconds average timeslice: ATS = 10
 2 disk drives: NDISC = 2

STEP 1. Calculate AVXRT and MINT:

$$\begin{aligned} \text{AVXRT} &= (50 - (\text{NDISC} * 5)) * \text{NDISC} \\ &= (50 - (2 * 5)) * 2 \\ &= (50 - 10) * 2 \\ &= 80 \text{ (transfers per second)} \end{aligned}$$

$$\begin{aligned} \text{MINT} &= (\text{KBUF} / \text{AVXRT}) * 1000 \\ &= (2 / 80) * 1000 \\ &= .025 * 1000 \\ &= 25 \text{ (seconds)} \end{aligned}$$

STEP 2. Calculate the SET-WRITES setting ('k') from a given time of TM.

For this example, let's say that we want to cycle through memory every 30 minutes.

$$\begin{aligned} \text{TM} &= 30 \text{ minutes} * 60 \text{ seconds} \\ &= 1800 \text{ seconds} \\ \text{NOTE: If } \text{TM} < \text{MINT} \text{ then } \text{TM} &= \text{MINT}. \end{aligned}$$

$$\begin{aligned} k &= \text{TM} / (\text{ATS} * \text{KBUF}) \\ &= 1800 / (10 * 2) \\ &= 1800 / 20 \\ &= 90 \end{aligned}$$

Thus, with the given parameters and the desire to cycle through memory every 30 minutes, the SET-WRITES setting would have to be 90.

USING SET-WRITES FORMULA #2

Formula #2 calculates the amount of time it takes for a full cycle-through of main memory given a SET-WRITES value of 'k'. This topic describes the equation to calculate TM and illustrates the use of this formula in Figure A.

To calculate TM, follow these steps:

1. Solve for TM as follows:

$$TM = k*ATS*KBUF$$

NOTE: If $TM < MINT$ then $TM = MINT$.

2. Use TM as the number of seconds it will take to perform a full cycle-through of main memory with a SET-WRITES setting of 'k'.

NOTE:

If the system goes into an idle state, where the condition is such that virtual processes are not being activated for some period of time, all buffers in main memory that are in a write-required state will be written out to disk by the system Monitor. The elapsed time over which this process will be completed is equal to or less than the value calculated for the MINT variable. There is a high probability that this process will take less time than MINT depicts because the value calculated for AVXRT is based on an I/O transfer rate for the disk sub-system that tends to be on the modest side.

Parameters used in this example:

1 Megabyte of main memory	KBUF = 2*1 = 2
10 Milliseconds average timeslice:	ATS = 10
2 disk drives:	NDISC = 2

STEP 1. Calculate AVXRT and MINT:

$$\begin{aligned}
 \text{AVXRT} &= (50 - (\text{NDISC} * 5)) * \text{NDISC} \\
 &= (50 - (2 * 5)) * 2 \\
 &= (50 - 10) * 2 \\
 &= 40 * 2 \\
 &= 80 \text{ (transfers per second)}
 \end{aligned}$$

$$\begin{aligned}
 \text{MINT} &= (\text{KBUF} / \text{AVXRT}) * 1000 \\
 &= (2 / 80) * 1000 \\
 &= .025 * 1000 \\
 &= 25 \text{ (seconds)}
 \end{aligned}$$

STEP 2. Calculate the approximate time it would take SET-WRITES to enforce one complete cycle-through of main memory given the SET-WRITES setting ('k').

Let's say that the SET-WRITES setting 'k' is 10,000.

$$\begin{aligned}
 \text{TM} &= k * \text{ATS} * \text{KBUF} \\
 &= 10,000 * 10 * 2 \\
 &= 100,000 * 2 \\
 &= 200,000 \text{ seconds} \\
 \text{NOTE:} & \text{ If } \text{TM} < \text{MINT} \text{ then } \text{TM} = \text{MINT}.
 \end{aligned}$$

STEP 3. Convert TM (the actual time in seconds) to minutes and/or hours:

$$\begin{aligned}
 \text{CYCLE TIME IN MINUTES} &= \text{TM} / 60 \\
 &= 200000 / 60 \\
 &= 3333.3333 \text{ minutes}
 \end{aligned}$$

Thus, with the given system parameters and the SET-WRITES setting of 10,000, it would take 3,333 minutes, or a little over two days (55.5555 hours), for SET-WRITES to cycle-through memory one time.

Figure A: Sample Use of SET-WRITES Formula #2

ACCOUNT-SAVE PROC

The ACCOUNT-SAVE PROC on SYSPROG allows you to save one account on tape.

This PROC functions similarly to the FILE-SAVE PROC, except that the 'I' option of the SAVE verb is used and no coldstart or ABS section is created. The general form of the input line is:

ACCOUNT-SAVE

The utility first prompts for the tape record block length:

Tape record block length, if other than 8192 ? .

Reply with an alternate block length, or <cr> to use 8192.

If there is more than one tape unit on the system, the utility then prompts for the tape units to be used:

Enter tape units desired, in sequence, separated by commas.
Maximum two half inch (1,2) or one quarter inch (3) ?

Reply with the unit number(s) of the tape units to be used.

A tape label is prompted for as follows:

Tape label if desired

Enter any text required to be included in the tape label.

The utility then prompts for the account to be saved:

ACCOUNT NAME=?

Respond to this prompt with an account name represented by a D-pointer in the SYSTEM Dictionary. No synonym Q-pointers in the SYSTEM Dictionary are saved.

The file \$STAT-FILE*n (where n is the port from which the PROC is run) is cleared and loaded with statistics for the specified account. A T-DUMP of this file is appended and the file statistics report is printed.

Figure A illustrates the ACCOUNT-SAVE PROC.

```

-----
:ACCOUNT-SAVE <cr>
13:21:25 19 SEP 1984
Tape record block length, if other than 8192 ? <cr> <--- Produces default
                                         of 8192 bytes

Enter tape units desired, in sequence, separated by commas.
Maximum two half inch (1,2) or one quarter inch (3)? 2<cr>
[1021] DEVICE(S) TAPE 2 NOW ASSIGNED,
      RECORD SIZE = 8192

Tape label if desired ACCOUNTS PAYABLE <cr> <--- Specifies tape label
                                         identification
. . .   START ACCOUNT SAVE   . . .

ACCOUNT NAME = ?ACCP <cr> <----- Account name
  ACCP
  :
  :
[99] 85 TAPE RECORDS WRITTEN.
. . .   END OF ACCOUNT SAVE   . . .
. . .   DUMPING STAT-FILE     . . .

67 ITEMS DUMPED
. . .   PREPARING FILE STATISTICS REPORT . . .

:
-----

```

Figure A: Dialogue for ACCOUNT-SAVE PROC

SYSTEM MAINTENANCE PROCEDURES: SYSPROG ACCOUNT

MULTIPLE ACCOUNT SAVE

| The M-A-S PROC allows you to save a number of accounts to tape. One or |
more of these accounts may be restored subsequently using the M-A-R PROC.

Accounts saved using this utility should be restored using the M-A-R (Multiple Account Restore) utility. The user may either specify a prestored list of accounts or may enter the account names.

The general form of this command is:

M-A-S

The utility first prompts for the tape record block length:

Tape record block length, if other than 8192?

Reply with an alternate block length or <cr> to continue.

If there is more than one tape unit on the system, the utility then prompts for the unit(s) to be used:

Enter tape units desired, in sequence, separated by commas.
Maximum two half inch (1,2) or one quarter inch (3) ?

Reply with the unit number(s) of the tape units to be used.

The user is then asked if statistics are required for each account:

Do you wish to generate and list stats for each account (Y/<ret>)?

Reply Y <cr> or just <cr> as required.

The utility now prompts for a prestored list name:

Enter prestored list name or return -

If there is a prestored list of the account names that are to be saved then enter its name. If there is no prestored list then reply with <cr>.

If there is no prestored list the following prompt is given:

Enter Account name or return

This prompt will appear repeatedly after each account name is entered until a <cr> reply is received.

The utility now proceeds with the Account Saves.

```
-----  
Multi Account Save  
  
This utility will produce a tape containing a series of account saves  
which may be restored using the M-A-R (Multi Account Restore) utility.  
  
You may specify a prestored list of accounts or enter the account names.  
  
Tape record block length, if other than 8192 ? <cr>  
Enter tape units desired, in sequence, separated by commas.  
Maximum two half inch (1,2) or one quarter inch (3)? 1<cr>  
[1021] DEVICE(S) TAPE 1 NOW ASSIGNED,  
RECORD SIZE = 8192  
  
Do you wish to generate and list stats for each account (Y/<ret>) ?Y <cr>  
  
Enter prestored list name or return - <cr>  
Enter Account name or return DUMMY <cr>  
Enter Account name or return ATP <cr>  
Enter Account name or return <cr>  
  
1 ITEMS DUMPED  
  
ACCOUNT NAME = DUMMY  
DUMMYFILE  
DL/ID  
[99] 2 TAPE RECORDS WRITTEN.  
  
ACCOUNT NAME = ATP  
ERRORS  
[99] 63 TAPE RECORDS WRITTEN.  
  
Account Save<s> Completed.  
-----
```

Figure A: Sample M-A-S (Multi Account Save) Dialogue

ACCOUNT-RESTORE

| The ACCOUNT-RESTORE verb adds a new account (from tape) to an existing |
system.

ACCOUNT-RESTORE

An account-restore can be performed from a formatted file-save of a whole system or from a save of a single account (Figures A and B). In either case, the entire account is restored and a pointer to the account is created in the System Dictionary. To use ACCOUNT-RESTORE, mount the tape at load point and select one of the following methods, depending on the type of tape being used:

- . For full file-save tapes, follow these steps:
 1. Attach the tape unit(s) as the default channel, specifying the proper record size (if known).
 2. If a bootstrap is present, enter:
 T-FWD <cr>
 3. Now enter:
 T-RDLBL 1 <cr>
 4. If an ABS section is present, enter:
 T-FWD <cr>
 5. Now that the tape is positioned at the files section, type:
 ACCOUNT-RESTORE <cr>
 6. Answer the prompt "ACCOUNT-NAME?" with the name of the account. The account name must be the same name used on the file-save tape.

- . For single account-save tapes, follow these steps:
 1. Attach the tape unit(s) as the default channel, specifying the proper record size (if known).
 2. Enter:
 T-RDLBL 1 <cr>
 ACCOUNT-RESTORE <cr>

The restore will proceed automatically.

Note:

This utility will not allow you to restore an account whose name matches any item in the System Dictionary. This means that you must delete an old account before you can restore another account with the same name.

A 'synonym' segment may be encountered with a base which has not been found on the tape. This happens if a D-pointer on the saved account points to a file on another account (illegal use of multiple D-pointers), or if a 'D' segment on the tape is unrecognizable because of a parity error. In this case no D-pointer is created and the following message is displayed:

'SYNONYM NOT FOUND'

```

:ASSIGN =TAPE 1
[1021] DEVICE(S) TAPE 1 NOW ASSIGNED
      RECORD SIZE = 8192
:T-RDLBL 1 <cr> <----- Reads label preceding account
LABEL = DOC.ACCT]08:01:33  01 JUL 1985

:ACCOUNT-RESTORE <cr> <----- Initiates account-restore
STATUS 11137,17,1
  ERRORS 11154,1,1
    DL/ID 11155,3,1
M/DICT 11137,17,1 (S)
DL/ID 11137,17,1 (S)
TP 11159,1,1
  DL/ID 11160,7,1
PARCEL 11306,3,1
  DL/ID 11309,503,1

: <----- Returns to TCL when finished

```

Figure A: Account-Restore from Account-Save Tape

```

:ASSIGN =TAPE 1 <cr>
[1021] DEVICE(S) TAPE 1 NOW ASSIGNED,
      RECORD SIZE = 8192
:T-FWD <cr> <----- Bypass coldstart section
[94] END OF FILE

:T-RDLBL 1 <cr> <----- Reads label preceding ABS section
LABEL = DOC.ACC]08:01:33  01 JUL 1985

:T-FWD <cr> <----- Bypass ABS section
[94] END OF FILE

:ACCOUNT-RESTORE <cr> <----- Initiates account-restore

ACCOUNT NAME?STATUS <cr> <----- Prompts for desired account
STATUS 11948,17,1
  ERRORS 11965,1,1
    DL/ID 11966,3,1
M/DICT 11948,17,1 (S)
DL/ID 11948,17,1 (S)
TP 11970,1,1
  DL/ID 11971,7,1
PARCEL 12117,3,1
  DL/ID 12129,503,1

: <----- Returns to TCL when finished

```

Figure B: Account-Restore from Complete File-Save Tape

SYSTEM MAINTENANCE PROCEDURES: SYSPROG ACCOUNT

MULTIPLE ACCOUNT RESTORE

| The M-A-R PROC restores some or all of the accounts that have previously |
been saved by M-A-S PROC.

To use M-A-R from a tape created by M-A-S, load a tape in the usual way and then enter:

M-A-R<cr>

If more than one tape unit is present on the system, the utility prompts for the unit(s) to be used as follows:

Enter tape units desired, in sequence, separated by commas.
Maximum two half inch (1,2) or one quarter inch (3) ?

Reply with the unit number(s) of the tape units to be used.

The utility then displays some instructions followed by the tape label. The operator must reply with <cr> after the message:

Enter return to see the accounts on this tape -

This prompt exists to allow users, particularly those with single screen displays, sufficient time to read the initial instructions.

The utility now lists the accounts on the tape and invites the user to specify which are to be restored:

Account Name or * -

Individual accounts may be specified in any order, one at a time - this message will reappear after each name entered. An asterisk (*) reply will restore all the accounts on the tape. When all necessary choices have been input a <cr> reply will terminate this process and the utility will action the required restores.

Enter tape units desired, in sequence, separated by commas.
Maximum two half inch (1,2) or one quarter inch (3)? 1<cr>
[1021] DEVICE(S) TAPE 1 NOW ASSIGNED,
RECORD SIZE = 8192

Multi Account Restore

This utility will restore accounts from a tape created by the M-A-S
(Multi-Account-Save) utility.

When prompted enter the names of the accounts to be restored (in any
order) or just enter "*" to restore the entire tape contents.

Accounts that already exist will not be overwritten.

LABEL = }14:02:29 19 SEP 1984

Enter return to see the accounts on this tape -

Dummy

ATP

Account Name or * - *<cr>

:

Figure A: Sample M-A-R (Multi Account Restore) Dialogue

SELECTIVE-RESTORES

| The selective-restore capability allows individual files or items to be |
loaded onto the system from a formatted file-save tape.

SEL-RESTORE

This verb restores a file from a formatted file-save or account-save tape (Figure A). Before using SEL-RESTORE, the tape must be positioned correctly. Mount the tape at load point and select one of the following methods, depending on the type of tape being used:

- . For full file-save tapes:
 1. Attach the tape unit(s) as the default channel, specifying the proper record size (if known).
 2. If a bootstrap is present, enter:
T-FWD <cr>
 3. Now enter:
T-RDLBL 1 <cr>
 4. If an ABS section is present, enter:
T-FWD <cr>

- . For single account-save tapes:
 1. Attach the tape unit(s) as the default channel, specifying the proper record size (if known).
 2. Enter:
T-RDLBL 1 <cr>

Now that the tape is positioned at the beginning of the files section, enter:

SEL-RESTORE file-name item-list* {(options)}

where "file-name" is the file on the system in which items will be placed. This file must be defined on the account from which the restore is run. "Item-list" is the list of items to be restored. An asterisk (*) may be used to indicate all items on the tape file. You are then prompted for:

ACCOUNT NAME? a-name
FILE NAME? f-name

where "a-name" is the name of the account under which the file was saved on tape, and "f-name" is the name of the file as it appears on the tape. Entering <cr> to FILE NAME? causes the account Master Dictionary to be restored. "F-name" may be of the format 'file', 'DICT file' or 'file DL/ID' or 'file zzzz' where "zzzz" is a D-pointer in the dictionary of 'file'.

SYSTEM MAINTENANCE PROCEDURES: SYSPROG ACCOUNT

Applicable options are:

- O Overlay items already on the file if they have the same item-ids.
- A The tape is already positioned in the desired account.
- N File is to be identified on tape by its file number, in which case the prompt will be "FILE#?". The required file number is the one which accompanies the file on the file statistics printout for the appropriate file-save (under the column labelled "ORDER").
- C This option must be used with the 'N' option. It causes every item before the next end of file to be a candidate for restore. This ensures data can be restored even if a D-pointer on tape is damaged.

When using SEL-RESTORE, remember that account dictionaries follow all other files for the account. To restore Q-pointers for the SYSTEM Dictionary, use the 'N' option with File# = 1. This will be the last file on the tape.

```
-----  
:ASSIGN =TAPE 1 <----- Attach tape  
[1021] DEVICE(S) TAPE 1 NOW ASSIGNED,  
      RECORD SIZE = 8192  
  
:T-FWD <cr> <----- Bypass coldstart section  
[94] END OF FILE  
  
:T-RDLBL 1 <cr> <----- Read label to assure proper size  
LABEL = DOC.ACC]08:01:33 01 JUL 1985  
  
:T-FWD <cr> <----- Bypass ABS section  
[94] END OF FILE  
  
:SEL-RESTORE DICT TEMP * (0) <cr> <---- Restore all items into DICT TEMP  
  
ACCOUNT NAME?SYSPROG <cr> <----- File on tape is called DICT STAT-  
FILE NAME? DICT STAT-FILE <cr> <----- FILE under SYSPROG account  
SYSTEM 1  
BLOCK-CONVERT 2  
DL/ID (S)  
DL/ID (S)  
PROCLIB 5  
DL/ID (S)  
SYSPROG 7  
MD (S)  
M/DICT (S)  
DL/ID (S)  
STAT-FILE 11  
[992] 33 ITEM(S) HAVE BEEN RESTORED  
-----
```

Figure A: Example of SEL-RESTORE

TCL INITIATION OF FILE-RESTORE: :FILES AND :ABS/FILES VERBS

A complete formatted file-restore is part of a complete system ('AF') load. A file-restore may also be initiated from TCL by using the :FILES or :ABS/FILES verb.

COMPLETE FILE-RESTORE

To use :FILES (Figure A), mount the first reel of the file-save and perform these steps:

1. Attach the tape unit(s) as the default channel, specifying the proper record size (if known).
2. If a bootstrap is present, enter:
 T-FWD <cr>
3. Now enter:
 T-RDLBL 1 <cr>
4. If an ABS section is present, enter:
 T-FWD <cr>
5. Now that the tape is positioned at the files section, type:
 :FILES <cr>

Note that all files (including spooler files) on the current system are overwritten, the spooler is restarted, and all lines are sent to logon.

The verb :ABS/FILES functions similarly to the :FILES verb, but may be used to restore both the ABS and files portions of a system. To use this verb, put the first reel of the save tape on the drive and perform these steps:

1. Attach the tape unit(s) as the default channel, specifying the proper record size (if known).
2. If a bootstrap is present, enter:
 T-FWD (cr>
3. Now enter:
 T-RDLBL 1 <cr>
 :ABS/FILES <cr>

```

-----
:ASSIGN =TAPE 1 SIZE=2048 <----- Attach tape unit;
[1021] DEVICE(S) TAPE 1 NOW ASSIGNED,      a block length of
      RECORD SIZE = 2048                    2048 bytes for the
:T-FWD <cr> <----- ABS and FILES
[94] END OF FILE                            sections is
                                           anticipated
T-RDLBL 1 <cr> <-----
LABEL = DOC.ACCT]08:01:04  11 APR 1986      To bypass bootstrap
TAPE SIZE CHANGED TO  8192 <-----        -- section

:T-FWD <cr>                                ----- Read tape label
[94] END OF FILE                            -----
::FILES <cr>                                ----- Label indicates
                                           record size of 8192
                                           bytes

SPOOLER STARTED
SYSTEM 3136,11,1
  BLOCK-CONVERT 3147,7,1
  DL/ID 3147,7,1 (S)
  CHANNEL 3154,1,1
  DL/ID 3155,7,1
DL/ID 3136,11,1 (S)
  PROCLIB 3162,7,1
  DL/ID 3162,7,1 (S)
  POINTER-FILE 3169,5,1
  DL/ID 3169,5,1 (S)
  SYSPROG 3174,69,1
  MD 3174,69,1 (S)
  M/DICT 3174,69,1 (S)
  DL/ID 3174,69,1 (S)
  STAT-FILE 3243,1,1
  DL/ID 3244,87,1
  DISC-ERRORS 3331,3,1
  DL/ID 3334,1,1
  ERRMSG 3335,17,1
  DL/ID 3335,17,1 (S)
  ACC 3154,1,1 (S)

LOGON PLEASE:                               Note: All lines go to Logon
-----

```

Figure A: Example of :FILES Verb from TCL Using Formatted File-Save Tape

SYSTEM MAINTENANCE PROCEDURES: SYSPROG ACCOUNT

INITIALISING THE SYSTEM: SYSTEM-SETUP PROC

| After you complete a full restore, system upgrade, ABS load or coldstart, |
initialise the system with the SYSTEM-SETUP PROC.

SYSTEM-SETUP

Execute the SYSTEM-SETUP PROC after a restore or coldstart to initialise system locks, reset the time and date, and to perform other functions associated with setup.

Figure A presents a sample dialogue resulting from the use of this PROC (the actual dialogue depends on the options selected and on the configuration and facilities of the system).

```

:SYSTEM-SETUP <cr>

This proc is used to initialise certain SYSTEM PARAMETERS
depending on what type of activity has just occurred

    (AF) FULL RESTORE or SYSTEM UPGRADE  AF or UPG
    (A)  ABS RESTORE                      A
    (X)  COLDSTART                        X

Enter the option required (or C/R to exit)  X<cr>
Were the spooler tables reinitialised (Y/N)
Do you wish to enable users to LOGON (Y/N)  Y<cr>

Do you wish undefined ports to be automatically defined (Y/N)  N<cr>

PORT 000 ALREADY LOGGED ON
PORT 001 CHARACTERISTICS SET
etc.

Now clearing the Accounting file of active users
Now clearing the Phantom History File

Do you want to clear the System Log File (Y/N) NO<cr>

Now setting the Forced Write frequency

The current system time and date is 10:32:26 30 JAN 1986

Do you wish to change the time or date (Y/N)  N <cr>

The X option of SYSTEM-SETUP done .... you are now being logged off

LOGON PLEASE:

```

Figure A: Sample Dialogue Using SYSTEM-SETUP PROC

FORMATTED FILE-SAVES AND FILE-SAVE PROC

| File-saves should be performed on a daily basis to protect the data base. |
| Formatted file-save tapes are also a means of transporting software and |
data to other machines.

A formatted (logical) file-save dumps files in hierarchical order. Because each item is treated individually, file-save tapes may be used for selective restores. It also allows statistics to be gathered by the save. This approach allows users on the system while the save is running, though updates to files or file groups already saved will not be saved on the tape.

The general form of the input line is:

FILE-SAVE

The FILE-SAVE PROC first prompts for the tape record size. The record size can be any value between 512 and 32,256 bytes inclusive. A null input or a value less than 512 produces a default of 8192 bytes. A value greater than 32,256 produces a default of 8190 bytes.

The process prompts for the tape unit(s) to be used (if there is more than one on the system), then attaches the magnetic tape unit(s), rewinds the first tape, writes an EOF mark, and rewinds the tape. The next prompt is:

TAPE LABEL IF DESIRED=

Any string up to 16 characters may be included in the tape label. A tape label is always written on the tape preceding the ABS section. It contains the time and date, reel number, and the tape record block size.

The file-save may now proceed. A bootstrap section, an ABS section, and files section are dumped. Upon completion, a T-DUMP of file &STAT-FILE*n (where n is the port from which the PROC is being executed) is appended. This T-DUMP could later be loaded into the STAT-FILE and the PROC 'LIST-FILE-STATS' could be executed to show the contents of that tape. After the T-DUMP, the PROC 'LIST&FILE-STATS*' is entered to produce a file statistics report.

The terminal is then logged-off.

Figure A illustrates a file-save process.

```

:FILE-SAVE <cr>
11:56:06  19 SEP 1985

TAPE RECORD BLOCK LENGTH (IF OTHER THAN 8192) ? <cr>
Enter tape units desired, in sequence, separated by commas.
Maximum two half inch (1,2) or one quarter inch (3)? 1<cr>
[1021] DEVICE(S) TAPE 1 NOW ASSIGNED,
      RECORD SIZE = 8192

TAPE LABEL IF DESIRED = WEDNESDAY <cr>

. . . START FILE SAVE . . . 11:56:30  19 SEP 1985
DUMPING BOOTSTRAP
51 ITEMS SELECTED.

A1-2179

SYSTEM
ERRMSG
CHANNEL ... etc

[99] 288 TAPE RECORDS WRITTEN
. . . END FILE SAVE . . 12:01:17  19 SEP 1985
T-DUMP STAT-FILE
NOW PRODUCING FILE-STATISTICS REPORT

FILE SAVE COMPLETE ...

*****
**          CONNECT TIME AT 12:01:36 = 8 MINUTES          **
**          CPU MS = 306828          DISC I/O = 6726          **
**          NUMBER OF ACTIVATIONS = 203389          **
*****
*** LOGGED OFF AT 12:01:36    ON 19 SEP 1985          ***

LOGON PLEASE:

```

Figure A: Sample FILE-SAVE Dialogue

FILE-SAVE TAPES: FORMAT

| A formatted file-save tape has many types of data. The format of a
file-save tape is discussed below.

A formatted file-save tape created by the FILE-SAVE PROC contains the following:

The coldstart section consisting of several 512-byte frames of system modes in a binary-copy format

An end-of-file tape mark

A tape label

An ABS section consisting of a series of tape segments formatted as follows:

_A^ffff^...(512 bytes binary)...^cccc

where:

-' segment mark
ffff 4 digit hex ASCII frame number
^ attribute mark
cccc 4 digit hex ASCII checkcode = sum of the 512 bytes

A final section consisting of an 'X' followed by enough blanks to complete the last tape record

An end-of-file tape mark

A file section containing various segments each separated by a segment mark

An end-of-file tape mark

A T-DUMP of the appropriate statistics file

An end-of-file tape mark

Several types of segment are in the files section (Figure A). The first byte of each is a segment mark. This was designed so that restores would not be stopped if some error destroyed tape segment(s). Segments are referred to by the second character or "segment type". Currently, there are D, Q, B, I, O and X segments. The following topic describes these segments in more detail.

<u>ABS SECTION SEGMENTS</u>	
TYPE A	<-----Represents one frame of object code.
TYPE X	<-----Indicates end of ABS section.

<u>FILES SECTION SEGMENTS</u>	
TYPE D	<-----Represents a new file.
TYPE Q	<-----A D-pointer with a base equal that of previously saved file.
TYPE B	<-----A DATA/BASIC catalogued program.
TYPE I	<-----An item.
TYPE O	<-----Save list or DATA/BASIC object code.
TYPE X	<-----End of Files Section.

Figure A: Formatted File-Save Segment Types

TAPE SEGMENTS IN FORMATTED FILE-SAVE

Files section segment types D, Q, I and B are described below.

D-segments identify a new file. Their format is:

(SM)D(AM)level(AM)filecount(AM)size...pointer...(AM)

where "level" is 0, 1, 2, or 3, and equals one plus the level of the dictionary in which the D-pointer is placed. There is only one level zero D-segment, which identifies the System Dictionary. This segment is derived from the item 'SYSTEM' in the System Dictionary. A level one D-segment defines a pointer to an account Master Dictionary, a level two D-segment defines a file dictionary, and a level three D-segment defines a pointer to a data file. "Filecount" is a sequential file number in hex assigned to D-segments. The "size" field and the rest of the pointer item is saved as it is in the dictionary.

A Q-segment is generated whenever a D-pointer is found with a base which is already in the table of dumped files. Its format is:

(SM)Q(AM)level(AM)size....pointer...(AM)

where "level" is as defined above.

I-segments are generated one per item dumped. Their format is:

(SM)Ifilecount(AM)group(AM)size....item...(AM)

The "filecount" (HEX) is identical to the filecount of the D-pointer which points to the file in which the item is found; "group" is the hex FID of the first frame of the group from which the item was dumped. This is saved to eliminate hashing, and to ensure that the scheme of the "fast method" of restoring files can proceed by sequential groups.

Items in the POINTER-FILE are saved as B-segments of the following form:

(AM)size...item...(AM)

In addition, each frame (the first of which is pointed to by a POINTER-FILE item containing the catalogued object or SAVE-LIST) is saved with a segment of the form:

(SM)OB(AM)filecount(AM)1000 bytes of frame in ASCII hexadecimal format (AM).

Each binary byte in the catalogued frame is converted to two ASCII hexadecimal bytes in the tape segment. Typical formatted tape segments are shown in Figure A.

RECORD = 5 ; 512 BYTES.

```

000: ME^A^6^.....R^5^5^_16^5519^001C5^S^5^BEGIN^5^.....
050: R^8^1^_16^5519^00192^S^2^RS^2^.....L^2^1^_16^5519^0
100: 01BRESULT^A^2^.....L^2^1^_16^5519^001B8^S^8^VERS^
150: 8^.....L^2^1^_Q^2^0022DL/ID^D^18546^47^1^.....L^20^1
200: ^_D^2^9^0026TEMP^D^20626^1^1^.....L^10^... (1,1)^_D^3
250: ^A^0027DL/ID^D^20627^1^1^.....L^10^... (1,1)^_D^2^B^0
300: 029NEWAC^D^21787^11^1^.....L^10^... (11,1)^_Q^3^0020D
350: L/ID^D^21787^11^1^.....L^10^_IB^551B^0014SP-EJECT^P
400: ^10A8^_IB^551B^0015BLOCK-PRINT^P^41^_IB^551B^0011M
450: ULD^PM^40A0^_IB^551B^001E^A9998^A^9998^.....R^4^1
500: ^_IB^551B^00
    
```

RECORD = 6 ; 512 BYTES.

```

000: 1D7^S^7^V/CONV^.....L^15^1^_IB^551B^000BAND^C1^_IB
050: ^551B^000FHEADING^CL^_IB^551B^0018T-WEOF^PZ^6033^B
100: 023^_IB^551B^001BV/MIN^A^11^.....L^3^1^_IB^551B^0
150: 017FORM-LIST^P^2^303A^_IB^551B^001B^A10^A^10^.....
200: ^L^10^1^_IB^551B^0019^A8^A^8^.....L^10^1^_IB^551B
250: ^0010USER^P^40BB^_IB^551B^001F11^S^11^V/MIN^11^....
300: ^L^3^1^_IB^551B^0061T-LOAD^PT^35^207C^_IB^551B^000
350: CDICT^CY^_IB^551B^002FBLOCK-CONVERT^Q^BLOCK-CONVER
400: T^.....L^10^1^_IB^551B^0015SP-ASSIGN^P^10A7^_1B^5
450: 51B^0022LP132^PQ^(DICT PROCLIB.LP132)^_IB^551B^001
500: 7DATE-FORMAT
    
```

(Segment marks (X'FF') print as '_' here).

Item SP-EJECT(size: 20) in group X'551B' _____
in the 11th file.

Q-segment (level 3: data file) defines DL/ID _____
which has same base, modulo and separation
as file NEWAC.

D-segment (level 2: dictionary file) defines _____
NEWAC file as the 11th file.

Block of I-segments, each defining an item in _____
the NEWAC file.

Figure A: Typical Formatted Tape Segments

OPERATION OF FORMATTED FILE-SAVE

This topic describes the operation of the formatted file-save process.

The file-save dumps all pointers and all subfiles at each level before dumping the data for that level. This is done so that D-pointers can be placed at the front of their group when the file is restored; e.g., for a simple system, the order might be:

SYSTEM POINTER

ACCOUNT1 POINTER

FILE-DICTIONARY1 POINTER

* DATA1 POINTER

* DATA1 ITEMS

* DICTIONARY1 ITEMS

FILE-DICTIONARY2 POINTER

DATA2 POINTER

* DATA2 ITEMS

* DICTIONARY2 ITEMS

* ACCOUNT1 ITEMS

ACCOUNT2 POINTER

* .

* .

* .

SYSTEM DICTIONARY ITEMS

* Blocks of I-segments

SYSTEM MAINTENANCE PROCEDURES: SYSPROG ACCOUNT

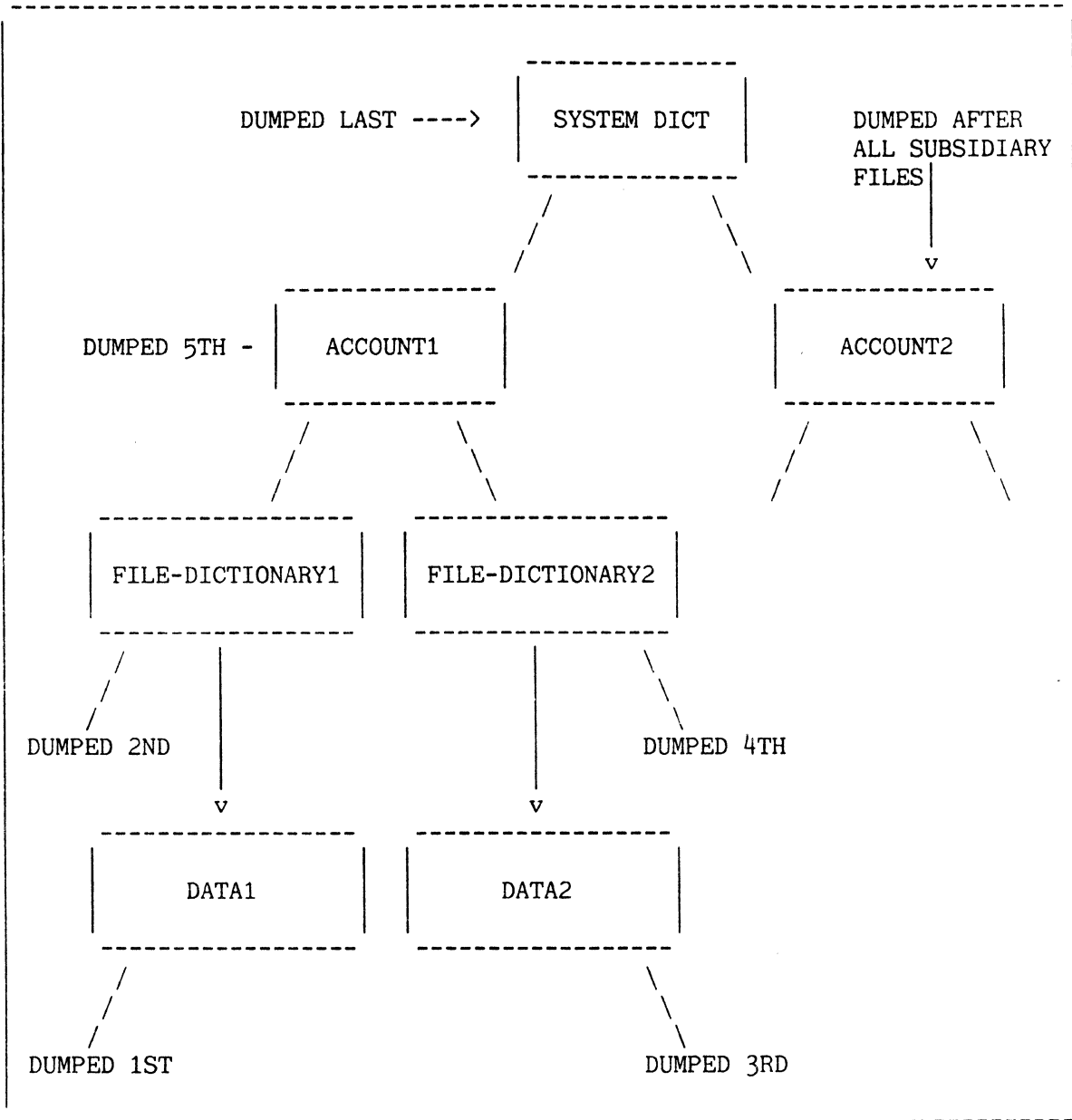


Figure A: Typical Layout of Formatted File-Save Process

SYSTEM MAINTENANCE PROCEDURES: SYSPROG ACCOUNT

PLANNED SHUTDOWN PROC

```

-----
The SHUTDOWN PROC allows you to perform a secure and complete halt of the
system (i.e., no data will be altered in a manner that cannot be
accurately recovered when power is restored). You must have a SYS2
privilege level to execute this PROC.
-----

```

You should use this PROC before you boot the system, change the system hardware, or leave the office for the evening or weekend.

Note: This PROC may only be executed from port 0.

To execute this PROC, enter the following command:

SHUTDOWN <cr>

The following message is then displayed:

```

ALL PORTS ARE NOW PREVENTED FROM LOGGING ON !!!
IF THIS PROC IS BROKEN ENTER 'ENABLE-LOGONS *' TO RE-ENABLE ALL PORTS

```

DO YOU WANT TO MESSAGE ALL USERS OF THE PENDING SHUTDOWN?

Enter 'N' if you do not wish to send the message; entering 'Y' causes this question to be displayed:

'HOW MANY MINUTES WARNING SHOULD BE GIVEN?'

Here you may enter a number from 1 to 5. Once a number is entered, the message "THE SYSTEM WILL SHUTDOWN IN x MINUTES. PLEASE LOGOFF" is sent each minute to each logged-on user, with 'x' showing the number of minutes remaining. When the specified time has elapsed, a System Status Report is displayed on the port 0 terminal.

The initial task of the Shutdown program is to evaluate the state of the system. By scanning through all configured PCBs, the system finds those users still logged on. Then, for each user logged on, their PIB and PCB is checked for type of process (i.e., TIPH, spooler, or normal) and its status. This data is then used to generate the following System Status Report covering all active processes:

REPORT OF ALL LOGGED ON USERS, EXCLUDING THIS PROCESS:

PORT	PIB STATUS	ACTIVITY	ACCOUNT NAME	LOCATION
##	status	NORMAL TIPH or SPOOLER	name	location

SYSTEM MAINTENANCE PROCEDURES: SYSPROG ACCOUNT

Following this report are the following messages:

```
WARNING!  THE SHUTDOWN SHOULD NOT PROCEED IF USERS ARE LOGGED ON,  
OF IF THE SPOOLER IS ACTIVE!
```

```
DO YOU WANT TO PROCEED WITH THE SHUTDOWN?  (ENTER 'Y' OR 'N').
```

If users are still logged on, or the spooler is active, enter 'N' to terminate the program and return to TCL. Enter 'Y' to begin the Shutdown. As soon as the Shutdown is completed, these messages are displayed:

```
SYSTEM SHUTDOWN  
TYPE "R" TO RESUME  
>
```

At this point, the system is in a secure state, and power to the system may be turned off. To an active terminal, the system now appears to have stopped; no one can log on to the system.

RESTARTING THE SYSTEM

There are two methods you can use to restart the system: 1) enter 'R' in response to the terminal prompt, 2) perform a coldstart.

You may enter 'R' in normal circumstances. This should reactivate each process, appearing as if a Shutdown had never occurred.

You must perform a coldstart if memory boards were changed or an error has occurred after the Shutdown.

COLDDUMP AND SAVE VERBS

The COLDDUMP verb creates a coldstart section on tape from items in SYSTEM-OBJECT. The SAVE verb allows you to dump ABS frames and files. These verbs use the currently assigned default channel.

The COLDDUMP verb creates a series of binary data 512-byte tape records necessary to perform a bootstrap on any system. An item called COLD-LIST in the SYSTEM-OBJECT file contains the list of item-ids necessary for the coldstart. The FORM-LIST verb is used to select these items (see Figure A). Modes included in the monitor and boot loader are not located in the ABS area. The monitor is always resident in main memory. The general form of the COLDDUMP command is:

COLDDUMP file-name {item-list} {(E)}

The 'E' option causes error messages to be printed.

The SAVE verb initiates dumping of the ABS and/or the files sections. It also generates file statistics. The general form of the SAVE verb is:

SAVE SYSTEM * {(options)}

where "options" are:

- A Dumps ABS frames. ABS limits will be prompted for. Enter a <cr> to dump all ABS as specified by the hardware, otherwise enter decimal numbers or ranges separated by commas. For example; 1-10,15,20-500.
- D Dumps data files.
- F Lists the name of every file as it is encountered. Otherwise, only account names are listed.
- I Saves individual account.
- K Kills output to tape. This option is used to generate stats only.
- S Creates file statistics in STAT-FILE.
- T Inhibits writing tape label.
- Z Creates file statistics in user-specified file.

Note: 'S', 'Z', 'I', and 'F' are meaningful only with 'D' option.

Any prompts (a prompt is indicated by "?") are made in this order:

1. An optional description, of up to 16 characters, to be included in the tape label. (Not produced with 'T' option.)
2. ABS limits. (Produced with 'A' option only)

3. The name of the file to hold file statistics. (Produced with 'Z' option only).

If the 'I' option was specified, the account name is then prompted for:

ACCOUNT NAME =

GROUP LOCKS

The file-save locks each group before accessing data in the group, to prevent data loss due to "soft" group format errors. Up to four groups may be locked at one time; one in the SYSTEM Dictionary, one in the account Master Dictionary, one in a file dictionary, and one in a data file.

TREATMENT OF GROUP FORMAT ERRORS

If the file-save encounters a group format error, the secondary space is copied into new frames in overflow, the frames previously in the secondary space are disconnected, and the new frames are connected to the group. This is done to alleviate conflict between files that are using common frames for their secondary space.

NON-MAVED ITEMS AND SKELETON FILES

An item in a dictionary whose D/CODE is 'DX' will not be saved. A file pointer whose D/CODE is 'DY' will be saved, but items in the file will not be saved, such as the data portion of the STAT-FILE, which is not valid after a restore.

USE OF OVERFLOW

If the 'S' or 'Z' option is not used, the file-save can be run when there is no available disc space on the system. No additional space is needed unless the total number of files on the system exceeds 1500. One frame of overflow is needed for every 125 files on the system in excess of 1500.

```

-----
:ASSIGN =TAPE 2
[1021] DEVICE(S) TAPE 2 NOW ASSIGNED,
      RECORD SIZE = 8192
:FORM-LIST SYSTEM-OBJECT COLD-LIST <cr>
xx ITEMS SELECTED.                                Gets item-ids for items
                                                    necessary for coldstart.
>COLDDUMP SYSTEM-OBJECT (E) <cr>                 Creates coldstart tape
                                                    suppressing messages except
                                                    errors.
-----

```

Figure A: Creating a Coldstart Tape

BOOTDUMP VERB

| The BOOTDUMP verb produces the filesave tape bootstrap section from the |
system disk.

This verb is used by the FILE-SAVE PROC and other utilities. Its general format is:

BOOTDUMP {(L)}

If the L option is used, you are prompted for a tape label, which may be any string of up to 16 characters.

A tape unit must be assigned before use of this verb: the default channel is used.

The tape created consists of the following sections:

- . tape and disk secondary boot
- . writable control store (WCS)
- . Monitor, Configurator and Terminal Control Program (TCP)
- . ABS frames

THIS PAGE INTENTIONALLY LEFT BLANK

SYSTEM MAINTENANCE PROCEDURES: SYSPROG ACCOUNT

CREATING AND ENABLING USER-ACCOUNTS

The CREATE-ACCOUNT PROC creates new user-accounts. The SETUP-ASSY PROC enables the user-account to assemble assembly language programs. All accounts are set up to run DATA/BASIC. Both of these PROCs are present in the SYSPROG account.

CREATE-ACCOUNT PROC

The CREATE-ACCOUNT PROC performs the following:

- . Creates a new file (to be the user's M/DICT) with the new user-name as the file definition item-id; the file definition item is placed in the SYSTEM Dictionary.
- . Copies the contents of the NEWAC file (the prototype M/DICT) to the newly created user M/DICT.

The CREATE-ACCOUNT PROC is called by typing in the PROC name:

```
CREATE-ACCOUNT
```

This PROC then prompts you for the required information which is summarized in Figure A. A null input (i.e., carriage return only) for any of the prompts (except ACCOUNT NAME, PRIVILEGES, and MODULO, SEPARATION) generates a null value for that attribute in the user identification item (see Section 5). A null input is not allowed for ACCOUNT NAME. A null input for "PRIVILEGES" generates a SYSO level and zero additional work space. The modulo and separation default values are 29 and 1, respectively.

After all prompts have been answered, the system further prompts:

```
ENTER NUMBER TO CHANGE, * TO CANCEL, RETURN TO CONTINUE:
```

A carriage return in reply to the above prompt causes the PROC to create the new account. A number causes the corresponding entry-screen prompt to be reopened. An asterisk cancels the PROC. Once a carriage return is entered, the PROC should not be interrupted with the BREAK function key.

The CREATE-ACCOUNT PROC should not be used to create a new synonym to an existing account; this should be done by using the EDITOR to create the file synonym definition item in the System Dictionary.

SETUP-ASSY PROC

The SETUP-ASSY PROC sets up a user-account so that it is able to assemble assembly language programs. To evoke this PROC, enter "SETUP-ASSY" followed by the account name. For example:

```
SETUP-ASSY USER3 <cr>
```

SYSTEM MAINTENANCE PROCEDURES: SYSPROG ACCOUNT

Once activated, the PROC asks a series of questions requiring input from you. These questions are self-explanatory. However, it should be noted that a response of "Y" to the prompt:

ENTER 'Y' TO CREATE CSYM AND XSYM FILES FOR THE ACCOUNT (Y/N)?

causes file %XSYM to be temporarily created on the SYSPROG account.

<u>PROMPTS</u>	<u>EXPLANATION</u>
ACCOUNT NAME <-----	The new account name may be up to 50 characters long and contain any characters except commas.
L/RET-CODE(S) <-----	This is the set of retrieval lock codes to be associated with the user; multiple values can be entered separated by value marks (<c>], X'FD'). (The retrieval lock codes are described in the section LOGON/LOGOFF).
L/UPD-CODE(S) <-----	This is the set of update lock codes associated with the user (same as retrieval lock codes above).
PASSWORD <-----	This is the password associated with the user's LOGON item in the System Dictionary.
PRIVILEGES ADDITIONAL WORK <---- SPACE	This is the code describing the user's privileges and work space assignment (described in the section LOGON/LOGOFF).
ACCOUNT FILE UPDATE Y/N <-----	If 'Y' is entered, a 'U' is placed in the 9th attribute and logon history data will be accumulated in the ACCOUNT file.
MODULO,SEPARATION <-----	This is modulo and separation for the M/DICT. Default value for modulo is 29; default value for separation is 1.

Figure A: Information Required by CREATE-ACCOUNT PROC

SYSTEM MAINTENANCE PROCEDURES: SYSPROG ACCOUNT

DELETING USER-ACCOUNTS: DELETE-ACCOUNT PROC

The DELETE-ACCOUNT PROC deletes all files belonging to a specified account, thus removing the account from the system.

DELETE-ACCOUNT is a PROC in SYSPROG-PL which prompts for the name of an account to delete. The dialogue is shown in Figure A.

After the account name is verified, the PROC displays all duplicate D/CODE pointers and Q-pointers to D/CODE pointers that point to the account. You are then given the option of cancelling the deletion or continuing with it. If you continue, all duplicate D/CODE and Q-pointers are deleted from the system.

DELETE-ACCOUNT also checks for users logged on to the master, duplicate D/CODE, and Q-pointer accounts. If any user is logged on, the delete is cancelled.

This PROC sets the group lock, as does the FILE-SAVE PROC, to prevent the account from being deleted during a FILE-SAVE operation.

Requirements to run DELETE-ACCOUNT are:

1. All other users should be logged off before running this PROC because an item in the System Dictionary will be deleted.
2. You must be logged on to SYSPROG.
3. DELETE-ACCOUNT uses the item-ids '%DAaccount-name%' and '%DAaccount-name%%' in SYSPROG's M/DICT; if the process is interrupted and terminated before completion for any reason, DELETE THESE ITEMS FROM SYSPROG's M/DICT!

```
:DELETE-ACCOUNT <cr>

      DELETE-ACCOUNT  !!! IF A FILE-SAVE IS SAVING THE ACCOUNT TO !!!
                      !!! BE DELETED, THEN DELETE-ACCOUNT WILL    !!!
                      !!! WAIT UNTIL THE ACCOUNT IS SAVED . . . . !!!

ENTER ACCOUNT NAME? DUMMY <cr>
IS 'DUMMY' THE RIGHT ACCOUNT ? Y <cr>
CHECKING FOR GROUP LOCKS ON FILE-SAVE

NOW PREPARING LIST OF FILES TO DELETE

NOW DELETING ACCOUNT LEVEL FILES

**** DUMMY'S DL/ID DELETED
**** DUMMY'S M/DICT DELETED
**** DUMMY'S MD DELETED
**** DUMMY'S PROC DELETED

**** DELETE ACCOUNT 'DUMMY' COMPLETED
```

Figure A: Sample DELETE-ACCOUNT Dialogue

SYSTEM GENERATION AND VERIFICATION

| This topic discusses SYSPROG PROCs that relate to the system generation |
| process. For further information regarding these PROC's, refer to the |
installation procedures supplied with the applicable software release.

SYS-UPDATE PROC

The SYS-UPDATE PROC updates system files after a new release of software is loaded. Because its use and operation are peculiar to each release of the system, detailed instructions are not given here.

UPDATE-ACCOUNT PROC

The UPDATE-ACCOUNT PROC updates accounts after a SYS-UPDATE to reflect changes to verbs, PROCs, and other elements of a M/DICT. It should never be used to update the SYSPROG account, which is always done separately as part of the system update procedure. Because the use and operation of this PROC is peculiar to each release of the system, detailed instructions are not given here.

VERIFY-SYSTEM PROC

The VERIFY-SYSTEM PROC verifies the object code in the SYSTEM-OBJECT file against the corresponding code loaded in the executable ABS region and lists modes that have mismatches.

If no options are specified, the MVERIFY 'E' option (list only errors) is assumed. Other options for the MVERIFY verb may be specified, such as, 'P' (output to printer) and 'A' (list all mismatch bytes).

This PROC should be executed whenever there is a suspicion that the system software is not operating correctly (e.g., abnormal aborts, etc.). If mismatches occur, the code loaded in the ABS area is bad and corresponding item from the SYSTEM-OBJECT file should be MLOADed. If a checksum error occurs, then most likely the item in the SYSTEM-OBJECT file has changed (or was patched incorrectly). In this case, it is best to T-LOAD the item from a SYS-GEN tape, apply applicable patches, and then MLOAD the item.

SYSTEM MAINTENANCE PROCEDURES: SYSPROG ACCOUNT

(THIS PAGE INTENTIONALLY LEFT BLANK)

VERIFYING, AND MAKING CONTIGUOUS, AVAILABLE SYSTEM SPACE

CLINK verifies the integrity of the file space chain and truncates the chain if necessary. TCLINK does the same for the temporary space chain, and optionally moves frames from the temporary space chain to the file space chain. CLAIM returns linked available space to contiguous available space. These verbs are normally only present on the SYSPROG account.

The CLINK and TCLINK verbs are provided to solve problems due to corrupted available space chains. Typical symptoms are: spooler output which appears to include intermixed or garbled data, or space chains which appear to be unusually large compared with their normal sizes (as reported using the POVVF verb: see Section 9). In the latter case, it is best to run CLINK and/or TCLINK before using CLAIM to verify that space really is fragmented rather than just appearing to be. It is recommended that these verbs are executed using a TIPH process, as they take some time to execute and do not produce output until verification is complete. All these verbs can be run while other processes are running, but may cause some degradation in performance.

The general form of CLINK is:

CLINK {(P)}

Use of the 'P' option directs all output to the system printer (or other currently assigned spooler device).

On entering CLINK the POVVF report is displayed. The file space chain is temporarily detached from the system and the chain is checked frame by frame, starting from the head of the chain. A count is maintained of frames checked. Each frame is checked for a backward link (frames in the chain should only be linked by forward links) and marked as having been checked, unless such a marker is already set; if so, the frame is linked into the chain more than once, (there is a 'loop' in the chain) which is also an error.

If the last frame in the chain (a frame with no forward link) is reached without errors, the count of frames checked is compared with the count of frames held by the system. If these correspond, the markers set in the chain are deleted, the chain is re-attached to the system and a second POVVF report is displayed (so that any activity occurring while the chain is being checked can be inspected).

If a discrepancy in the count, a backward link or a loop is discovered, one of the following messages is displayed:

DIFFERENCE IN THE NUMBER OF FRAMES
BACKWARD LINK IN CHAIN or
LOOP IN CHAIN

The frame containing the backward link, or which was found twice, is detached from the chain together with any linked frames, in the latter two cases, and the decimal FIDs of the last 'good' frame and of the 'bad' frame are reported on separate lines after the error message. The system count is then updated. The chain is cleared of markers, re-attached for system use and the POVf report is re-displayed.

The general form of the TCLINK verb is:

```
TCLINK {(option)}
```

Where option may be 'P' (output to spooler) and/or 'X' (link temporary space chain to file space chain, after verification). This verb verifies the temporary space chain in exactly the same way that CLINK verifies the file space chain. The 'X' option is provided so that if the temporary space chain becomes too large it can be joined onto file space, which is more commonly used.

The CLAIM verb (which has no options) converts linked space to contiguous space. It sorts the FIDs of linked frames and forms contiguous frames into blocks, which are placed in the contiguous space tables. Individual frames are added to contiguous blocks if possible. Any remaining frames are left in their respective chains in linked form. The POVf report is then displayed.

Figure A illustrates the use of these verbs.

```

-----
:CLINK <cr>

[POVF display]                (as illustrated in Section 9)

BACKWARD LINK IN CHAIN
654390
047791

[POVF DISPLAY]

:TCLINK (PX)

[POVF display]                ---
DIFFERENCE IN THE NUMBER OF FRAMES      |
[POVF DISPLAY]                |       Output directed to spooler
: CLAIM <cr>                    |       and temporary space chain
[POVF DISPLAY]                |       re-attached to file space
                               |       chain
                               ---
-----

```

Figure A: Examples of Use of CLINK, TCLINK and CLAIM Verbs

DISK AND PLAN ERROR RECORDING

The system records errors for use by customer engineers.

When enabled (once the spooler is active), the system records disk errors in the SYSTEM-LOG file. All types of disk errors except DMA channel overruns are recorded.

When an uncorrectable disk error occurs, an ampersand (&) is printed on the terminal that caused the disk operation. The rest of the system will not normally fail on one disk error.

The data recorded includes: date and time, process number, controller number, channel number, command (read or write), cylinder, head and sector numbers, transfer length requested and achieved, and controller status.

Disc error recording is started automatically after a restore. The SYSTEM-LOG file remains in its current state after an 'X' or 'A' restore. However, after an 'AF' restore, the table contains the errors recorded from the tape used to execute the restore. Therefore, the SYSTEM-SETUP PROC option to initialise the SYSTEM-LOG file should be selected after an 'AF' restore.

The PROC LIST-DISC-ERRORS produces a report of the disk errors contained in the SYSTEM-LOG file. The (P) or LPTR option directs the report to the printer. Figure A shows an example of a disk error report.

The Plan controller records errors detected on the network, and these errors are logged once every minute in the SYSTEM-LOG file. They may be reported using the LIST-PLAN-ERRORS PROC, which produces a report similar to that shown in Figure B.

This report shows: the date and time at which errors were detected, the process and controller numbers and related number of errors (CRCC - Cyclic Redundancy Character Check - column), and the four controller status bytes (each byte represented by two hex digits).

Run after any type of coldstart, the SYSTEM-SETUP PROC starts the logging of Plan errors and also prompts for whether or not to clear errors previously recorded in the SYSTEM-LOG file.

```

:LIST-DISC-ERRORS (P) <cr>

DISC ERRORS
DATE .....TIME.  PRCS CONT CHAN COMM CYL HEAD SEC LENG LENG. STATUS
                                TRNS

28 NOV 1984 18:00  38  08  04 READ  25   1 13  512  512  40 CO 11 9A
28 NOV 1984 18:00   7  08  04 READ  32   1 13  512  512  40 CO 12 9A
28 NOV 1984 18:00   7  08  04 READ  32   1 13  512  512  40 CO 12 9A
28 NOV 1984 18:00  16  08  04 READ  36   1 13  512  512  40 CO 32 9A
28 NOV 1984 18:00  16  08  04 READ  36   1 13  512  512  40 CO 32 9A
28 NOV 1984 18:00  16  08  04 READ  36   1 13  512  512  40 CO 32 9A
    
```

Figure A: LIST-DISC-ERRORS Command

```

:LIST-PLAN-ERRORS <cr>

P-LAN ERRORS
DATE..... TIME. PRCS CONT CRCC STATUS.....
15 APR 1986 14:15  12  04  1 00 00 C8 OE
    
```

Figure B: LIST-PLAN-ERRORS Command

SYSTEM MAINTENANCE PROCEDURES: SYSPROG ACCOUNT

SYSTEM ERROR RECORDING

The system records system errors for use by customer engineers.

System errors are recorded, along with disk and Plan errors, in the SYSTEM-LOG file.

System errors include overheating/power fail errors, bus errors, memory parity errors, illegal service request errors and lost service request errors. The errors are accumulated after a complete restore and are not cleared until the next complete restore. A sample report is shown in Figure A. An explanation of the table follows:

DATE	-	date on which error occurred
TIME	-	time at which error occurred
MEM ERRS	-	parity error count with row of totals (per memory board)
BUS ERRS	-	bus error count with total
ISR ERRS	-	illegal service request count with total
LSR ERRS	-	lost service request count with total
POWER FAIL	-	PWF (powerfail)/AIRF (overheat)
ISR/LSR CONT	-	controller number
ISR/LSR CHAN	-	channel number

The PROC LIST-SYS-ERRORS lists all system errors. The (P) or LPTR option directs this report to the printer.

A typical system error report is shown in Figure A.

The PROC LIST-OVF-ERRORS produces a report showing any occasions on which the error buffer was overflowed, causing errors not to be logged in the SYSTEM-LOG file.

Figure B shows an example of the report produced by the LIST-OVF-ERRORS PROC.

```

-----
:LIST-SYS-ERRORS (P) <cr>

SYSTEM-ERRORS
DATE..... TIME. MEMO MEM1 MEM2 MEM3 BUS. ISR. LSR POWER ISR/LSR ISR/LSR
ERRS ERRS ERRS ERRS ERRS ERRS ERRS FAIL CONT CHAN

30 NOV 1984 11:40 0 0 0 1 0 0 0 AIRF 00 00
30 NOV 1984 11:42 0 0 0 0 1 0 0 AIRF 00 00
30 NOV 1984 11:43 0 0 0 0 1 0 0 PWF 00 00
30 NOV 1984 11:44 0 0 0 1 0 0 0 NONE 00 00
30 NOV 1984 11:45 0 0 0 0 1 0 0 NONE 00 00
0 0 0 2 3 0 0
-----

```

Figure A: LIST-SYS-ERRORS Command

```

-----
:LIST-OVF-ERRORS (P) <cr>

OVERFLOW ERRORS
ID. DATE..... TIME.

OVF 30 NOV 1984 11:41
OVF 30 NOV 1984 11:44
OVF 30 NOV 1984 11:49
OVF 30 NOV 1984 12:18
OVF 30 NOV 1984 13:01
-----

```

Figure B: LIST-OVF-ERRORS Command

SYSTEM MAINTENANCE PROCEDURES: SYSPROG ACCOUNT

DISK MAPPING: CALCFID AND DSKAD VERBS

| The CALCFID verb calculates a decimal frame identity (FID) from its disk |
| address. The DSKAD verb calculates a disk address for a given decimal |
FID.

The general form of the CALCFID verb is:

CALCFID

You will be prompted for the required information, as shown in Figure A.

The general form of the DSKAD verb is:

DSKAD frame-number

Figure A shows examples of the CALCFID and DSKAD verbs.

```
:CALCFID<cr>
Drive No      :0<cr>
Controller No :8<cr>
Channel No    :4<cr>
Cylinder No   :3<cr>
Head No       :0<cr>
Sector No     :10<cr>

FID is        :442

:DSKAD 442<cr>

DISK ADDRESS FOR FRAME NO. :442
DRIVE          : 00
CONTROLLER     : 08
CHANNEL        : 04
CYLINDER       : 3
HEAD NO.       : 0
SECTOR        : 10
:
```

Figure A: Sample Usage of CALCFID and DSKAD Verbs

FILE STATISTICS REPORT: LIST-, ALL-, AND ACCOUNT-FILE-STATS

| The File Statistics Report is a valuable tool for data base management. |
| This report is automatically created by the PROCS LIST-FILE-STATS, |
LIST-£STAT-FILE*, ALL-FILE-STATS, and ACCOUNT-FILE-STATS.

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.

Note: the statistics report is always printed.

The report is broken down by account. The following information is included for each file in the account:

- Base, modulo and separation parameters
- Order of the file in the file save
- Total number of items in file
- Size of the file in bytes
- Maximum and minimum number of items in a group
- Total frames (including linked) used by file
- Breakdown of groups by percent full
- Group Format Errors (GFEs) in file (if any)

A total line is generated for each account, which lists the total:

- Items
- Bytes (characters)
- Frames (includes linked)
- Group format errors

Figure A shows a portion of a File Statistics Report.

The previous method for creating a File Statistics Report, using STAT-FILE, has been maintained for compatibility with user-written PROCS. In this method, the SAVE verb is used with the 'S' option to generate the stats. The LIST-FILE-STATS PROC can then be used to output the File Statistics Report.

The new method for creating a File Statistics Report is used by all other system PROCS (including FILE-SAVE and ACCOUNT-SAVE). These PROCS use the SAVE verb with the 'Z' option to generate the stats into a file called £STAT-FILE*'port no.' (where 'port no.' is the port from which the PROC is run). The stats are then output using the PROC LIST-£STAT-FILE*.

ALL-FILE-STATS uses the 'Z' option with the SAVE verb and clears, generates into and outputs from a data file called £STAT-FILE*'port no.' using PROC LIST-£STAT-FILE*. The resulting File Statistics report covers all accounts on the system.

SYSTEM MAINTENANCE PROCEDURES: SYSPROG ACCOUNT

The ACCOUNT-FILE-STATS PROC is similar to ALL-FILE-STATS, except that it prompts for one account name and generates statistics for that account only.

Note: The appendix shows the dictionary which is used for STAT-FILE and &STAT-FILE*n files : PROCs may be written to report the defined attributes as required.

FILE	BASE	MOD	SEP	ORDER	ITEMS	SIZE	MX/OP	NN/OP	FRAMES	25%	50%	75%	100%	125%	150%	200%	>200%	CPE
XBSOLETEX	33167	1	1	1	20	1	34	1	1	1	0	0	0	0	0	0	0	0
XBSOLETEX*DL/ID	33168	1	1	1	21	2	1,499	2	2	3	0	0	0	0	0	0	0	0
APATCHFILE	33732	1	1	1	31	1	32	1	1	1	0	0	0	0	0	0	0	0
APATCHFILE*DL/ID	33733	1	1	1	32	2	1,001	1	0	12	10	0	0	0	0	0	0	0
BASIC-COMPILERS	32898	1	1	1	11	1	35	1	1	1	1	0	0	0	0	0	0	0
BASIC-COMPILERS*DL/ID	32899	1	1	1	12	5	79,236	1	0	167	6	0	0	0	0	0	0	0
D																		
BP	33248	1	1	1	29	1	35	1	1	1	1	0	0	0	0	0	0	0
BP*DL/ID	33249	43	1	1	30	34	224,959	4	0	483	20	1	0	2	0	1	0	19
CSVM	33103	1	1	1	18	1	35	1	1	1	1	0	0	0	0	0	0	0
CSVM*DL/ID	33104	63	1	1	19	0	0	0	0	63	63	0	0	0	0	0	0	0
DL/ID	32797				35													
M/DICT	32797				17													
MD	32797				26													
MEMAC	33172	1	1	1	23	1	35	1	1	1	0	0	0	0	0	0	0	0
MEMAC*DL/ID	33173	19	1	1	23	373	8,850	27	14	27	0	0	4	9	5	1	0	0
PATCHFILE	33090	1	1	1	15	1	32	1	1	1	1	0	0	0	0	0	0	0
PATCHFILE*DL/ID	33091	11	1	1	15	2	999	1	0	12	10	0	0	0	0	0	0	0
PHISTORY	33236	1	1	1	27	1	32	1	1	11	11	0	0	0	0	0	0	0
PHISTORY*DL/ID	33237	11	1	1	28	0	0	0	0	11	11	0	0	0	0	0	0	0
QA	40661	1	1	1	36	1	34	1	1	3	2	0	0	0	0	0	0	0
QA*DL/ID	40662	3	1	1	37	2	272	2	0	3	2	0	0	0	0	0	0	0
RELEASE-LETTER	33198	1	1	1	24	1	35	1	1	1	1	0	0	0	0	0	0	0
RELEASE-LETTER*DL/ID	33199	37	1	1	25	0	0	0	0	37	37	0	0	0	0	0	0	0
SYSPROG-PL	33756	1	1	1	38	1	32	1	1	1	1	0	0	0	0	0	0	0
SYSPROG-PL*DL/ID	33757	61	1	1	39	88	185,225	5	0	408	14	0	0	3	1	0	0	0
TSVM	33745	11	1	1	33	1	35	1	0	11	11	0	0	0	0	0	0	0
TSVM*DL/ID	33745				34													
XSYM	33066	1	1	1	13	7	215	7	7	1	0	1	0	0	0	0	0	0
XSYM*DL/ID	33067	23	1	1	14	0	0	0	0	23	23	0	0	0	0	0	0	0
***					537		502,663			1,272								

Figure A: Sample Printout of LIST-FILE-STATS PROC

SYSTEM MAINTENANCE PROCEDURES: SYSPROG ACCOUNT

FILE STATISTICS REPORT: STAT-FILE DICTIONARY

| The STAT-FILE is created as part of a system update. The SYS-GEN tape |
contains the standard dictionary for this file.

Creation of the STAT-FILE dictionary and data areas is part of a system update. 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.

Dictionary items as on a SYS-GEN tape are shown in Figure A. A more complete description of these items is given in the Appendix.

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.

Running the ALL-FILE-STATS PROC also clears the data file and builds the current file statistics.

The STAT-FILE data area will also be empty after a file-restore is done, because Attribute 1 of the DL/ID is a "DY". This is desirable as the statistics would no longer be applicable.

It is helpful to make synonym accounts in the System Dictionary as Q-pointers, leaving only the unique and principal accounts as D-pointers. The Statistics Report will then summarize accounts using the same unique account names.

SYSTEM MAINTENANCE PROCEDURES: SYSPROG ACCOUNT

STAT-FILE.	D/CODE	A/AMC	V/CONV.....	V/CORR...	V/TYPE	V/MAX
ORDER.	A	0			R	4
ACCNAME.	A	1		G*1	L	0
ACCOUNTNAME	S	1		G*1	L	0
FILE.	A	1		G1*2	L	20
FILENAME.	A	1		G1*1	L	10
LEVEL.	A	2			R	3
BASE.	A	3			R	5
MOD.	A	4			R	4
SEP.	A	5			R	2
ITEMS.	A	6	MDO,		R	5
SIZE.	A	7	MDO,		R	10
MAX-I	A	8	MDO,		R	6
MIN-I	A	9	MDO,		R	4
MX/GP	A	10			R	3
MN/GP	A	11			R	3
FRAMES	A	12	MDO,		R	9
25%	A	13			R	3
50%	A	14			R	3
75%	A	15			R	3
100%	A	16			R	3
125%	A	17			R	3
150%	A	18			R	3
200%	A	19			R	3
>200%	A	20			R	3
GFE	A	21			R	3
ROTOTAL	A	22			L	3
RITEMS	A	23			L	3
POINTERR	A	24			L	3
OERR	A	25			L	3
ROMAX	A	26			L	3
ROMIN	A	27		F;6;7;/	L	3
AVE.SIZE	S	99		F;5;4;*;12	R	7
SECONDARY.	A	30		' -	R	8

Figure A: STAT-FILE Dictionary

SYSTEM MAINTENANCE PROCEDURES: SYSPROG ACCOUNT

DETERMINING MEMORY BUFFER STATUS: BUFFERS VERB

| The BUFFERS verb may be used to obtain statistics regarding memory |
buffers.

To generate a buffer report, enter the BUFFERS verb in the form:

BUFFERS {(options)}

The status of each memory buffer used as virtual memory is displayed on the screen (or on the printer with the 'P' option). The status taken is a 'snapshot' and is self-consistent. The column labelled "LOC" is the location of the buffer in the memory. The number of buffers displayed is (k of memory)*2-4. Buffers are sorted by buffer number, unless the 'S' option is used, in which case the listing is sorted by frame.

The 'N' option inhibits paging of output to the terminal. The 'P' option directs output to the printer. The 'T' option produces a memory contents report. Information displayed includes the number of buffers occupied by the monitor, ABS frames, work space frames, user program data frames, I/O busy buffers, memory locked frames, and write required frames. The 'Z' option prints the status of buffers with a FID of zero (i.e., monitor code).

FIDs which print as X'FFFF10' through X'FFFF1F' are fake FIDs used to reserve memory buffers for tape I/O and vary according to the tape record size attached. Fake FIDs which print as X'FFFFFB' through X'FFFFDA' are the PIBs.

Figure A presents the general form of the BUFFERS verb. Sample usage of BUFFERS is illustrated in Figure B.

```
-----
BUFFERS {(options)}
```

```
-----
| N - inhibit paging of terminal output |
| P - output to printer                 |
| S - sort by frame                     |
| T - produce memory contents report    |
| Z - show monitor code buffers         |
|-----|
```

Figure A: General Form of BUFFERS Verb

```
-----
:BUFFERS <cr>
```

LOC:	FID:	I/O BUSY	CORELOCK	M.WRITE	WRITE REQD
002000	FFFFFC		X		X
002200	FFFFFB		X		X
002400	FFFFFA		X		X
002600	FFFFF9		X		X
002800	FFFFF8		X		X
002A00	FFFFF7		X		X
002C00	FFFFF6		X		X
etc.					

```
:BUFFERS (T) <cr>
```

```
MEMORY CONTENTS:
```

```
3453 MONITOR BUFFERS
70  ABS FRAMES
312 WORK SPACE FRAMES
261 USER PROGRAM DATA FRAMES

0  I/O BUSY FRAMES
205 MEMLOCKED BUFFERS
9  WRITE REQUIRED BUFFERS
```

Figure B: Sample Usage of BUFFERS Verb

SYSTEM MAINTENANCE PROCEDURES: SYSPROG ACCOUNT

SYSPROG UTILITY VERBS: LOCK-FRAME AND UNLOCK-FRAME

This topic describes two special utility verbs in the SYSPROG account.

LOCK-FRAME VERB

The LOCK-FRAME verb locks a frame in main memory. The general form of this verb is:

LOCK-FRAME number

where "number" is a decimal frame number. The LOCK-FRAME verb responds with the absolute hexadecimal word address of the memory buffer in which the frame is locked. The frame remains locked in memory until it is released by the UNLOCK-FRAME verb.

UNLOCK-FRAME VERB

The UNLOCK-FRAME verb clears the memory locked buffer status of the frame indicated. The general form of this verb is:

UNLOCK-FRAME number

where "number" is a decimal frame number.

Figure A presents the format of LOCK-FRAME and UNLOCK-FRAME verbs.

```
-----  
LOCK-FRAME number <----- Core-locks specified frame  
UNLOCK-FRAME number <----- Unlocks specified frame  
-----
```

Figure A: General Forms of SYSPROG Utility Verbs

GROUP LOCKS AND EXECUTION LOCKS

The LIST-LOCKS verb displays group locks. The CLEAR-BASIC-LOCKS verb clears (resets) all execution locks. The :INIT-SYSTEM verb clears all group locks and execution locks.

LIST-LOCKS

The LIST-LOCKS verb lists the beginning FID of each group that is locked along with the line number which has the group locked. There are up to 128 group locks.

On earlier releases, groups become locked by the file-save processor, or with DATA/BASIC READU, READVU and MATREADU statements, or PROC F-UREAD and FBU commands, for example. Groups become unlocked by write operations into the group or by specific commands issued by a process. If an attempt is made to lock a group when all 128 locks are being used, the process will wait for a lock to become available.

DATA/BASIC and PROC statements which previously used group locks now use item locks instead (see the next topic).

Figure A presents an example of the LIST-LOCKS command.

CLEAR-BASIC-LOCKS

The CLEAR-BASIC-LOCKS verb resets the 64 execution locks (0-63) that may be set by DATA/BASIC and PROC.

:INIT-SYSTEM

The :INIT-SYSTEM verb unlocks all locked groups and resets all execution locks.

```
-----  
:LIST-LOCKS <cr> <----- Display lines that have groups locked.  
                               Show FID of first frame in group.  
09 17319  
15 31712  
07 15334  
-----
```

Figure A: Sample Usage of LIST-LOCKS Verb

SYSTEM MAINTENANCE PROCEDURES: SYSPROG ACCOUNT

ITEM LOCKING: OVERVIEW

Item locking is a system facility which allows individual items to be locked without locking the remaining items in the same group.

Item locking allows single items to be locked without affecting other items in the same group (as was previously the case using group locks) and allows any number of items to be locked simultaneously.

The characteristics of the feature are:

- . Number of item locks available is user-selectable
- . Locking an item does not lock any other item
- . All locks set by a process may be deleted at once

Item locks are now used instead of group locks by PROC, DATA/BASIC and ALL.

This facility is implemented by use of a table containing, for each locked item, a four-byte lock number derived from the item-id and the base of the file. The lock number is almost certain to be unique for each item and the probability of locking an item unnecessarily is therefore very low. The item lock table is accessed via a hashing system for speed.

The facing page shows the general format of the commands available with this feature. These are described in detail on the following pages.

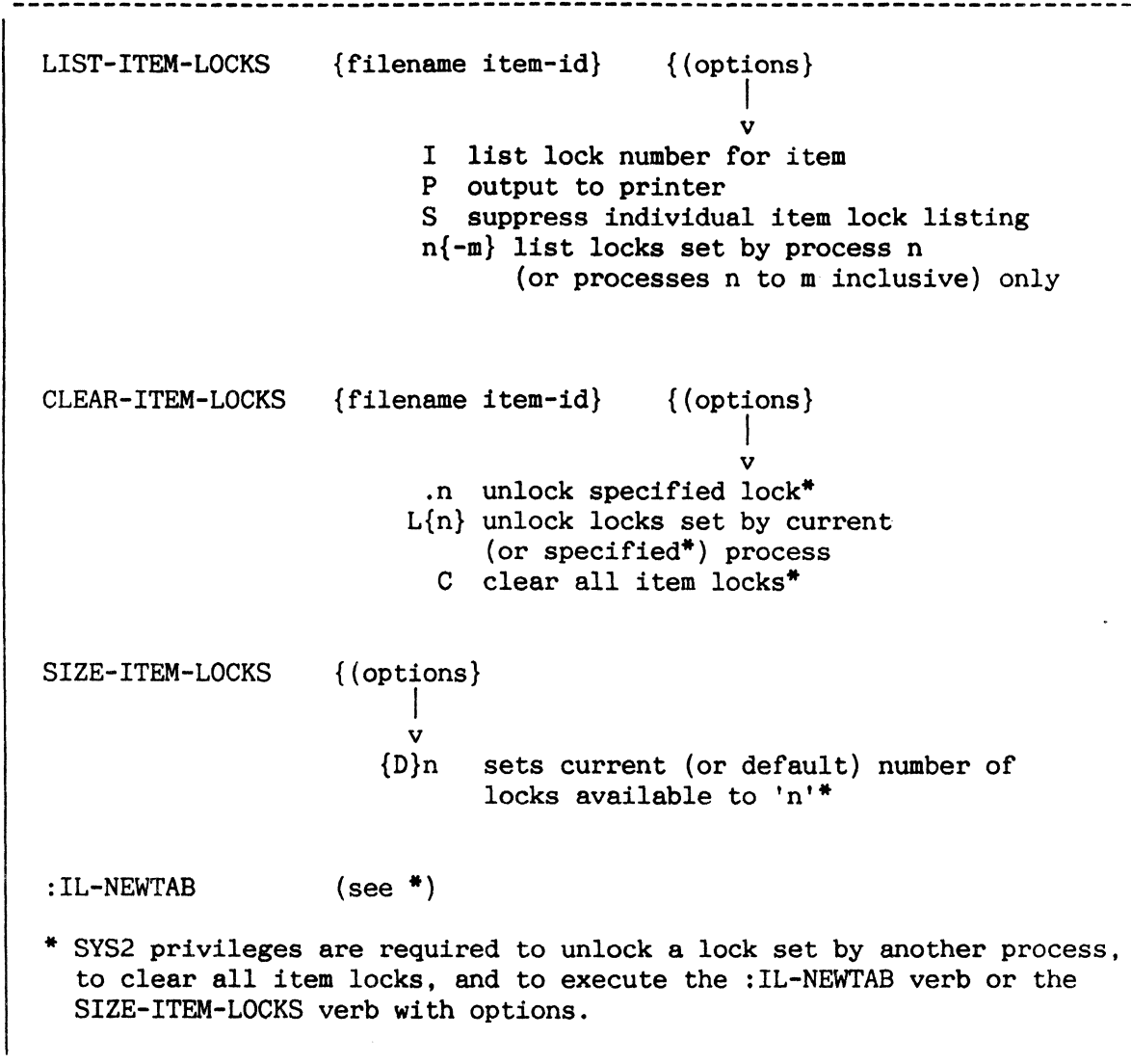


Figure A : Item Locking - Summary of Verbs

SYSTEM MAINTENANCE PROCEDURES: SYSPROG ACCOUNT

ITEM LOCKING: LIST-ITEM-LOCKS VERB

LIST-ITEM-LOCKS lists all or selected item locks currently set, or the process which locked a specified item, or the item lock number corresponding to a specified item.

The LIST-ITEM-LOCKS command has the general format:

LIST-ITEM-LOCKS {filename item-id} {(options)}

where option may be:

- I list the lock produced by the specified item (filename and item-id must be entered with this option)
- P direct output to printer (or other currently-assigned spooler device)
- S suppress listing of individual locks (list number of locks only)
- n list locks for process 'n' only
- n-m list locks for processes 'n' to 'm' inclusive

This command entered without parameters or options lists all item locks currently set, by line and context level, followed by a count of the total number of item locks set. 'Context level' refers to the level of PERFORM statement currently being executed within a DATA/BASIC program.

Specification of a filename and item-id without options causes the process holding a lock on that item (if any) to be reported. If the option 'I' is also entered, the four byte lock number corresponding to the item-id is displayed instead, represented as eight hex digits.

Note: the file must be accessible from the account on which the command is executed.

Option 'S' suppresses the detailed listing of locks set; option 'n' limits the report, and the count, to the locks set by process 'n'; option 'n-m' limits the report and the count to locks set by processes 'n' to 'm' inclusive.

Note: if 'file-name' is entered, 'item-id' must also be entered. The 'I' option is ignored if file-name and item-id are omitted; if file-name and item-id are specified all options except 'I' and 'P' are ignored. Otherwise, any number of options may be specified in any order, and comma separators do not need to be used.

The facing page illustrates the use of the LIST-ITEM-LOCKS command.

```

:LIST-ITEM-LOCKS <cr>                                list all item locks currently set

LI#  LEVEL  LOCK.... LOCK.... LOCK.... LOCK.... LOCK.... LOCK.... LOCK..
      67E4BE49
  4    0      4A32BE3A
  9    0    0062DBFC 0072DC04 02D8DBE0
      1    251F7021 25217022

[973] 7  LOCKS LISTED

:LIST-ITEM-LOCKS (3-5 <cr>                            list locks set by processes
                                                         3, 4 and 5

LI#  LEVEL  LOCK.... LOCK.... LOCK.... LOCK.... LOCK.... LOCK.... LOCK..
      67E4BE49 4A32BE3A

[973] 2  LOCKS LISTED

:LIST-ITEM-LOCKS (S9P<cr>                             lists number of locks set by
                                                         process 9, with output to
                                                         printer (or other assigned
                                                         device)

[973] 5  LOCKS LISTED

:LIST-ITEM-LOCKS BP TESTPROG <cr>                    reports process which locked
                                                         item TESTPROG in file BP

[975] ITEM IS LOCKED BY LINE 4 AT LEVEL 0

:LIST-ITEM-LOCKS BP TESTPROG (I <cr>                list lock number for item
                                                         TESTPROG in file BP

4A32BE3A

```

Figure A : Examples of Use of LIST-ITEM-LOCKS Verb

ITEM LOCKING: CLEAR-ITEM-LOCKS VERB

CLEAR-ITEM-LOCKS unlocks one, or selected, or all current item locks.
SYS2 privileges are required to unlock a lock set by another process.

The CLEAR-ITEM-LOCKS command has the general format:

CLEAR-ITEM-LOCKS {filename item-id} {(options)}

where options may be:

.n unlock a single lock (specified in the form of eight hexadecimal digits, as listed using LIST-ITEM-LOCKS).

L{n} unlock all item locks set by current (or specified) process.

C clear the entire item lock table.

The command requires that either a file-name and an item-id, or a single option, is entered.

If a filename and an item-id are specified, the lock on that item is cleared (if set).

Note: the file must be accessible from the account on which the command is executed.

If a lock is specified directly (by entering .n as an option) that lock is cleared (if set). SYS2 privileges are required if the lock was not set by the current process (the process from which command is executed).

Option 'L' clears all locks set by the current process.

Option 'Ln' clears all locks set by process 'n'. SYS2 privileges are required if process 'n' is not the current process.

Option 'C' clears all item locks currently set. SYS2 privileges are required to use this option.

The facing page illustrates the use of the CLEAR-ITEM-LOCKS command.

```
-----  
:CLEAR-ITEM-LOCKS BP TESTPROG <cr> clear lock on item TESTPROG in file BP  
:CLEAR-ITEM-LOCKS (L8 <cr>          clear all locks set by line  
                                     (process) 8  
:CLEAR-ITEM-LOCKS (C <cr>          clear all item locks  
:CLEAR-ITEM-LOCKS (.4A32BE3A       clear item lock 4A32BE3A  
-----
```

Figure A : Examples of Use of CLEAR-ITEM-LOCKS verb

ITEM LOCKING: SIZE-ITEM-LOCKS AND :IL-NEWTAB VERBS

SIZE-ITEM-LOCKS displays or changes the number of item locks available and the default on re-initialising the item lock table.
:IL-NEWTAB unconditionally rebuilds the item lock table with the current default number of entries. SYS2 privileges are required to execute these verbs (except SIZE-ITEM-LOCKS without options). :IL-NEWTAB is usually only available on the SYSPROG account.

The SIZE-ITEM-LOCKS command has the general format:

SIZE-ITEM-LOCKS {(options)}

where options may be:

{D}n set number of locks available (or default locks available) to 'n'.

Execution of this command without options displays the current and default item lock table size as a number of locks. Both are initially set to 350.

Specifying just a decimal number as an option resets the current number of locks available to that number, provided no item locks are currently set. If any locks are set the message:

[971] ITEM LOCK TABLE MUST BE EMPTY TO CHANGE THIS SIZE

is displayed.

Entering 'Dn' as an option does not change the current table size, but alters the default size. The default size is put into effect only when :IL-NEWTAB is executed.

The verb :IL-NEWTAB unconditionally rebuilds the item lock table with the current default number of entries. It should only be used in case of an irrecoverable error in the item locking system. Space occupied by the current item lock table is lost to the system until a file restore is carried out. This SYSPROG verb requires no options or parameters.

The facing page illustrates the use of the SIZE-ITEM-LOCKS and :IL-NEWTAB commands.

```
:SIZE-ITEM-LOCKS <cr>

[970] CURRENT ITEM LOCK TABLE SIZE IS 350
      DEFAULT ITEM LOCK TABLE SIZE IS 350

:SIZE-ITEM-LOCKS (500 <cr>

[970] CURRENT ITEM LOCK TABLE SIZE IS 500
      DEFAULT ITEM LOCK TABLE SIZE IS 350

:SIZE-ITEM-LOCKS (D400 <cr>

[970] CURRENT ITEM LOCK TABLE SIZE IS 500
      DEFAULT ITEM LOCK TABLE SIZE IS 400

::IL-NEWTAB <cr>

[970] CURRENT ITEM LOCK TABLE SIZE IS 400
      DEFAULT ITEM LOCK TABLE SIZE IS 400
```

Figure A : Example of Use of SIZE-ITEM-LOCKS and :IL-NEWTAB Commands

VERIFICATION OF ASSEMBLY LANGUAGE ROUTINES FROM TAPE

A copy of system software may be kept in T-DUMP format on tape for periodic system verification. This frees space which would otherwise be needed for the SYSTEM-OBJECT file.

To copy assembly language routines to tape, the following procedure is suggested:

A. Upon receipt of a full system update tape:

1. Load the assembly language verification file as directed by the updating instructions.
2. Dump (ST-DUMP) SYSTEM-OBJECT by frame.
3. Copy the modes necessary for coldstart tape sections to a temporary file by entering:

```
:FORM-LIST SYSTEM-OBJECT COLDLIST <cr>
```

```
xx ITEMS SELECTED.  
>COPY SYSTEM-OBJECT <cr>  
TO:(DICT %TEMP-SO% 13,1) <cr>
```

```
:COPY SYSTEM-OBJECT COLDLIST <cr>  
TO:(DICT %TEMP-SO%) <cr>
```

4. Delete the data section of the SYSTEM-OBJECT file.
5. Recreate the data section of the SYSTEM-OBJECT file with a smaller modulo and copy the coldstart modes by entering:

```
:COPY DICT %TEMP-SO% * <cr>  
TO:(DATA SYSTEM-OBJECT 13,1) <cr>
```

6. Delete the temporary %TEMP-SO% file.
7. Put the tape away and ensure that it will never be overwritten.

B. If the system includes additional assembly language routines:

1. Dump and then delete from the disk the additional modes.
2. Be sure to indicate the contents of the tape.

SYSTEM MAINTENANCE PROCEDURES: SYSPROG ACCOUNT

C. As additional updates are received:

1. T-LOAD the latest SYSTEM-OBJECT tape (it may be necessary to temporarily copy some files to tape to allow enough space).
2. Follow the update instructions accompanying the update.
3. Create a new system tape as outlined in step A above.

D. To verify the system:

1. Mount the tape(s) created above.
2. Enter MT-VERIFY.

To verify a system from a SYS-GEN tape:

1. Use the SPACE PROC on SYSPROG to position the tape at the start of the SYSTEM-OBJECT file.
2. Enter MT-VERIFY. Note that if patches are made, mismatches in affected modes will occur.

11. TERMINAL INDEPENDENT PROCESS HANDLER (TIPH)

=====

| This section describes the functions and use of the Terminal Independent
| Process Handler (TIPH), which allows processes to be run without an
associated terminal.

TERMINAL INDEPENDENT PROCESS HANDLER

OVERVIEW

A "process" is defined as the session which occurs between logging on and logging off. Each process is associated with a line (or port), and (normally) each line is associated with a terminal. The Terminal Independent Process Handler (TIPH) initiates a process without an associated terminal. Any terminal output (such as error messages, logon/logoff messages, etc.) is placed in a spooler hold file, the hold file number being assigned by the spooler processor. TIPH may be used for those tasks which do not require operator responses, such as file-saves, massive updates, etc.

TIPH provides a kind of foreground/background capability that allows processes to run without the need of a terminal. "Resident" programs, such as a task scheduler or a transaction logger, may use TIPH. TIPH increases the efficiency of the system by freeing the terminals for their intended purpose, which is user interaction.

A TIPH process is like any other process, except that it is not associated with a terminal. It has its own work space, and has available to it all of the resources of the system, with the exception of terminal input. You cannot access the system through a terminal after the line associated with it has been logged on by a TIPH process.

Although terminal input is not allowed, you may "stack" input in the command stream sent to the TIPH processor. This allows programs to be run which require operator input, where the input responses are known in advance. The processor passes one command string to each input required by the program (this is similar to the secondary input buffer used by the PROC processor).

Once activated, a TIPH process takes its place in the process schedule queue. Because interactive processes are given priority, TIPH processes have low priority by default.

Because a TIPH process uses its own PCB, work space, etc., its port number must be defined on the system, i.e., the port allocated to the TIPH process must be within the configuration of the system.

Certain lines may be designated as TIPH-only lines by the system hardware. These lines may be displayed by entering the WHAT verb from TCL. Normal lines (those normally associated with a terminal) may be designated as being available for use by TIPH by using the PH-ALLOCATE verb.

Note: Your account must have at least 27 additional frames of work space (specified in the System Dictionary) to initiate a TIPH process. You may log the TIPH port onto an account other than the current user account. In this case you must have SYS2 privileges.

A PH-HISTORY file exists in the System Dictionary, which contains all of the information concerning each TIPH process. This information is dynamic as the process continues, displaying the status of the TIPH processes.

The LOGOFF verb will not deactivate (logoff) a TIPH process.

TIPH COMMANDS

The TIPH verbs and PROCs are summarised in Figure A. The remaining topics in this section describe each in detail.

PH-ALLOCATE	Allocates a port to be available for use by TIPH.
PH-DELETE	Designates an allocated port as no longer being available for use by TIPH.
PH-KILL	Aborts the designated TIPH process.
PH-LINES	Lists the lines currently available on which to activate a TIPH process.
PH-RESUME	Reactivates a temporarily halted TIPH process.
PH-START	Initiates a TIPH process.
PH-STATUS	Displays the current status of all TIPH processes listed in the PH-HISTORY file.
PH-SUSPEND	Temporarily halts a TIPH process.

Figure A: Summary of TIPH Commands

TERMINAL INDEPENDENT PROCESS HANDLER

PORT ASSIGNMENT: PH-ALLOCATE, PH-DELETE, PH-LINES

Certain ports may be designated as TIPH-only ports by the system hardware. These are always allocated to TIPH. Lines which are normally associated with a terminal may also be used by TIPH. Before a normal line may be used by a TIPH process, it must be listed in a table of available TIPH ports. The PH-ALLOCATE verb enters the designated line into the table. The PH-DELETE verb removes it from the table. The PH-LINES verb lists the lines available for use by a TIPH process.

PH-ALLOCATE

The format of the PH-ALLOCATE verb is:

```
PH-ALLOCATE {line#}
```

where "line#" is the number of the line to be allocated. If this number is not entered, it will be prompted for (Figure B). The entered port number must be within the range set by the particular system's hardware configuration (this may be displayed by entering the WHAT verb from TCL). The line remains allocated until either the PH-DELETE verb is used, or the system undergoes a coldstart ('X' option).

TIPH-only lines (as designated by the hardware) need not be allocated via the PH-ALLOCATE verb.

Note: A normal line designated for use by TIPH via PH-ALLOCATE may still be used as a normal terminal line when a TIPH process is not running on it. A TIPH process may not be initiated on a line already logged on or in use by another TIPH process.

PH-DELETE

The format of the PH-DELETE verb is:

```
PH-DELETE {line#}
```

If "line#" is not entered, the system prompts for the line to be deleted from the TIPH list. After the line is deleted, the line is no longer available for use by TIPH, unless it is reallocated with the PH-ALLOCATE verb.

PH-LINES

The format of the PH-LINES verb is:

```
PH-LINES
```

This verb lists all the lines currently available for use by TIPH. If an allocated line is already in use by TIPH (or any other process), it is not displayed here.

```

PH-ALLOCATE {line#}

PH-DELETE {line#}

PH-LINES
    
```

Figure A: General Form of PH-ALLOCATE, PH-DELETE, and PH-LINES Verbs

```

:PH-ALLOCATE <cr> <----- Evoke the PH-ALLOCATE verb.

INPUT LINE TO ALLOCATE? 1 <cr> <- Designates that Line 1 may be used by a
TIPH process.

:PH-DELETE <cr> <----- Evoke the PH-DELETE verb.

INPUT LINE TO DELETE? 22 <cr> <-- Designates Port 22 as no longer being
available for use by TIPH.

LINE DELETED FROM PHANTOM SERVICE.

:PH-LINES <cr> <----- Display the lines currently available
LINES AVAILABLE FOR TIPH USE      for use by a TIPH process. In this
1                                  example, Ports 1, 11 and 16 are
11                                 available.
16

3 ITEMS LISTED
    
```

Figure B: Example of Correct Usage of PH-ALLOCATE, PH-DELETE, and PH-LINES Verbs

INITIATING A TIPH PROCESS: THE PH-START VERB

A TIPH process is initiated by entering the PH-START verb from TCL. The TIPH process begins by logging on the specified line. After executing all the commands in the command stream, the line is logged off. All terminal output, including the logon and logoff messages, is placed in a hold file. The command stream may contain any TCL verb except LOGTO.

The format of the PH-START verb is:

PH-START

The TIPH processor prompts for a task-id, the desired line number, any desired options, an account name, an optional password, and the command(s) to be executed (Figures A and B).

The task-id is a user-selected alphanumeric character string which identifies the TIPH process in the PH-STATUS display (see the topic DISPLAYING THE TIPH PROCESSES: THE PH-STATUS VERB). It is not the item-id of the item in the PH-HISTORY file. It may be up to 140 alphanumeric characters; however, only the first 10 are displayed.

The line number must be either a TIPH-only line or have been allocated (via PH-ALLOCATE) for use by a TIPH process prior to evoking the PH-START verb. If the line cannot be used by TIPH, the message "NO LINE AVAILABLE" is displayed. If another user (or another TIPH process) is already logged on to the specified line, the message "LINE ALREADY LOGGED ON" appears. Note that if another TIPH process is logged to the line, TIPH will not queue the current user's TIPH process to wait for it to finish (logoff).

The available options are explained in the topic PH-START OPTIONS.

The account name you enter is the account the TIPH process will logon to execute the command stream. The password is that specified in the System Dictionary for the particular account. If there is no password, then just press <RETURN>.

The command stream may consist of TCL statements, a PROC name, or inputs necessary to perform the desired tasks. A null input is stacked by entering the character '<' in the command stream. Note that the verb LOGTO cannot be used. Each command (or input value) is entered one at a time, separated by pressing <RETURN>. The verb "OFF" should be the last entry in the command stream.

If an operation is to be performed requiring operator input (such as a DATA/BASIC program which contains INPUT statements), the input value(s) must be included in the command stream. Each is entered individually, separated by <RETURN>. Note: To decrease the number of commands entered through TIPH, it is recommended that you create a PROC prior to using TIPH, and then evoke this PROC in the command stream.

```

PH-START
    
```

Figure A: Format of the PH-START Verb

```

:PH-START <cr> <----- Evoke the TIPH processor.

ENTER TASKID FOR PH TASK? TEST <cr> <-- The task-id identifies the TIPH
process in the PH-HISTORY file. For this example, the task-id is
'TEST'.

ENTER LINE# FOR PH TASK? 14 <cr> <----- This TIPH process will logon to
Line 14 to execute the command string. If this line has not
been allocated to be used by TIPH, or if the associated
terminal is already in use, an error message appears, and the
process is aborted.

ENTER OPTIONS? <cr> <----- The options are listed in the
topic PH-START OPTIONS. For this example no options are specified
(i.e., all terminal output goes to a hold file).

ENTER ACCOUNT FOR PH JOB? INV <cr> <--- The TIPH process logs on to the
INV account to execute the command string.

ENTER PASSWORD? KRIS <cr> <----- The INV account's password is
KRIS.

ENTER COMMAND STREAM <CR TO END>? <----- This TIPH process performs a SORT
>SP-ASSIGN HS <cr> on the INVENTORY file. The
>SORT INVENTORY (P) <cr> output is printed to a hold file
>OFF <cr> (separate from the hold file
><cr> containing the TIPH logon and
logoff messages). The TIPH
process ends by logging off the
line.

TASK TEST STARTED ON LINE 14 FOR
ACCOUNT INV
    
```

Figure B: Example of Correct Usage of TIPH

TERMINAL INDEPENDENT PROCESS HANDLER

PH-START OPTIONS

Normally, all terminal output for a TIPH process is passed to a hold file, so that you may print the output when the process is completed. Certain options are available to either prevent the creation of the hold file or to print the hold file automatically when the TIPH process is complete. You may also hush the TIPH processor message which appears on the "master" terminal.

The following options are available:

- H Hush the message displayed by the PH-START verb when the TIPH process has been started. This message is normally displayed on the terminal which started the TIPH process. This option may be useful if the TIPH process itself starts another TIPH process, to prevent the message from being passed into the hold file. The 'H' option is not shown in the TIPH status display (see the topic DISPLAYING THE TIPH PROCESSES: THE PH-STATUS VERB), because it pertains to the "master" process, rather than to the TIPH process.
- P Print the hold file on the printer after the TIPH process has ended. The result is the same as if you had printed the hold file using SP-EDIT (see the topic EXAMINING JOBS WAITING TO BE PRINTED in Section 7). The hold file contains all terminal output generated by the TIPH process (such as the logon and logoff messages, the responses from the system after a SELECT, etc.). Note that the hold file is not deleted by this option.
- N Do not maintain a hold file for terminal output. This option results in all terminal output being "lost". This may be useful when executing update programs, file-saves, etc., when the hold file would only contain messages not deemed important for a particular application.
- T Display all terminal output on the terminal associated with the line. This may be used, for example, to print a report on a hardcopy terminal. If there is no associated terminal, or if the process is executing on a TIPH-only line, the TIPH process will "hang" until the process is aborted. No hold file is created because the output is to the terminal.

Because the logon and logoff messages are also sent to the terminal, you are cautioned that, if the line is associated with a hardcopy terminal, the messages may appear on the same page as the report.

The options 'HN', 'HP', and 'HT' are also available; they have the same effect as the 'N', 'P' and 'T' options (respectively), but also cause the PH-START message to be hushed. Even though no comma is placed between the 'H' and the other options, the 'H' will not appear in the PH-STATUS display.

TERMINAL INDEPENDENT PROCESS HANDLER

H	<-----	Hush the PH-START message.
P	<-----	Print the hold file on the printer when the TIPH process is complete.
N	<-----	No logging; no hold file is created. All terminal output is lost.
T	<-----	Allows normal terminal output. Assumes a terminal is associated with the TIPH port. No hold file is created.
HP	<-----	Hush the PH-START message. In addition, print the hold file on the printer when the TIPH process is complete.
HN	<-----	Hush the PH-START message. In addition, do not create a hold file. All terminal output is lost.
HT	<-----	Hush the PH-START message. In addition, allow terminal output. No hold file is created.

Figure A: PH-START Verb Options

TERMINAL INDEPENDENT PROCESS HANDLER

HALTING AND REACTIVATING A TIPH PROCESS

A TIPH process may be aborted or temporarily halted using the PH-KILL and the PH-SUSPEND verbs, respectively. A temporarily halted process may be reactivated using the PH-RESUME verb.

The general form of the three verbs is:

PH-KILL task-id line#

PH-SUSPEND task-id line#

PH-RESUME task-id line#

where "task-id" is your specified identification of the TIPH process, and "line#" is the line on which the TIPH process is running. An asterisk (*) in place of the task-id implies the process running on the specified line. An asterisk in place of the line# implies all processes with the specified task name. Any of the three verbs followed by a single asterisk implies all TIPH processes (the asterisk following PH-RESUME reactivates all temporarily halted processes).

PH-KILL causes the specified process to abort (logoff). Any updating being performed will not be undone. If the TIPH process (or processes) is currently running on an account other than your own, SYS2 privileges are required.

PH-SUSPEND temporarily halts a TIPH process. This may be useful if additional processing resources are required for another line (i.e., to speed up its process). By suspending the TIPH process(es), more time is available for the other jobs. SYS2 privileges are required if the TIPH process (or processes) is executing on an account other than your own.

Note: The TIPH process must already be active (beyond logging on) before the PH-SUSPEND verb may be used. Otherwise, it may "hang" the line. You must coldstart ('X' option) the system to release the line. Because the remainder of the system is unaffected, it is recommended that you wait until all other users are logged off of the system to avoid inconvenience and possible group format errors.

PH-RESUME reactivates a TIPH process which had been temporarily halted with the PH-SUSPEND verb. SYS2 privileges are required, as noted above. Note that a TIPH process may be reactivated from any terminal, not just the terminal from which it was halted.

PH-KILL task-id line#
PH-SUSPEND task-id line#
PH-RESUME task-id line#

Figure A: General Form of the PH-KILL, PH-SUSPEND, and PH-RESUME Verbs.

TERMINAL INDEPENDENT PROCESS HANDLER

DISPLAYING THE TIPH PROCESSES: THE PH-STATUS VERB

| PH-STATUS displays the status of a TIPH process as it is recorded in the |
PH-HISTORY file.

The general form of PH-STATUS is:

```
PH-STATUS {task-id line#} {(P)}
```

where "task-id" is the user-specified name of the TIPH process, and "line#" is the line to which the process was assigned. If an asterisk (*) is entered in place of "task-id", then all processes in the PH-HISTORY file which ran on the specified line are displayed (see note below). If an asterisk is entered in place of the line number, then all processes with the specified name are displayed. PH-STATUS with no task-id or line# specified displays the status of all processes in the PH-HISTORY file. The 'P' option specifies printing the status on the printer.

PH-STATUS displays the following information: the task-id, the line number, the account to which the TIPH process is logged, the name of the account which started the process, the date and time started, the current status of the process, the PH-START options (if any), and the number of the hold file which contains the terminal output (Figure B).

The status of each TIPH process is indicated by displaying an alphabetic status code. 'A' indicates that the process is still active; 'S' indicates that the process has been suspended; 'T' indicates that the process has terminated normally; 'X' indicates that the process was aborted by the process itself (abnormal abort); 'K' indicates that the process was aborted by the PH-KILL verb.

'OUTFL#' on the PH-STATUS display is the output file, i.e., the spooler hold file to which the terminal output has been sent. Note: If the process printed something to a hold file (using SP-ASSIGN HS), this hold file number is not displayed. The displayed hold file number contains the output which normally would have appeared on the terminal. There is no terminal output file if the 'N' option was specified with the PH-START verb.

Note:

Items in the PH-HISTORY file are not automatically deleted. Therefore, when displaying all TIPH processes, many processes may be shown with the same line number and name. The data portion of the file should be cleared occasionally, using the CLEAR-FILE verb, as shown in Figure C. Do not clear the file while a TIPH process is still active!

```
-----
| PH-STATUS {task-id line#} {(P)} |
|-----
```

Figure A: General Form of the PH-STATUS Verb

```
-----
| TASK-ID... LINE ACCOUNT... STARTER-ACC DATE.... TIME... STATUS OP OUTFL# |
|-----
```

TASK-ID...	LINE	ACCOUNT...	STARTER-ACC	DATE....	TIME...	STATUS	OP	OUTFL#
LGRADE	4	REG	ADM	14 FEB 1985	08:05:37	K	T	
FS	7	SYSPROG	DOC	21 FEB 1985	17:30:17	A	P	023
CMP.BP	12	CHRIS	CHRIS	07 FEB 1985	12:05:11	S	P	002
S.INV	17	INV	MASTER	05 FEB 1985	17:00:57	T		003
T3	18	JS	SYSPROG	11 FEB 1985	09:13:45	X	N	

```
-----
|
| where the status code is one of the following:
|
| A = Active
| S = Suspended
| T = Normal termination
| X = Abort by process
| K = Abort by PH-KILL
|
| Note that the 'H' option is not displayed.
|
|-----
```

Figure B: PH-STATUS Display

```
-----
| :CLEAR-FILE (DATA PH-HISTORY) <cr> |
|-----
```

Figure C: Example of Clearing the Data Portion of the PH-HISTORY File

TERMINAL INDEPENDENT PROCESS HANDLER

TIPH INITIALISATION MESSAGES

Initialisation messages are mainly used to indicate errors in the specification of a TIPH process. They may appear while using the PH-START verb.

The following messages exist:

NO LINE AVAILABLE - If you did not specify a line, this message indicates that all the TIPH lines (both TIPH-only and those normally associated with a terminal) are in use. If a line was specified, it indicates that the specified line was not allocated for use by TIPH (using the PH-ALLOCATE verb). You should allocate a line, if necessary, and re-enter the PH-START verb.

PH-PWD ERROR - The incorrect password was entered for the specified account. You should re-enter the PH-START verb.

PH-ID ERROR - No user account was specified. You should re-enter the PH-START verb.

HISTORY FILE PROBLEM - No PH-HISTORY file exists, or the current account does not have a Q-pointer to it. If the file does not exist, you may create it using the CREATE-FILE verb and the EDITOR. The procedure is described in the Appendix at the back of this manual. If no Q-pointer exists, then you should create it using the EDITOR (Figure B).

LINE ALREADY LOGGED ON - The specified line is being used either by an actual user or another TIPH process. You may enter the PH-LINES verb to display the currently available lines.

HISTORY FILE ACCESS ERROR - For some reason, TIPH cannot properly access the PH-HISTORY file.

MASTER PRIVILEGE ERROR - You attempted to use TIPH on an account other than your own, and did not have SYS2 privileges. If you are able, you should log to the SYSPROG account and increase your privilege level.

ACC FILE ACCESS ERROR - for some reason TIPH cannot access the ACC file to indicate which lines have been logged on.

[82] YOUR SYSTEM PRIVILEGE LEVEL IS NOT SUFFICIENT FOR THIS STATEMENT - You attempted to initiate a TIPH process and your account only had SYSO privileges.

NOT ENOUGH WORKSPACE - You attempted to initiate a TIPH process and your account did not have at least 27 additional frames of work space assigned. If you are able, you should log to the SYSPROG account and increase the account's work space assignment.

NO LINE AVAILABLE <-----	Either a line number was not entered, or the specified line was not allocated to be used by a TIPH process (using PH-ALLOCATE).
PH-PWD ERROR <-----	The entered password was incorrect for the specified account.
PH-ID <-----	No user account was specified.
HISTORY FILE PROBLEM <-----	Either the PH-HISTORY file is missing from the SYSTEM dictionary, or a Q-pointer to it does not exist in the current account's M/DICT.
LINE ALREADY LOGGED ON <-----	The specified line is in use by either an actual user or another TIPH process.
HISTORY FILE ACCESS ERROR <-----	TIPH cannot access the PH-HISTORY file.
MASTER PRIVILEGE ERROR <-----	You attempted to log the TIPH process onto an account other than your own, and your account did not have SYS2 privileges.
ACC FILE ACCESS ERROR <-----	TIPH cannot access the ACC file.
[82] YOUR SYSTEM PRIVILEGE <----- LEVEL IS NOT SUFFICIENT FOR THIS STATEMENT	Your account has only SYSO privileges.
NOT ENOUGH WORKSPACE <-----	Your account does not have at least 27 additional frames of workspace assigned.

Figure A: Initialisation Error Messages

PH-HISTORY	---in user account's M/DICT
001 Q	
002 SYSTEM	
003 PH-HISTORY	
004	---null
005	---null
006	---null
007	---null
008	---null
009 L	
010 10	

Figure B: Q-Pointer to PH-HISTORY File

TIPH TERMINATION MESSAGES

| The TIPH Termination Messages are placed in the hold file along with the |
| other terminal output. These messages indicate the reason the process |
halted.

The following are the Termination Messages:

T PROCESS STOP - Indicates that the TIPH process has completed normally and has logged off the line.

T PROCESS HAS BEEN LOGGED OFF BY PH-KILL - Indicates that the TIPH process has been aborted by the PH-KILL verb.

T PROCESS ABORT - Indicates that the TIPH process has aborted due to some reason related to the process itself (i.e., not caused by TIPH). For example, a TIPH process executing a DATA/BASIC program which aborted due to a programming error. This message may be preceded by one of the messages below.

E INPUT ERROR - Indicates that a process prompted for terminal input, and there was no TIPH stacked input available. This may occur if:

- . A TCL verb which requires user input (such as SSELECT or COPY) was inadvertently used without specifying the required subsequent responses in the command.
- . The verb "OFF" was not the last command in the command stream.

E PROCESS IN DEBUGGER - Indicates that the process has trapped to the system debugger. The TIPH process will abort with the *T* PROCESS ABORT message.

TERMINAL INDEPENDENT PROCESS HANDLER

[910] *T* PROCESS STOP <----- TIPH process terminated normally.

[911] *T* PROCESS HAS BEEN <----- TIPH process terminated by PH-KILL.
LOGGED OFF BY PH-KILL

[912] *T* PROCESS ABORT <----- TIPH process aborted due to an
error by the process being executed
(such as a DATA/BASIC program).

[913] *E* INPUT ERROR <----- The TIPH process prompted for
terminal input, and there were no
responses in the command string.

[914] *E* PROCESS IN DEBUGGER <---- The TIPH process has trapped to the
system debugger.

Figure A: TIPH Termination Messages

APPENDIX A

=====

ASCII CHARACTER SET

Decimal	Hex	EBCDIC Equiv.	ASCII Char.	Display	Key	Special Use
0	00	00	NUL	None	<c>@	Delay character
1	01	01	SOH	None	<c>A	Home command
2	02	02	STX	None	<c>B	
3	03	03	ETX	None	<c>C	End of text
4	04	37	EOT	None	<c>D	
5	05	2D	ENQ	None	<c>E	
6	06	2E	ACK	None	<c>F	Cursor forward
7	07	2F	BEL	None	<c>G	Bell
8	08	16	BS	None	<c>H	Backspace
9	09	05	HT	None	<c>I	Tab
10	0A	25	LF	None	<c>J	Cursor down
11	0B	0B	VT	None	<c>K	Vertical address
12	0C	0C	FF	None	<c>L	Screen erase
13	0D	0D	CR	None	<c>M	Carriage return
14	0E	0E	SO	None	<c>N	
15	0F	0F	SI	None	<c>O	
16	10	10	DLE	None	<c>P	Horizontal address; blank compression character
17	11	11	DC1	None	<c>Q	
18	12	12	DC2	None	<c>R	Retype entire line; enable slave printer
19	13	3A	DC3	None	<c>S	Dump screen to slave printer (option)
20	14	3C	DC4	None	<c>T	Disable slave printer
21	15	3D	NAK	None	<c>U, <s>4	Cursor back
22	16	32	SYN	None	<c>V	
23	17	26	ETB	None	<c>W	
24	18	18	CAN	None	<c>X	Cancel line
25	19	19	EM	None	<c>Y	
26	1A	3F	SUB	None	<c>Z, <s>8	Cursor up
27	1B	27	ESC	None	ESC, <c>[EDITOR command delimiter. Evokes SCREENPRO command-mode
28	1C	1C	FS	None		
29	1D	1D	GS	None		
30	1E	1E	RS	None		
31	1F	1F	US	None		
32	20	40	blank		space	
33	21	5A	!	!	!	

ASCII CHARACTER SET

Decimal	Hex	EBCDIC Equiv.	ASCII Char.	Display	Key	Special Use
34	22	7F	"	"	"	String delimiter in ENGLISH AND BASIC
35	23	7B	#	#	#	
36	24	5B	£	£	£	
37	25	6C	%	%	%	
38	26	50	&	&	&	
39	27	7D	'	'	', <c>"	String delimiter in ENGLISH and BASIC
40	28	4D	(((
41	29	5D)))	
42	2A	5C	*	*	*	
43	2B	4E	+	+	+	
44	2C	6B	,	,	, or <c><	
45	2D	60	-	-	-	
46	2E	4B	.	.	., <c>>	
47	2F	61	/	/	/, <c>?	
48	30	F0	0	0	0, <c>)	
49	31	F1	1	1	1, <c>!	
50	32	F2	2	2	2	
51	33	F3	3	3	3, <c>#	
52	34	F4	4	4	4, <c>£	
53	35	F5	5	5	5, <c>%	
54	36	F6	6	6	6	
55	37	F7	7	7	7, <c>8	
56	38	F8	8	8	8, <c>*	
57	39	F9	9	9	9, <c>(
58	3A	7A	:	:	:	
59	3B	5E	;	;	;, <c>:	
60	3C	4C	<	<	<	
61	3D	7E	=	=	=, <c>+	
62	3E	6E	>	>	>	
63	3F	6F	?	?	?	
64	40	7C	@	@	@	
65	41	C1	A	A	<s>A	
66	42	C2	B	B	<s>B	
67	43	C3	C	C	<s>C	
68	44	C4	D	D	<s>D	
69	45	C5	E	E	<s>E	
70	46	C6	F	F	<s>F	
71	47	C7	G	G	<s>G	
72	48	C8	H	H	<s>H	
73	49	C9	I	I	<s>I	
74	4A	D1	J	J	<s>J	
75	4B	D2	K	K	<s>K	
76	4C	D3	L	L	<s>L	
77	4D	D4	M	M	<s>M	
78	4E	D5	N	N	<s>N	
79	4F	D6	O	O	<s>O	
80	50	D7	P	P	<s>P	
81	51	D8	Q	Q	<s>Q	
82	52	D9	R	R	<s>R	

ASCII CHARACTER SET

Decimal	Hex	EBCDIC Equiv.	ASCII Char.	Display	Key	Special Use
83	53	E2	S	S	<s>S	
84	54	E3	T	T	<s>T	
85	55	E4	U	U	<s>U	
86	56	E5	V	V	<s>V	
87	57	E6	W	W	<s>W	
88	58	E7	X	X	<s>X	
89	59	E8	Y	Y	<s>Y	
90	5A	E9	Z	Z	<s>Z	
91	5B	80	[[[ENGLISH string search delimiter
92	5C	E0	\	\	\	RUNOFF control character
93	5D	90]]]	ENGLISH string search delimiter
94	5E	5F	^	^	^	ENGLISH string search delimiter
95	5F	6D	¯	¯	¯	RUNOFF control character
96	60	79	ˆ	@	<cs>0	
97	61	81	a	A	A	
98	62	82	b	B	B	
99	63	83	c	C	C	
100	64	84	d	D	D	
101	65	85	e	E	E	
102	66	86	f	F	F	
103	67	87	g	G	G	
104	68	88	h	H	H	
105	69	89	i	I	I	
106	6A	91	j	J	J	
107	6B	92	k	K	K	
108	6C	93	l	L	L	
109	6D	94	m	M	M	
110	6E	95	n	N	N	
111	6F	96	o	O	O	
112	70	97	p	P	P	
113	71	98	q	Q	Q	
114	72	99	r	R	R	
115	73	A2	s	S	S	
116	74	A3	t	T	T	
117	75	A4	u	U	U	
118	76	A5	v	V	V	
119	77	A6	w	W	W	
120	78	A7	x	X	X	
121	79	A8	y	Y	Y	
122	7A	A9	z	Z	Z	
123	7B	C0	{	[<c>;	
124	7C	6A		\		
125	7D	D0	}]	<c>=	
126	7E	A1	~	^	<c>>	
127	7F	07	DEL	None		
128	80	04		None		
129	81	06		None		
130	82	08		None		
131	83	09		None		

ASCII CHARACTER SET

Decimal	Hex	EBCDIC Equiv.	ASCII Char.	Display	Key	Special Use
132	84	0A		None		
133	85	13		None		
134	86	14		None		
135	87	15		None		
136	88	17		None		
137	89	1A		None		
138	8A	1B		None		
139	8B	20		None		
140	8C	21		None		
141	8D	22		None		
142	8E	23		None		
143	8F	24		None		
144	90	28		None		
145	91	29		None		
146	92	2A		None		
147	93	2B		None		
148	94	2C		None		
149	95	30		None		
150	96	31		None		
151	97	33		None		
152	98	34		None		
153	99	35		None		
154	9A	36		None		
155	9B	38		None		
156	9C	39		None		
157	9D	3B		None		
158	9E	3E		None		
159	9F	41		None		
160	A0	42		None		
161	A1	43		None		
162	A2	44		None		
163	A3	45		None		
164	A4	46		None		
165	A5	47		None		
166	A6	48		None		
167	A7	49		None		
168	A8	4A		None		
169	A9	4F		None		
170	AA	51		None		
171	AB	52		None		
172	AC	53		None		
173	AD	54		None		
174	AE	55		None		
175	AF	56		None		
176	B0	57		None		
177	B1	58		None		
178	B2	59		None		
179	B3	62		None		
180	B4	63		None		
181	B5	64		None		
182	B6	65		None		

ASCII CHARACTER SET

Decimal	Hex	EBCDIC Equiv.	ASCII Char.	Display	Key	Special Use
183	B7	66				None
184	B8	67				None
185	B9	68				None
186	BA	69				None
187	BB	70				None
188	BC	71				None
189	BD	72				None
190	BE	73				None
191	BF	74	Blank			None
192	C0	75	@	@		
193	C1	76	A	A		
194	C2	77	B	B		
195	C3	78	C	C		
196	C4	8A	D	D		
197	C5	8B	E	E		
198	C6	8C	F	F		
199	C7	8D	G	G		
200	C8	8E	H	H		
201	C9	8F	I	I		
202	CA	9A	J	J		
203	CB	9B	K	K		
204	CC	9C	L	L		
205	CD	9D	M	M		
206	CE	9E	N	N		
207	CF	9F	O	O		
208	D0	AO	P	P		
209	D1	AA	Q	Q		
210	D2	AB	R	R		
211	D3	AC	S	S		
212	D4	AD	T	T		
213	D5	AE	U	U		
214	D6	AF	V	V		
215	D7	BO	W	W		
216	D8	B1	X	X		
217	D9	B2	Y	Y		
218	DA	B3	Z	Z		
219	DB	B4	[[
220	DC	B5	\	\		
221	DD	B6]]		
222	DE	B7	^	^		
223	DF	B8	_	_		
224	E0	B9	@	@		
225	E1	BA	A	A		
226	E2	BB	B	B		
227	E3	BC	C	C		
228	E4	BD	D	D		
229	E5	BE	E	E		
230	E6	BF	F	F		
231	E7	CA	G	G		
232	E8	CB	H	H		
233	E9	CC	I	I		

ASCII CHARACTER SET

Decimal	Hex	EBCDIC Equiv.	ASCII Char.	Display	Key	Special Use
234	EA	CD	J	J		
235	EB	CE	K	K		
236	EC	CF	L	L		
237	ED	DA	M	M		
238	EE	DB	N	N		
239	EF	DC	O	O		
240	FO	DD	P	P		
241	F1	DE	Q	Q		
242	F2	DF	R	R		
243	F3	E1	S	S		
244	F4	EA	T	T		
245	F5	EB	U	U		
246	F6	EC	V	V		
247	F7	ED	W	W		
248	F8	EE	X	X		
249	F9	EF	Y	Y		
250	FA	FA	Z	Z		
251	FB	FB	[[<u>System Delimiters:</u>
252	FC	FC	\	\	<c>\	Start Buffer (SB)
253	FD	FD]]	<c>]	Subvalue Mark (SVM)
254	FE	FE	^	^	<c>^	Value Mark (VM)
255	FF	FF	_	_	<c>_	Attribute Mark (AM)
						Segment Mark (SM)

Note: On all recent terminals the position of the SHIFT Key is irrelevant when entering CTRL characters.

APPENDIX B

=====

SYSTEM ERROR MESSAGES

 This appendix lists system error messages located in the ERRMSG file, except for messages prefixed with 'B', which are listed in the DATA/BASIC Reference Manual, and messages prefixed with '#', which are listed in the SCREENPRO Reference Manual.

Message Number	Message
1	ILLEGAL USE OF DOUBLE-QUOTE IN ITEM-ID
2	UNEVEN NUMBER OF SINGLE OR DOUBLE QUOTE-SIGNS ('')
3	'input string' IS NOT A VERB
4	MODE 'mode' CHECKSUM ERROR: FRAME = n MODE = cccc ABS = aaaa
5	THE WORD "word" IS ILLEGAL
6	FILE NAME 'file-name' IS ILLEGAL
7	A VALUE MUST FOLLOW THE HEADING, FOOTING, TAG OR GRAND-TOTAL CONNECTIVE
8	A WINDOW SPECIFICATION STRING MUST FOLLOW THE "WINDOW" CONNECTIVE
9	SYSTEM DL/ID MISSING
10	FILE NAME MISSING
11	FRAME LOCKED AT LOCATION X'nnnn'
13	DATA LEVEL DESCRIPTOR MISSING
14	TAPE INPUT ILLEGAL
15	THE FILE-NAME IS PRECEDED BY AN ILLEGAL CONNECTIVE
17	"WITHIN" VALID ONLY IN COUNT/LIST STATEMENTS
18	LAST WORD MAY NOT BE A CONNECTIVE
19	VALUE WITHOUT AN ATTRIBUTE NAME IS ILLEGAL
21	CONFLICT BETWEEN USER & SYSTEM NAMES
22	MEMORY CONTENTS:
	n MONITOR BUFFERS
	n ABS FRAMES
	n WORK SPACE FRAMES
	n USER PROGRAM DATA FRAMES
	n I/O BUSY FRAMES
	n MEMLOCKED BUFFERS
	n WRITE REQUIRED BUFFERS
23	MAXIMUM NUMBER OF NEW CONTEXT LEVELS EXCEEDED
24	THE WORD "word" CANNOT BE IDENTIFIED
25	"WITH" MAY NOT IMMEDIATELY PRECEDE A VALUE
26	ATTRIBUTE VALUES MAY NOT BOTH PRECEDE AND FOLLOW AN ATTRIBUTE NAME
28	NO CONFIGURATION CHIP

SYSTEM ERROR MESSAGES

Message Number	Message
29	AT LEAST ONE ITEM-ID MUST BE SPECIFIED FOR A "WITHIN"-TYPE STATEMENT
30	FORMAT ERROR IN M/DICT ENTRY DEFINING VERB
32	**A REQUIRED ITEM-ID IS MISSING
33	n SYSTEM PRIVILEGE LEVEL
	n ADDITIONAL FRAMES OF WORK SPACE
42	- ILLEGAL MULTIPLE CONNECTIVES EXIST -
43	ERROR - RECEIVE OR TRANSMIT FUNCTION NOT SPECIFIED
44	ERROR - FORMAT SPECIFICATION INCORRECT
46	ERROR - INCORRECT DEVICE TYPE
53	ERROR - TRANSMIT ID SPECIFIED
58	ID 'n' IS NOT IN TABLE
61	ERROR - INCORRECT OPTION SPECIFICATION
62	ID ENTRY IN TABLE FILE MISSING
63	ERROR - INVALID RECEIVE OPERATION SPECIFIED
64	ONLY ONE ITEM ID IS PERMITTED IN A "WITHIN" TYPE STATEMENT
65	data
	DATA SET NOT READY
66	data
67	TOO MANY IDS SPECIFIED - (MAXIMUM OF 10)
71	AN ILLEGAL CONNECTIVE MODIFIES THE WORD 'word'
72	THE VALUE "value" IS MEANINGLESS
75	DATA SET NOT READY
79	THE NUMBER OF SEPARATE AND CLAUSE SETS CANNOT EXCEED 9
80	A SYSTEM ERROR HAS OCCURRED IN MODE: mode
	THIS MAY BE DUE TO SORT-KEY(S) PRECEDING SELECTION CRITERIA
81	SECURITY CODE VIOLATION
82	YOUR SYSTEM PRIVILEGE LEVEL IS NOT SUFFICIENT FOR THIS STATEMENT
83	FRAME IS PART OF BASIC RUNTIME SYSTEM - NOT UNLOCKED
86	FILE REFERENCE ATTEMPTED ON FILE NOT PREVIOUSLY OPENED
90	TAPE IS NOT T-DUMP FORMAT
91	THE REQUESTED TAPE BUFFER CANNOT BE ATTACHED
93	TAPE NOT ATTACHED
94	END OF FILE
96	BOT
97	END OF TAPE
98	TAPE BUFFER SIZE OF n EXCEEDED
99	n TAPE RECORDS WRITTEN
111	ITEM 'item-id' IS NOT ON FILE
112	NO ENTRIES IN TRACE BUFFER
113	FORMAT ERROR IN TRACE BUFFER
117	A DELETE STATEMENT MUST CONTAIN EITHER ITEM-IDS OR SELECTION CRITERIA
120	'value' NEGATIVE BALANCE NOT PERMITTED
136	'value' DOES NOT MATCH THE G-CORRELATIVE SPECS.

SYSTEM ERROR MESSAGES

Message Number	Message
151	LINE FID PS RETURN STACK
158	AN ILLEGAL CONNECTIVE OF THE FORM "x" MODIFIES "value"
163	ATTRIBUTE FOR SORT-KEY MISSING
164	A D1 OR D2 ATTRIBUTE HAS CORRELATIVES THAT ILLEGALLY PRECEDE THE D-CORRELATIVE
166	ILLEGAL ATTRIBUTE NAME-- 'name'
167	NON-NUMERIC AMC
168	ILLEGAL CHARACTER IN A-CORRELATIVE
169	ILLEGAL A-CORRELATIVE
170	MISSING LEFT PAREN IN A-CORRELATIVE
171	MISSING RIGHT PAREN IN A-CORRELATIVE
172	MISSING RIGHT BRACKET IN A-CORRELATIVE
173	MISSING SEMI-COLON IN A-CORRELATIVE
175	END OF TAPE....
180	MESSAGE NOT IN QUEUE
183	END OF QUEUE
199	INSUFFICIENT WORK SPACE FOR ITEM 'item-id'
200	FILE NAME?
201	'file-name' IS NOT A FILE NAME
202	'item-id' NOT ON FILE
203	ITEM NAME?
204	FILE DEFINITION 'file-name' IS MISSING
205	NO STATEMENTS TO BE ASSEMBLED
206	'item-id' ASSEMBLED
207	UNDEFINED SYMBOLS
208	ERROR IN ITEM-ID LIST
209	'POINTER-FILE' MISSING A 'B' IN THE 13TH ATTR. CONTACT THE SYSTEM MANAGER OR PROGRAMMER
210	FILE 'file-name' IS ACCESS PROTECTED
211	MODE 'item-id' NO ASSEMBLED CODE CAN BE FOUND
212	MODE 'item-id' HAS MISSING OR ILLEGAL FRAME STATEMENT
213	MODE 'item-id' LOCATION COUNTER ERROR AT LINE NO. n
214	MODE 'item-id' EXCEEDS MAX. OF 512 BYTES AT LINE NO. n
215	MODE 'item-id' HEX ERROR AT LINE NO. n
216	MODE 'item-id' LOADED; FRAME = n SIZE = n CKSUM = n
217	MODE 'item-id' VERIFIED; FRAME = n SIZE = n CKSUM = n
218	MODE 'item-id' FRAME = n HAS N MISMATCHES
219	ILLEGAL COMMAND: command
220	'item-id' EXITED
221	'item-id' FILED
222	'item-id' DELETED
223	'item-id' EXISTS ON FILE
224	VERIFIED
225	CATALOGED PROGRAM HAS n MISMATCHES
226	TAPE FORMAT ERROR
227	LIST 'param' EXCEEDS WORK SPACE SIZE
228	LIST 'param' IS NOT TERMINATED BY A SEGMENT MARK
229	'account-name' IS NOT A VALID ACCOUNT NAME
231	CARD READER MECHANICAL ERROR

SYSTEM ERROR MESSAGES

Message Number	Message																														
234	ITEM SIZE EXCEEDS 32,000 BYTES																														
235	YOU ARE USING AN OBSOLETE OSYM FILE																														
240	A "SELECT" OR "SSELECT" MUST BE USED PRECEDING A 'SAVE-LIST' STATEMENT!																														
241	'item-id' CATALOGED; n FRAMES USED																														
242	'item-id' DELETED																														
260	BATCH-STRING LOCK CODE ELEMENT ERROR AT: n																														
267	PROC XFER TO 'item-id' CANNOT BE COMPLETED																														
268	THE DESTINATION OF THE PROC "GO" STATEMENT: statement, CANNOT BE FOUND																														
270	FORMAT ERROR IN THE PROC STATEMENT : statement																														
272	A VALUE EXISTS FOR THE ATTRIBUTE REFERENCED BY THE ELEMENT : element																														
273	ERROR IN COLUMN-NUMBER/FIELD-WIDTH OR FORMAT SPECIFICATION AT : statement																														
274	UNRECOGNISABLE BATCH-STRING ELEMENT : element																														
275	Y OR F SUB-ELEMENT ERROR AT BATCH-STRING ELEMENT : element																														
276	D-2 UPDATE WITHOUT D-1 BEING SPECIFIED, AT BATCH-STRING ELEMENT : element																														
277	J ELEMENT MISSING AT BATCH-STRING ELEMENT : element																														
278	ERROR IN PROCESSING SECONDARY BATCH-STRING ELEMENT : element																														
279	INCORRECT SCALING FACTOR IN F* BATCH-STRING ELEMENT : element																														
280	FILE-DEFINITION BATCH ELEMENT ERROR AT : element																														
281	D1 MUST HAVE Y11 STORAGE CORRELATIVE ... ERROR AT : element																														
282	DATA INPUT LINE TO BATCH AFTER A SELECT MUST CONTAIN AT LEAST ONE ITEM-ID SUBSTITUTION CODE (ASTERISK *)																														
287	TIMESLICE FOR LINE n EQUALS m MILLISECONDS																														
288	CURSOR CONTROL IN PROC STATEMENT : statement SPECIFIES COL#/ROW# GREATER THAN TERM SETTING.																														
289	<table border="0" style="width: 100%;"> <thead> <tr> <th></th> <th style="text-align: center;">TERMINAL</th> <th style="text-align: center;">PRINTER</th> </tr> </thead> <tbody> <tr> <td>PAGE WIDTH :</td> <td style="text-align: center;">n</td> <td style="text-align: center;">n</td> </tr> <tr> <td>PAGE DEPTH :</td> <td style="text-align: center;">n</td> <td style="text-align: center;">n</td> </tr> <tr> <td>LINE SKIP :</td> <td style="text-align: center;">n</td> <td></td> </tr> <tr> <td>LF DELAY :</td> <td style="text-align: center;">n</td> <td></td> </tr> <tr> <td>FF DELAY :</td> <td style="text-align: center;">n</td> <td></td> </tr> <tr> <td>BACKSPACE :</td> <td style="text-align: center;">n</td> <td></td> </tr> <tr> <td>TERMINAL :</td> <td style="text-align: center;">n</td> <td></td> </tr> <tr> <td>BAUD RATE :</td> <td style="text-align: center;">n</td> <td></td> </tr> <tr> <td>PCI :</td> <td style="text-align: center;">n</td> <td></td> </tr> </tbody> </table>		TERMINAL	PRINTER	PAGE WIDTH :	n	n	PAGE DEPTH :	n	n	LINE SKIP :	n		LF DELAY :	n		FF DELAY :	n		BACKSPACE :	n		TERMINAL :	n		BAUD RATE :	n		PCI :	n	
	TERMINAL	PRINTER																													
PAGE WIDTH :	n	n																													
PAGE DEPTH :	n	n																													
LINE SKIP :	n																														
LF DELAY :	n																														
FF DELAY :	n																														
BACKSPACE :	n																														
TERMINAL :	n																														
BAUD RATE :	n																														
PCI :	n																														
290	THE RANGE OF THE PARAMETER "parameter" IS NOT ACCEPTABLE.																														
291	'file-name' FILE-DEFINITION IS MISSING																														

SYSTEM ERROR MESSAGES

Message Number	Message
293	TOTAL NUMBER OF CONTIGUOUS FRAMES = n NUMBER OF FRAMES IN THE TEMPORARY SPACE BLOCK = n NUMBER OF FRAMES IN THE TEMPORARY SPACE CHAIN = n
294	START OF LINKED OVERFLOW : n TOTAL NUMBER OF LINKED FRAMES : n TOTAL NUMBER OF CONTIGUOUS FRAMES : n
295	OVERFLOW TABLE hh:mm:ss dd mmm yyyy :
298	FORMAT ERROR IN SPECIFICATIONS.
316	WHICH LINE?
318	COUNT TOO HIGH
323	TRY AGAIN. UNABLE TO SET STOP ROADBLOCK
330	ILLEGAL LINE NUMBER.
331	THE ACCOUNT FILE IS MISSING.
332	THE LINK FROM PRIMARY HS WORKSPACE TO ADDITIONAL WORKSPACE WAS INVALID
333	THE FORMAT OF THE ADDITIONAL WORK-SPACE PARAMETER: parameter IS ILLEGAL FOR THIS ACCOUNT NAME. ADDITIONAL WORK-SPACE HAS NOT BEEN ASSIGNED
334	n FRAMES OF ADDITIONAL LOGON WORKSPACE ARE NOT AVAILABLE
335	Logon proprietary message
336	*** LOGGED OFF AT time ON date ***
337	NO MESSAGE SENT.
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	IMPROPER OR UNDEFINED FORMAT FOR DATE CONVERSION
340	***** *** CONNECT TIME AT hh:mm:ss = n MINUTES *** *** CPU MS. = n DISC I/O = n *** *** NUMBER OF ACTIVATIONS = n *** *****
341	A VALUE MUST FOLLOW THE CHARGE-TO VERB.
350	INDENTATION PAST END OF LINE.
398	THE FILE OR DL/ID 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
402	FILE DEFINITION ITEM 'item-id' CANNOT BE DELETED OR OVERWRITTEN
403	ONE ITEM LISTED
404	n ITEMS SELECTED.
405	n ITEMS LISTED.
406	ITEM COUNT = n, BYTE COUNT = n, AVG. BYTES/ITEM = n, AVG. ITEMS/GROUP = n, STD. DEVIATION = n, AVG. BYTES/GROUP = n
407	n ITEMS COUNTED.
408	ONE ITEM COUNTED.
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.
413	THE FILE NAME ALREADY EXISTS IN THE MASTER DICTIONARY.

SYSTEM ERROR MESSAGES

Message Number	Message
414	ILLEGAL OR MISSING MODIFIER USED IN DEFINING THE FILE AREA(S).
415	'item-id' EXISTS ON FILE.
416	RANGE ERROR IN MODULO OR SEPARATION PARAMETER.
417	FILE 'file-name' CREATED; BASE = n, MODULO = n, SEPAR = n
418	FILE-DEFINITION ITEM 'item-id' WAS NOT COPIED.
419	THE SPECIFIED FILE CANNOT BE CLEARED OR DELETED!
420	DICTIONARY FILE DELETION CANNOT BE DONE WITHOUT DELETION OF DATA FIRST.
421	STATISTICS OF attribute-name: TOTAL = n AVERAGE = n COUNT = n
422	BYTE STATISTICS OF attribute-name: TOTAL = n AVERAGE = n ITEMS = n CKSUM = n BITS = n
423	TOTAL OF attribute-name IS : n
424	DISK ADDRESS FOR FRAME NO. n DEVICE ADDRESS : n UNIT NO. : n HEAD NO. : n CYLINDER : n SECTOR : n
425	INVALID FRAME-ID REQUEST
426	DATA FILE ALREADY EXISTS!
427	THERE IS NO DATA SECTION FOR THIS FILE.
469	ILLEGAL FORM QUEUE NAME
489	OPERATOR INTERVENTION REQUIRED
520	NO DATA FOR BLOCK OUTPUT
521	TOO MANY CHARACTERS IN WORD TO BLOCK
522	BLOCK-CONVERT FILE MISSING OR IMPROPERLY DEFINED
523	BLOCK OUTPUT WOULD EXCEED PAGE WIDTH
524	INPUT CHARACTER 'x' IS NOT IN THE BLOCK-CONVERT FILE.
525	INPUT CHARACTER 'x' IS IMPROPERLY FORMATTED IN BLOCK-CONVERT FILE
529	ILLEGAL ATTEMPT TO LOGOFF PHANTOM PROCESS
530	ALREADY LOGGED ON
531	PROCESS ROADBLOCKED
532	ILLEGAL USER ID
533	LOGON SUCCESSFUL
534	LOGOFF SUCCESSFUL
535	ILLEGAL LINE NUMBER.
536	Illegal system privilege parameter, 0-2 only!
550	A REQUIRED NUMERIC PARAMETER IS MISSING OR INVALID
551	SPECIFY NUMBER OF LINES TO SKIP
552	ITEM 'item-id' HAS INVALID FORMAT
599	OBJECT CODE IN 'item-id' IS IN INCORRECT FORMAT.
603	DEFAULT ASSIGNMENT IS TO QUEUE "STANDARD"
604	'x' IS AN UNDEFINED LABEL REFERENCE.
605	FORM QUEUE = name DEVICE TYPE = type DEVICE NUMBER = n SPOOLER OPTIONS = options COPIES = n
607	UNACCEPTABLE VALUE
609	THE PAGE YOU REQUESTED IS NOT AVAILABLE
610	LINE# 'n' IS ASSIGNED TO THIS QUEUE. FORM QUEUE CANNOT BE DELETED!

Message Number	Message
611	TYPE Y (YES) OR N (NO):
612	THE DEFAULT FORM QUEUE MAY NOT BE DELETED
613	THE FORM QUEUE IS NOT EMPTY
614	NOT A PORT PROCESS
615	DEVICE TYPE:
616	CANNOT MOVE A FORMQ TO ITSELF
617	DEVICE DOES NOT EXIST
618	N OPTION DISREGARDED
619	LINE# --OR-- LINE# FORM-QUEUE:
620	LPTR ATTACHED!
621	CHECK SPOOLER!
622	SPOOLER ABORTED. RESTART IS BEING ATTEMPTED.
623	ILLEGAL ACTION CODE
624	ENTER ACTION CODE / PAGE NUMBER (P#):
625	FORM-QUEUE PAGES:
626	NO OVERFLOW FRAMES
627	ENTER ACTION CODE:
628	PRINT-JOB# OPTIONS:
629	FORM-QUEUE FORMTYPE:
630	'STOP' IN EFFECT. USE 'RESUME'!
631	OLD-NAME NEW-NAME:
632	PRINT JOB # COPIES:
633	PRINTFILE #:
634	FORM-QUEUE:
635	FORM-QUEUE OR PRINT JOB #:
636	FORM-QUEUE --OR-- DEVICE# TYPE:
637	FROM-FORM-QUEUE TO-FORM-QUEUE:
638	NEW-FORM-QUEUE PRINT JOB LIST:
639	PRINT JOB LIST:
640	FORM-QUEUE DEVICE-TYPE(LPTR,TAPE,PORT) DEVICE#:
641	THERE ARE n HOLD FILES
642	EJECT AND SKIP VALID ONLY FOR PRINTERS
643	THE FORM QUEUE TABLE IS FULL
644	PRINTING SUSPENDED ON JOB #n
645	THERE IS NO ACTIVE JOB TO RESTART
646	***** TOP OF PAGE *****
647	THE DEVICE IS ALREADY ASSIGNED AND CANNOT BE CHANGED.
648	DEVICE TYPE DEVICE#:
649	NEW-NAME:
650	FORMTYPE:
651	CANNOT ALIGN - CANNOT ATTACH DEVICE
652	TOO MANY PAGES!
653	OPTIONS:
654	ILLEGAL OUTPUT STRING LENGTH!
655	NOT ATTACHED!
656	JOB ALREADY BEING EDITED BY LINE #n
657	FILE NOT CLOSED
658	AN INSTANT PRINT JOB WITHOUT AN H OPTION CANNOT BE EDITED
659	MUST SUSPEND PRINTING TO EDIT THIS JOB
660	INSUFFICIENT PRIVILEGES OR DEVICE ATTACHED TO SPOOLER
661	TAPEOUT TERMINATED BECAUSE OF ASSIGN TAPE.

SYSTEM ERROR MESSAGES

Message
Number

Message

662 PRINT JOBS PAGE n OF n PAGES

PRINT	DATE
QUEUE NAME	JOB ACCOUNT LINE CREATED STATUS OPT SIZE COPIES
663	1. MOVE FORM QUEUE 5. DELETE JOB 9. SUSPEND PRINTING
	2. MOVE PRINT JOB 6. STOP PRINTING 10. CHANGE PRIORITY
	3. CHANGE OPTIONS 7. RESUME PRINTING 11. SP-STATUS
	4. CHANGE #OF COPIES 8. EDIT PRINT JOB 99. EXIT

664 PAGE x OF y PAGES

665 THE QUEUE IS EMPTY

666 ILLEGAL NUMBER: n

667 WRONG PASSWORD FOR PRINT JOB # n

668 JOB NOT QUEUED FOR OUTPUT

669 PRINT JOB # n DOES NOT EXIST

671 PRINT JOB # n ALREADY PRINTING

672 COPY RANGE IS 1-100

673 INVALID OPTION: option

674 THE DEVICE YOU WANTED IS CURRENTLY ASSIGNED TO AN ACTIVE FORM QUEUE. PLEASE SUSPEND THAT FORM QUEUE AND THEN CHANGE THE DEVICE ASSIGNMENT.

675 FORM QUEUE ASSIGNMENT PAGE n OF m PAGES

QUEUE NAME	FORM TYPE	DEVICE ASSIGNED	DEVICE TYPE	DEVICE STATUS	NUMBER ENTRIES	PAGE SKIP
676	PRINT JOB SELECTED IS NOT A HOLD FILE					
677	STRINGS?					
678	OCCURRENCE MUST BE NUMERIC					
680	SYNTAX ERROR					
681	INVALID SPOOL TAPE FORMAT					
682	THE FORM-QUEUE CANNOT BE FOUND					
683	PRINT JOB CANNOT BE OPENED ACCOUNT NAME AND PASSWORD TOO LONG					
685	PRINT JOB TABLE FULL					
686	PRINT JOB FULL					
687	A DEVICE HAS NOT BEEN ASSIGNED TO THIS QUEUE					
688	INCOMPATIBLE OPTIONS: NI, NH, N COPIES, I COPIES, OR N PORT					
689	ACCOUNT NAME CANNOT BE FOUND					
690	QUEUE IS ALREADY ASSIGNED TO THIS DEVICE. OK TO CHANGE?					
691	THE LIST OF PORTS WITH NO DEVICES ASSIGNED IS FULL					
692	INVALID DEVICE TYPE					
693	INVALID DEVICE# RANGE					
694	FORM QUEUE ALREADY EXISTS					
695	FORM QUEUE ACTIVE. ITS ASSIGNMENT CANNOT BE CHANGED					
696	NAME TOO LONG					
697	PRINT JOB # n REQUIRES A PASSWORD					
698	PRINT JOB # n IS NOT CLOSED					
699	ACTION CODES					
	1. CREATE FORM QUEUE	4. CHANGE DEVICE ASSIGNMENT	7. DELETE FORM QUEUE			
	2. CHANGE FORM QUEUE	5. CHANGE PAGE SKIP	8. PAGE EJECT			
	3. CHANGE FORM TYPE	6. LIST PRINT JOBS	99. EXIT			
700	INVALID G-CORRELATIVE DEFINITION : definition					

Message Number	Message
701	INVALID FUNCTION CORRELATIVE DEFINITION : function-definition
703	INVALID C-CONVERSION/CORRELATIVE DEFINITION : definition
704	INVALID P-CONVERSION DEFINITION : definition
705	ILLEGAL CONVERSION CODE : conversion-code
706	THE TRANSLATE CONVERSION CODE : 'code' IS ILLEGAL.
707	DL/ID ENTRY FOR T-CONVERSION : conversion-code NOT FOUND.
708	'item-id' CANNOT BE CONVERTED.
709	CONFLICT IN T-CONV DEFINITION; REQUESTS VERIFY BUT THERE IS A NULL AMC
710	INVALID L- OR R- CONVERSION DEFINITION : definition
712	VALUE TEST IMPOSSIBLE ON ATTRIBUTE WITH F-CONVERSION: definition
715	NON-NUMERIC DIGIT WITHIN PARENTHESIS : 'n'
716	MISSING LEFT PARENTHESIS : 'n'
717	'P' IS MISSING IN PICTURE CLAUSE : 'n'
718	ILLEGAL CHARACTER FOUND ON PICTURE CLAUSE : 'n'
719	LITERAL EXCEEDS MAX. OF 20: 'n'
720	MISSING RIGHT PARENTHESIS: 'n'
721	MISSING ' IN CURRENCY SYMBOL: 'n'
722	MISSING " IN LITERAL SYMBOL: 'n'
723	MAX. LENGTH DEFINED IN DICTIONARY IS TOO SMALL FOR MF CONVERSION
736	NO SCALING WITH NON-NUMERIC DATA: PAGE DATA PAGE DATA PAGE DATA
737	TRUNCATION ERROR FOUND IN STORED DATA: PAGE DATA PAGE DATA PAGE, DATA
738	NON-ALPHABETIC CHARACTER FOUND: PAGE DATA PAGE DATA PAGE DATA
739	NON-NUMERIC CHARACTER FOUND: PAGE DATA PAGE DATA PAGE DATA
740	SPECIAL CHARACTER FOUND IN DATA: PAGE DATA PAGE DATA PAGE DATA
741	INSUFFICIENT DATA TO MATCH PICTURE CLAUSE: PAGE DATA PAGE DATA PAGE, DATA
743	ILLEGAL OPERATION FOR THIS DRIVE. COMMAND IGNORED
744	ILLEGAL DEVICE TYPE SPECIFIED IN UNIT LIST CANNOT MIX 1/4" AND 1/2" DRIVES
745	NUMBER OF CORRECTED ERRORS: n
746	TAPE ABORTED
747	READ ATTEMPTED AFTER WRITE. COMMAND IGNORED
748	TAPE POSITION ERROR ON WRITE. COMMAND IGNORED
749	NO DATA DETECTED. COMMAND IGNORED
780	'item-id' NOT ON FILE
781	'item-id' ADDED
782	'item-id' UPDATED
783	'item-id' DELETED
800	<n> ITEMS DUMPED
801	:
802	n ITEMS DUMPED.
803	n ITEMS LOADED
805	n ITEMS COPIED
860	CHECK CODE ERROR
888	INVALID DATE
890	TERMINAL LOAD INHIBITED

SYSTEM ERROR MESSAGES

Message Number	Message
900	MUST USE A TCL COMMAND BEFORE AN "IF E" COMMAND
901	RECEPTION ABORTED
902	TRANSMISSION ABORTED
903	ILLEGAL HEADER ENCOUNTERED 'header'
904	WAIT-TIME MATURED
905	n ITEMS TRANSMITTED AS m RECORDS
906	n ITEMS RECEIVED IN m RECORDS
907	PROTOCOL ERROR IN ITEM n RECORD m
908	n PRINT JOBS SPOOLED - TOTAL OF m RECORDS
909	FATAL PROTOCOL ERROR
910	*T* PROCESS STOP.
911	*T* PROCESS HAS BEEN LOGGED OFF BY PH-KILL
912	*T* PROCESS ABORT DUE TO TIPH PROCESSING ERROR
913	*E* INPUT ERROR
914	*E* PROCESS IN DEBUGGER.
915	ERROR FOUND IN SUSPENDING LINE
916	ERROR FOUND IN RESUMING LINE
917	ERROR FOUND IN KILLING LINE
918	WRONG-INPUT
919	SYS2 ERROR
920	LINE # MISSING
949	ILLEGAL FRAME-ID IN PRIMARY SPACE! PROGRAM TERMINATED.
950	LINKFIELD ERROR - GROUP AT n FRAME n LINKS n n n n
951	ILLEGAL FORWARD LINK. GROUP AT n FRAME n LINKS n n n n
952	ILLEGAL BACKWARD LINK. GROUP AT n FRAME n LINKS n n n n
953	SHOULD LINK BACK TO: GROUP AT n FRAME n LINKS n n n n
954	n NEW ERROR ITEMS CREATED IN DICT TSYM
970	CURRENT ITEM LOCK TABLE SIZE IS n DEFAULT ITEM LOCK TABLE SIZE IS m
971	ITEM LOCK TABLE MUST BE EMPTY TO CHANGE THE SIZE.
972	LI# LEVEL LOCK.... LOCK.... LOCK... LOCK... LOCK... LOCK...LOCK...
973	n LOCKS LISTED
975	ITEM LOCKED BY LINE n AT LEVEL m
976	'lock no.' UNLOCKED
977	SIZE MUST BE IN THE RANGE 1 TO 4826
987	x x x x x x
990	INVALID ABSDUMP SPECIFICATION
991	MM/DICT ENTRY MISSING, OR REQUIRED SPECIFICATION(S) MISSING.
992	n ITEM(S) HAVE BEEN RESTORED
999	INSUFFICIENT CONTIGUOUS BLOCKS
1004	ITEM 'item-id' IS NOT ON FILE.
1006	ITEM 'item-id' EXISTS ON FILE
1010	THE PARAMETERS BEING CHANGED ARE NOT COMPATIBLE WITH THE PORT OPTION.
1011	BAUD RATE AND PCI TRANSMISSION CHARACTERISTICS CANNOT BE MODIFIED WITH THE SET-TERM VERB. USE "TERM" INSTEAD.
1014	ENFORCED WRITE FREQUENCY IS 1 WRITE PER n ACTIVATIONS

Message Number	Message
1021	DEVICE(S) TAPE unit list NOW ASSIGNED, RECORD SIZE = n
1023	SORT TERMINATED DUE TO ATTEMPT TO SORT VALUES > X'E8E8E8'
1025	DEVICE TAPE n IS ASSIGNED TO LINE m
1026	TAPE MARK(S) MISSING AFTER HEADER/TRAILER LABELS n
1027	RECORD LENGTH ERROR n
1028	BLOCK COUNT ERROR n
1029	BLOCK SEQUENCE INDICATOR ERROR n
1100	x IS THE SYSTEM SERIAL NUMBER xxx IS THE PROM MICROCODE VERSION NUMBER x IS THE WRITABLE CONTROL STORE MICROCODE VERSION NUMBER xxxxK OF MEMORY xxxx ABS FRAMES xx ACTIVE LINES xx IS THE LINE TO WHICH THE SPOOLER IS ASSIGNED xxxx IS THE FRAME ID OF THE PCB FOR LINE ZERO xxxx IS THE FRAME ID OF THE BASE OF THE SYSTEM DICTIONARY xxxxxx IS THE MAXIMUM FRAME ID FOR THIS SYSTEM
	DISC DRIVE CONFIGURATION: DRIVE CONTROLLER CHANNEL
1101	DISK ADDRESS FOR FRAME NO.: xxxx DRIVE :xx CONTROLLER :xx CHANNEL :xx CYLINDER :xx HEAD NO. :xx SECTOR :xx
1102	ERROR IN WRITE INSTRUCTION TEST NUMBER n
1103	DEVICES READS WRITES TOTAL I/O
1104	I / O P E R S E C O N D
1111	MISSING OR ILLEGAL CHANNEL NAME.
1112	ILLEGAL ASSIGN STATEMENT FORMAT.
1113	ILLEGAL UNIT SPECIFICATION
1114	MISSING OR ILLEGAL RECORD SIZE
1115	EXTRANEIOUS INPUT AT END OF STRING
1116	DUPLICATE ENTRIES IN UNITLIST
1119	DEVICE TAPE 'unit #' ASSIGNED TO THIS LINE, BUT FOR ANOTHER CHANNEL.
1120	DEVICE(S) TAPE unit list NOW ASSIGNED TO CHANNEL 'channel name', RECORD SIZE = size.
1121	ILLEGAL CHANNEL SPECIFICATION.
1122	TAPE UNIT NUMBER MUST BE SPECIFIED
1123	ILLEGAL RECORD SIZE
1124	DEVICE TAPE 'unit #' NOT ON THIS SYSTEM
1125	UNIT NOT ASSIGNED.
1128	DEVICE(S) 'TAPE n' NOW ASSIGNED
1129	NO SUCH TAPE ASSIGNMENT EXISTS.
1200	LABEL 'VOL1' FORMAT ERROR n
1201	LABEL 'HDR1' FORMAT ERROR n
1202	LABEL 'HDR2' FORMAT ERROR n
1203	LABEL 'EOF1' FORMAT ERROR n

SYSTEM ERROR MESSAGES

Message Number	Message
1204	LABEL 'EOF2' FORMAT ERROR n
1205	BLOCK LENGTH ERROR n
1206	TAPE MARK(S) MISSING AFTER HEADER/TRAILER LABELS n
1207	RECORD LENGTH ERROR n
1208	BLOCK COUNT ERROR n
1209	BLOCK SEQUENCE INDICATOR ERROR n
1210	SELECTED FILE WAS SAVED AS A DUPLICATE 'D' POINTER PLEASE USE THE INITIAL FILE NAME ADOPTED BY THE SAVE PROCESS (REFER TO THE FILE- STATS ASSOCIATED WITH THIS TAPE)
1240	MODE 'mode' NOT USED ON THIS SYSTEM
1241	UNABLE TO OPEN SYSTEM-LOG FILE
2001	n MISMATCHES IN INTERMEDIATE OBJECT CODE.
2002	ABORT. RE-EXECUTE TCL-II VERB 'STOREPF'
2321	ITEM 'item-id' ERROR
9009	PQ-SELECT REGISTER OUT OF RANGE 1 - 5
9012	UPDATE ATTEMPTED ON AN INPUT FILE AT STATEMENT : statement
9020	COMPILATION ABORTED.
9021	PROC FORMAT ERROR IN LABEL
9022	DUPLICATE LABEL FOUND.
9025	FORMAT ERROR IN 'GO' STATEMENT
30000	n TAPES DO NOT VERIFY
30001	TAPES VERIFY
LOGON	L

APPENDIX C

=====

STAT-FILE DICTIONARY

<u>A/AMC</u>	<u>Name</u>	<u>Definition</u>
0	ORDER.	Sequence in which file was encountered during the file-save.
1	ACCOUNTNAME.	First segment of Attribute 1 (G*1). Specifies the account the file resides in.
1	FILENAME.	Second segment of Attribute 1 (G1*1). The name of the file.
1	FILE.	Second and third segments of Attribute 1 (G1*2). If the third segment of Attribute 1 is present, this specifies the data section of the file named in second segment.
2	LEVEL.	System level of the specified file 0 = System Dictionary 1 = Master Dictionary 2 = Data Dictionary 3 = Data File
3	BASE.	Beginning frame number of the file
4	MOD.	Modulo of the file
5	SEP.	Separation (number of frames per group) of the file
6	ITEMS.	Total number of items in the file
7	SIZE.	Size of the file in bytes (characters)
8	MAX-I	Size (in bytes) of the largest item in the file
9	MIN-I	Size (in bytes) of the smallest item in the file
10	MX/GP	Maximum number of items found in any group in the file
11	MN/GP	Minimum number of items found in any group in the file
12	FRAMES.	Total frames, including linked, used by the file
13	25%	Number of groups that are 25% full or less

STAT-FILE DICTIONARY

<u>A/AMC</u>	<u>Name</u>	<u>Definition</u>
14	50%	Number of groups that are 25% to 50% full
15	75%	Number of groups that are 50% to 75% full
16	100%	Number of groups that are 75% to 100% full
17	125%	Primary space of groups in this category is full and 25% additional linked space is required
18	150%	Primary space of groups in this category is full and 50% additional linked space is required
19	200%	Primary space of groups in this category is full and additional linked space equals the primary space
20	>200%	Primary space of groups in this category is full and additional linked space is greater than the primary space.
21	GFE	Group format errors, if any, in the file
99	SECONDARY.	Computes the number of linked frames required for the file by multiplying modulo times separation and subtracting the product from the total frames (F;5;4;*;12;-)
99	AVG.SIZE	Computes the average item size by dividing the size by the total items in the file (F;6;7;/)
	DL/ID	DL/ID of STAT-FILE. Note that Attribute 1 is DY, not D.

APPENDIX D

=====

CREATING THE PH-HISTORY FILE

The Terminal Independent Process Handler (TIPH) uses the PH-HISTORY file to indicate the status of the TIPH processes. If the file does not exist, you may create it by using the CREATE-FILE verb, adding the necessary attribute definition items using the EDITOR.

To create the PH-HISTORY file, perform the following:

1. Create the file on the SYSPROG account using the CREATE-FILE verb. The modulo and separation of the dictionary section should both be "1". The modulo of the data section should be "37", the separation should be "1".
2. Use the EDITOR to enter the attribute values shown in Figure A to create the attribute definition items in the dictionary portion of the PH-HISTORY file.
3. Use the EDITOR to change the file's D-pointer in SYSPROG's M/DICT to an 'E'.
4. Copy the E-pointer to the SYSTEM file.
5. Use the EDITOR to change the E-pointer back into a D-pointer.
6. Place a Q-pointer to PH-HISTORY in the M/DICT of each account which requires access to it.

Figure B shows an example of the creation procedure.

PH-HISTORY FILE

Attribute	ACCOUNT	DATE	LINE	OP	OUTFL#	STARTER- ACC	STATUS	TASK- ID	TIME
1. D/CODE	A	A	A	A	A	A	A	A	A
2. A/AMC	1	2	6	7	8	4	9	5	3
3.	*	*	*	*	*	*	*	*	*
4.	*	*	*	*	*	*	*	*	*
5.	*	*	*	*	*	*	*	*	*
6.	*	*	*	*	*	*	*	*	*
7.	*	D	*	*	*	*	*	*	*
8.	*	*	*	*	G 1	*	*	*	MTS
9. V/TYPE	L	L	L	R	R	L	R	L	L
10. V/MAX	10	12	2	2	3	11	6	10	8
* null									

Figure A: Attribute Definition Item Information for PH-HISTORY

- ```

1. :CREATE-FILE (PH-HISTORY 1,1 37,1) <cr>

[417] FILE 'PH-HISTORY' CREATED; BASE = xxxxxx, MODULO = 1, SEPAR = 1
[417] FILE 'DL/ID' CREATED; BASE = xxxxxx, MODULO = 37, SEPAR = 1

2. :ED DICT PH-HISTORY ACCOUNT <cr>
NEW ITEM
TOP
.I <cr>
001 A <cr>
etc.
.
.
'ACCOUNT' FILED.

Use the EDITOR to enter the other attribute definition items in a
similar fashion.

3. :ED MD PH-HISTORY <cr>
TOP
.R/D/E <cr>
001 E
.F1 <cr>
'PH-HISTORY' FILED.

4. :COPY MD PH-HISTORY (D) <cr>
TO:(SYSTEM) <cr>
1 ITEM COPIED.

5. :ED SYSTEM PH-HISTORY <cr>
TOP
.R/E/D <cr>
001 D
.FI <cr>
'PH-HISTORY' FILED.

6. :ED MD PH-HISTORY <cr> <----- (while logged onto the user's account)
NEW ITEM
TOP
.I <cr>
001 Q <cr>
etc.
.
.
'PH-HISTORY' FILED.

```

The format of the Q-pointer to the PH-HISTORY file is shown in Figure B of the topic TIPH INITIALISATION MESSAGES.

Figure B: Creating the PH-HISTORY File



## APPENDIX E

\*\*\*\*\*

SAFEGUARDING SAVE TAPES

Tape errors may result from the effects of numerous sources. Tape transport and controller hardware defects and misalignments are the cause of many tape errors. A reliable save should not be expected from a faulty medium. On the tape transport, the tension bars must be properly adjusted to prevent tape cinching and edge damage. Head misalignment, signal level improperly set, and excessive wear from overuse of the same tape can all contribute to unrecoverable tape errors.

However, the more common source of tape errors is not mechanically induced by faulty equipment; instead, it is the result of careless handling of the tape during its use or its improper care while in storage. If the data to be saved is properly written to tape, the data cannot be expected to remain intact if the tape is left: 1) lying around the computer room, out of its case, near electromagnetic field and dust-generating equipment 2) around eating and smoking people, 3) underneath stacks of listings, or 4) next to an open window on a windy, hot day.

The tape should be stored inside its case in a vertical position, in an environment whose temperature is within 5 degrees C of 24 degrees C, whose humidity is within 10% of 10 RH, and will be exposed to an electromagnetic field of less than 50 oersteds.

When the tape is used during a save or restore, care should be taken to ensure that the tape reel is handled in a manner which will not cause it to get dirty. The tape's case should be closed when the reel is not inside so that the tape is not exposed to dust. The reel should be held with the thumb inside the hub, not by the flanges, to prevent excessive pressure on the sensitive edges of the tape.

It is necessary to emphasize that correct handling and storage of the magnetic tape is an integral part of any Save/Restore program.

For more information on care and maintenance of magnetic tapes, refer to the following manuals:

- 1) Computer Tape Care and Handling, Memorex Corp., 1979.
- 2) Magnetic Tape Management, Computer-Link Corp., 1981.



## INDEX

:  
:ABS/FILES verb, 10-18  
:FILES verb, 10-18  
:IL-NEWTAB verb, 10-66  
:INIT-SYSTEM verb, 10-58  
:RESTART-SPOOLER verb, 7-70  
:SP-NEWTAB verb, 7-70

A

A/AMC attribute, 3-8  
aborts, 4-12  
ABS area, 2-2  
ACC(Accounting History) File,  
3-14,5-24etc.  
account  
create, 10-36  
definition item, 5-14  
delete, 10-38  
restore, 10-12,10-14  
save, 10-8,10-10  
synonym, 5-14  
ACCOUNT(ACC) File, 5-24etc.  
ACCOUNT-FILE-STATS PROC, 10-51  
ACCOUNT-RESTORE verb, 10-12  
ACCOUNT-SAVE PROC, 10-8  
ADDD verb, 9-2,9-3  
ADDX verb, 9-2,9-3  
ALL-FILE-STATS PROC, 10-50  
arithmetic verbs, 9-2  
assembler utilities, F-1etc.  
assembly language routines  
verification from tape, 10-68  
ASSIGN verb, 8-2  
attribute, 2-10  
definition items, 3-8etc.  
mark, 2-12  
mark count (AMC), 3-8  
auto disconnect, 9-10  
autoprint, H-1etc.  
prog. info (9747), H-16etc.  
programming information, H-11etc.  
available space, 2-6etc.  
displaying, 9-56

B

backspace, 9-12

base, 2-10  
baud rate, 9-11  
bit, 2-10  
BKOFF verb, 9-30  
BKON verb, 9-30  
BLOCK-CONVERT file, 3-14,9-32  
BLOCK-PRINT verb, 9-32  
BLOCK-TERM verb, 9-32  
BOOTDUMP verb, 10-34  
BREAK function  
enabling/disabling, 9-30  
BREAK-KEY-OFF verb, 9-30  
BREAK-KEY-ON verb, 9-30  
BUFFERS verb, 10-54  
byte, 2-10

## C

CALCFID verb, 10-48  
cartridge tape unit, 8-8  
CHANNEL file, 5-25  
character  
ASCII set, A-1etc.  
CHARGE-TO verb, 5-8  
CHARGES verb, 5-8  
CHECK PROC, 9-73  
CHECK-SUM verb, 9-58  
CLAIM verb, 2-8,10-42  
CLEAR-ASSIGN verb, 8-2  
CLEAR-BASIC-LOCKS verb, 10-58  
CLEAR-FILE verb, 6-8  
CLEAR-ITEM-LOCKS verb, 10-64  
CLEAR-MSG verb, 9-44  
CLINK verb, 10-42  
COLDDUMP verb, 10-32  
COMP verb, 9-64  
COMPARE verb, 9-72  
configuration(display of), 9-45,  
9-50  
connectives(list of), 9-71  
control characters, 4-4  
CTRL-X, 9-14  
conventions of documentation, 1-7  
conversion utilities, F-21etc.  
<> --> (), F-26  
Dec./Hex.-->Binary, F-24  
Decimal <--> Hex., F-24  
Full Stop <-> Comma, F-22  
Full Stop/Comma/Space, F-22

conversion utilities cont.  
 Reduce Repeated Chars. To One,  
 F-27  
 Replace Character, F-26  
 Strip All Except 0-9 And A-Z,  
 F-25  
 Strip Character, F-25  
 Strip Spaces, F-25  
 COPY verb, 6-10etc.  
 options, 6-14  
 count field (in item), 2-12  
 CP PROC, 9-70  
 CREATE-ACCOUNT PROC, 10-36  
 CREATE-FILE verb, 6-4  
 CT PROC, 9-70  
 CTS flow control, 9-10

D

D-pointers, 3-4  
 D/CODE attribute, 3-4,3-6,3-8  
 data file, 1-10  
 data structures, 2-1etc.  
 DATA/BASIC  
 program development aid, G-3etc.  
 DATA/BASIC utilities, F-15etc.  
 Check 'Select' Etc., F-16  
 Check Cataloged, F-15  
 Concatenate Character, F-16  
 Get Number Of Items Selected,  
 F-16  
 Get Process Number Etc., F-15  
 Get Run-Time Options, F-15  
 Get System Serial Number, F-15  
 Get TCL Input, F-15  
 Get Terminal Page Width, F-16  
 Honeywell utilities, F-55  
 ICL read and write 1900, F-40etc.  
 ICL utilities, F-51etc.  
 Input And Reduce Characters, F-19  
 Leading Zeros, F-17  
 Overpunched To/From Numeric, F-18  
 Set Process to Sleep, F-15  
 Set Terminal Echo, F-16  
 Toggle Warning Message Print,  
 F-16  
 Trailing Spaces, F-17  
 DATE-FORMAT verb, 9-46  
 DB PROC, G-3  
 debug facility, 4-12  
 selected commands, 4-13  
 DELETE PROC, 9-66  
 DELETE-ACCOUNT PROC, 10-38

DELETE-FILE verb, 6-8  
 device status, 7-13  
 DEVICES file, 3-14,9-4  
 dial-up line, 9-10  
 dictionary, 3-1etc.  
 attribute definition items, 3-8  
 file definition items, 3-4  
 hierarchy, 3-3  
 master, 1-10,3-10  
 summary of def. items, 3-10etc.  
 system, 1-10,3-10  
 types of item, 3-3  
 DISCIO verb, 9-50  
 disk  
 activity report, 9-50  
 error recording, 10-44  
 mapping, 10-48  
 DIVD verb, 9-2,9-3  
 DIVX verb, 9-2,9-3  
 DL/ID item, 3-5,6-4  
 DSKAD verb, 10-48  
 DSR/DCD monitoring, 9-10,9-11  
 DTX verb, 9-2,9-3  
 DUMP verb, 9-38  
 DUMP-FILES PROC, G-6  
 dumping  
 dict./data files, 9-36  
 frames, 9-38  
 to tape, 8-12

E

EBASIC verb, 9-74  
 ECOPEY verb, 9-74  
 edit print job commands, 7-48etc.  
 alignment, 7-54  
 backward positioning, 7-52  
 exit, 7-54  
 forward positioning, 7-52  
 list, 7-50  
 locate, 7-50  
 print, 7-54  
 top of print job, 7-52  
 EEDIT (EED) verb, 9-74  
 ENGLISH utilities, F-14etc.  
 Buffer Printer Output, F-6  
 Running Total of Values, F-14  
 Running Totals of Attrib. etc,  
 F-14  
 erase, 9-14  
 ERRMSG file, 3-12,B-1etc.  
 displaying messages in, 9-34  
 item LOGON, 5-6

error messages, B-1etc.  
error recording, 10-44etc.  
ESEARCH verb, 9-62  
EVFU-SETUP program, G-2  
execution locks, 4-12,10-58

## F

F/BASE F/MOD F/SEP attributes, 3-4  
F/REALLOC attribute, 3-5  
FEATURES file, 3-14  
FF delay, 9-12  
file, 2-10  
  clearing, 6-8  
  comparison, 9-64,9-72  
  creation, 6-4,6-14  
  definition items, 3-4,3-10  
  deletion, 6-8  
  FID information, 9-40  
  hashing statistics, 9-42  
  management processors, 6-1etc.  
  organization, 1-10  
  reallocation, 3-5  
  restoring, 10-18  
  saving, 10-22  
  searching, 9-62  
  single-level, 3-2,3-12  
  statistics report, 10-50  
  structure, 2-10etc.  
  synonym definition items, 3-6  
  update, 2-18  
FILE-SAVE PROC, 10-22  
file-save tape format, 10-24  
files and dictionary files,  
  3-1etc.  
FIX-FILE-ERRORS verb, 9-60  
form queue, 7-2  
  creation of, 7-16  
  STANDARD, 7-2  
formatted file-save, 10-22  
  operation of, 10-28  
four column output, G-9

## G

group, 2-10,2-16etc.,6-6  
  format errors, 6-9,9-60,10-32  
  locks, 4-12,10-33,10-58  
GROUP verb, 9-40

## H

HASH-TEST verb, 9-42  
hashing algorithm, 2-16  
high disk space (file area), 2-4  
hold files, 7-4,7-7  
Honeywell utilities, F-55  
HUSH verb, 4-4

## I

I-DUMP verb, 9-36  
ICL utilities  
  miscellaneous, F-51  
  read and write 1900, F-40  
  READ/WRITE-1900, F-33  
  READ/WRITE-2900, F-45  
initial system files, 3-12  
initialising the system, 10-20  
input statements, 4-4  
intraprocessor linkages, 4-3  
ISTAT verb, 9-42  
item, 2-10  
  deletion, 9-66  
  locks, 10-58,10-60etc.  
  storage, 2-16  
  structure (logical), 2-14  
  structure (physical), 2-12  
ITEM verb, 9-40  
item-id, 2-10

## L

L/RET L/UPD attributes, 3-4,5-22  
LF delay, 9-12  
LIMITS verb, 9-50  
line 25 use, 9-13  
line characteristics, 9-8etc.  
line skip, 9-12  
LIST-STAT-FILE\* PROC, 10-50  
LIST-DISC-ERRORS PROC, 10-44  
LIST-FILE-STATS PROC, 10-50  
LIST-ITEM-LOCKS verb, 10-62  
LIST-LOCKS verb, 10-58  
LIST-OVF-ERRORS PROC, 10-46  
LIST-PLAN-ERRORS PROC, 10-44  
LIST-PORTS PROC, 9-20  
LIST-SYS-ERRORS PROC, 10-46  
LIST4 PROC, G-9  
LISTACC PROC, 9-70  
LISTCONN PROC, 9-71  
LISTDICTS PROC, 9-71



LISTF4 PROC, G-9  
 LISTFILES PROC, 9-71  
 LISTPF PROC, 9-71  
 LISTPROCS PROC, 9-71  
 LISTU PROC, 9-72  
 LISTVERBS PROC, 9-72  
 LOAD-FILES PROC, G-6  
 location of DTE, 9-10  
 LOCK-FRAME verb, 10-56  
 locks  
   display/clear grp. & exec., 10-58  
   execution, 4-12  
   group, 4-12,10-33  
   item, 10-58,10-60etc.  
   retrieval/update, 5-22  
 LOGOFF verb, 11-2  
 LOGON item  
   in ERRMSG, 5-6  
 LOGON PROC, 5-6  
 logon work space table (LWST),  
   5-18  
 logon/logoff, 5-1etc.  
   general system message, 5-6  
 LOGTO verb, 5-10  
 low disk space, 2-4  
 LP-type PROCs, 9-72  
 LPTR line skip, 9-14  
 LPTR page width/depth, 9-14

M

M-A-R PROC, 10-14  
 M-A-S PROC, 10-10  
 M/DICT (Master Dictionary), 1-10,  
   3-1etc.  
   file definition items, 3-4,3-6  
   items, 3-10  
   verb/PROC def. items, 4-6  
 magnetic tape units, 8-1etc.  
   cartridge (1/4 in) unit, 8-8  
   control verbs, 8-6etc.  
   dump/load files (PROCs), G-6  
   input/output, 8-10etc.  
   multiple reel operations, 8-22  
   safeguarding tapes, E-1  
   sequencing, 8-3  
   tape labels, 8-18etc.  
 memory buffer status, 10-54  
 MESSAGE verb, 9-44  
 messages, 9-14  
 Model 12080 Terminal, H-1  
 Model 6749 Terminal, H-13  
 Model 6762 Terminal, H-13

modulo, 2-10  
   selection of, 6-6  
 MSG verb, 9-44  
 MT-LOAD verb, 8-16  
 MT-VERIFY verb, 8-16  
 MUL verb, 9-2,9-3  
 multiple account  
   restore, 10-14  
   save, 10-10  
 multivalued, 2-10,2-12  
 MULX verb, 9-3

N

NEWAC file, 3-12  
 NOHUSH verb, 4-4

O

OFF verb, 5-1,5-20  
 ON/XOFF flow control, 9-10  
 overpunched numeric fields,  
   F-17etc.  
 U01A7, F-18  
 U11A7, F-19  
 U21A7, F-19

P

P-ATT verb, 7-66  
 P-DET verb, 7-66  
 page width/depth, 9-12  
 parity errors, 7-12,8-7,9-73  
 PCI setting, 9-10,9-11  
 PCM, 9-4  
 PD PROC, G-9  
 PH-ALLOCATE verb, 11-4  
 PH-DELETE verb, 11-4  
 PH-HISTORY file, 3-14,11-2,11-12  
   attribute definition, D-2  
   creating the, D-1  
 PH-KILL verb, 11-10  
 PH-LINES verb, 11-4  
 PH-RESUME verb, 11-10  
 PH-START verb, 11-6  
   error messages, 11-14  
   options, 11-8  
 PH-STATUS verb, 11-12  
 PH-SUSPEND verb, 11-10  
 phantom processes, 11-1etc.

PPlan  
   driving printers on, 7-64  
   error recording, 10-44  
   planned shutdown, 10-30  
 POINTER-FILE file, 3-5,3-14,10-26  
   report, 9-71  
 port assignment, 11-4  
 port configuration maintenance,  
   9-4  
 PORT-SETUP program, 9-20  
 PORTOUT verb, 7-58  
 POVf verb, 2-8,9-56  
 PP terminal executive, H-13  
 primary work space, 2-2  
 Print Dictionary Items, G-9  
 print format parameters, G-2  
 print job, 7-2  
 print screen  
   prog. info (9747), H-15etc.  
 PRINT-ERR verb, 9-34  
 printer control, 7-66  
 PRINTRONIX program, 9-68,G-2  
 privilege levels, 5-14  
   displaying, 9-50  
   setting, 9-54  
 PROC  
   list of PROCs, 9-71  
   trace facility, G-9  
 PROC utilities, F-2etc.  
   Check Tape Attached, F-8  
   conventions and definitions, F-2  
   Enable/Disable Break Key, F-13  
   Error Handling Routine, F-11  
   Get Line No./Acnt. Name, F-10  
   Get System Time/Date, F-9  
   Polish String Evaluation, F-4  
   Set/Unset "R" Flag, F-13  
   Straight-through On, F-6,F-7  
   Target Designation, F-3  
 PROCLIB file, 3-14  
 program interruption, 4-12

## Q

Q-pointers, 3-6  
   setting-up, 9-66  
 QFILE item, 9-66  
 QUEUE-SETUP PROC, 7-64,G-9

## R

raise DTr & RTS, 9-10

READ-1900 verb, F-34etc.  
 READ-2900 verb, F-48etc.  
 retrieval locks, 5-22

## S

S-DUMP verb, 9-36  
 S/NAME attribute, 3-6  
 safeguarding save tapes, E-1  
 SAVE verb, 10-32  
 SEARCH verb, 9-62  
 security, 5-22  
 SEL-RESTORE verb, 10-16  
 selective-restores, 10-16  
 separation, 2-10  
   selection of, 6-6  
 SET-ALLOCATION program, 6-6  
 SET-DATE verb, 9-46  
 SET-FILE PROC, 9-66  
 SET-PRIV verb, 9-54  
 SET-TERM verb, 9-4,9-26  
 SET-TIME verb, 9-46  
 SET-WRITES verb, 10-2etc.  
   formula #1, 10-4  
   formula #2, 10-6  
 setting device characteristics,  
   9-24etc.,G-1etc.  
 SETUP-9843 PROC, G-2  
 SETUP-ASSY PROC, 10-36  
 SHUTDOWN PROC, 10-30  
 single-level file, 3-2,3-12  
 SIZE-ITEM-LOCKS verb, 10-66  
 SLEEP verb, 9-48  
 SORT4 PROC, G-9  
 SP-ASSIGN verb, 7-6  
 SP-COPIES verb, 7-36  
 SP-CREATE verb, 7-16  
 SP-DELETE verb, 7-38  
 SP-DEVICE verb, 7-22  
 SP-DUMP verb, 9-36  
 SP-EDIT verb, 7-46  
 SP-EJECT verb, 7-26  
 SP-FORM verb, 7-18  
 SP-FQDELETE verb, 7-24  
 SP-JOBS verb, 7-28etc.  
   Act. 1 (Move FQ), 7-30  
   Act. 10 (Change Priority), 7-56  
   Act. 2 (Move Print Job), 7-32  
   Act. 3 (Change Options), 7-34  
   Act. 4 (Chge copies), 7-36  
   Act. 5 (Delete Job), 7-38  
   Act. 6 (Stop Printing), 7-40  
   Act. 7 (Resume printing), 7-42

SP-JOBS verb cont.  
   Act. 8 (Edit Print Job), 7-46etc.  
   Act. 9 (Suspend printing), 7-44  
 SP-KILL VERB, 7-44  
 SP-LOOK verb, 7-10  
 SP-MOVEQ verb, 7-30  
 SP-OPTS verb, 7-34  
 SP-PORTOFF verb, 7-62  
 SP-PORTON PROC, 7-58  
 SP-PRIORITY verb, 7-56  
 SP-RESUME verb, 7-42  
 SP-SKIP verb, 7-26  
 SP-STATUS verb, 7-12  
   Act. 1 (Create FQ), 7-16  
   Act. 2 (Change FQ), 7-18  
   Act. 3 (Chge. Form Type), 7-20  
   Act. 4 (Chge. Dev. Ass.), 7-22  
   Act. 5 (Change page skip), 7-26  
   Act. 6 (List Print Jobs), 7-28  
   Act. 7 (Delete FQ), 7-24  
   Act. 8 (Page eject), 7-26  
 SP-STOP verb, 7-40  
 SP-SUSPEND verb, 7-44  
 SP-SWITCH verb, 7-32  
 SP-TAPEOUT verb, 7-68  
 SP-TYPE verb, 7-20  
 SPACE PROC, 9-72  
 special control characters, 4-5  
 special system utilities, 9-1etc.  
 spooler, 7-1etc.  
   options with, 7-7  
   summary of verbs, 7-5  
 spooler form-queues, G-9  
 ST-DUMP verb, 8-12  
 STANDARD form queue, 7-2  
 STAT-FILE dictionary, 10-52,  
   C-1etc.  
 status (device), 7-13  
 straight-through printing, H-1etc.  
   assembler utilities, F-6etc.  
   prog. info. (9747), H-15etc.  
   programming information, H-3etc.  
   with SP/MP T.E., H-1etc.  
 SUBD verb, 9-2,9-3  
 subvalue mark (SVM), 2-12  
 subvalues, 2-10  
 SUBX verb, 9-2,9-3  
 SYS-UPDATE PROC, 10-40  
 SYSPROG account, 3-12,10-1etc.  
 SYSPROG-PL file, 3-12  
 system  
   'error' message formats, 5-4  
   'error' messages, B-1etc.  
   configuration display, 9-45,9-50

system cont.  
   error recording, 10-46  
   maintenance procedures, 10-1etc.  
   overview, 1-8etc.  
   privilege levels, 5-14,9-50,9-54  
 System Dictionary (SYSTEM), 1-10,  
   3-10  
   user identification items,  
     5-12etc.  
   user(account) name item, 10-36  
 SYSTEM-LOG file, 3-14  
 SYSTEM-OBJECT file, 3-12  
 SYSTEM-SETUP PROC, 5-20,7-64,  
   10-20,10-44

## T

T-ATT verb, 8-4  
 T-BCK verb, 8-6,8-8  
 T-CHK verb, 8-7,9-73  
 T-COPY verb, 8-24  
 T-DET verb, 8-6  
 T-DUMP verb, 8-12  
 T-ERASE verb, 8-8  
 T-FWD verb, 8-6,9-72  
 T-LOAD verb, 8-14  
 T-PRINT PROC, 7-68  
 T-RDLBL verb, 8-20  
 T-READ verb, 8-10  
 T-RETENSION verb, 8-8  
 T-REW verb, 8-7  
 T-UNLOAD verb, 8-7  
 T-WEOF verb, 8-7  
 TABLE-CHECK verb, F-28etc.  
 TABLE-LOAD verb, F-28etc.  
 tape  
   cartridge(1/4 in), 8-8  
   labels, 8-18  
   verbs, 8-1  
 TCL PLUS, I-1  
   enter mode, I-1,I-3  
   modes of operation, I-1  
   recall mode, I-1,I-2  
   SELECT & SSELECT command, I-1,I-3  
 TCL+  
   see TCL PLUS, I-1  
 TCL-I  
   input statements, 4-10  
   option specifications, 4-5  
 TCL-II  
   input statements, 4-8  
   option specifications, 4-9  
 TCLINK verb, 10-42

TE-VERIFY verb, H-17  
 temporary work space, 2-8  
 TERM verb, 9-4,9-24  
 terminal characteristics, 9-4,  
     9-12,9-24etc.,G-1etc.  
 terminal control language, 4-1etc.  
 terminal executives, H-1etc.  
     Model 12080, H-1  
     Model 6749, H-13  
     Model 6762, H-13  
 terminal independent process,  
     11-1etc.  
 terminal printer, G-1etc.,H-1etc.  
 TERMINAL PROC, 9-4,9-28  
 terminal type, 9-12  
 TERMINAL-OBJECT file, 3-12  
 TIME verb, 9-46  
 TIMESLICE verb, 9-52  
 TIPH, 11-1etc.  
     displaying processes, 11-12  
     halting and reactivating, 11-10  
     initialisation messages, 11-14  
     initiating a process, 11-6  
     summary of commands, 11-3  
     termination messages, 11-16  
 TO-VERIFY verb, H-17  
 TRACE PROC, G-9  
 TSYM file, 9-60  
 typeahead, 9-14,9-22  
 TYPEAHEAD-OFF verb, 9-22  
 TYPEAHEAD-ON verb, 9-22

## U

UOOEO check program cataloged,  
     F-15  
 UO190 polish string, F-4  
 UO19F stop<-->comma<-->space, F-22  
 UO1A4 ASCII <--> ICL 7 track, F-51  
 UO1A7 overpunch./num. conv. 1,  
     F-18  
 UO1A9 strip spaces, F-25  
 UO200 Read/Write Tape Block,  
     F-40etc.  
 UO201 read/write SOD sentinel,  
     F-42  
 U10DD get system serial no., F-15  
 U10EO get run-time opts., F-15  
 U119B straight-through on, F-6,  
     H-10  
 U119Fstop<-->comma, F-22  
 U11A4 ICL 1900 bin <--> dec, F-51

U11A7 overpunch./num. conv. 2,  
     F-19  
 U11A9 strip specified char, F-25  
 U1201 read/write EOD sentinel,  
     F-42  
 U20EO get TCL input stat., F-15  
 U219A ASCII<-->H'Well 7 track,  
     F-55  
 U219B buffer printer output, F-6,  
     H-10  
 U21A4 fixed length records, F-53  
 U21A7 overpunch./num. conv. 3,  
     F-19  
 U21A8 get system time/date, F-9  
 U21A9 strip all excpt 0-9 A-Z,  
     F-25  
 U21AC, F-28,F-32  
 U2201 tape block-size, F-44  
 U307A set process to sleep, F-15  
 U30EO check for SELECT lists, F-15  
 U319A ASCII<-->H'Well 7 track,  
     F-55  
 U319B straight-through off, F-7,  
     H-10  
 U319F total multi/sub values, F-14  
 U31A4 fixed length records, F-53  
 U31A8 get line no. + acc. name,  
     F-10  
 U31A9 rep char with another, F-26  
 U31AC reduce rptd chars to one,  
     F-27  
 U407A set process to sleep, F-15  
 U40EO toggle msg. print., F-16  
 U419F total attr + values, F-14  
 U41A4 check tape attached, F-8  
 U41A9 <> to (), F-26  
 U41AC input and reduce chars, F-19  
 U50BB get process no./acct., F-15  
 U519F <> to/from (), F-23  
 U51A8 error handling routine, F-11  
 U60EO get term. pge width, F-16  
 U61A8 formatted error handling,  
     F-11  
 U61A9 enable break key, F-13  
 U70EO set term. echo on, F-16  
 U719F decimal <--> hex, F-24  
 U71A9 disable break key, F-13  
 U80EO set term. echo off, F-16  
 U819F get no. of items sel., F-16  
 U81A9 set R flag, F-13  
 U90EOt concat.char.'t', F-16  
 U919F dec/hex --> binary, F-24  
 U91A9 unset R flag, F-13  
 UNLOCK-FRAME verb, 10-56

update locks, 5-22  
UPDATE-ACCOUNT PROC, 10-40  
upper case only, 9-13  
user characteristics, 9-14  
user identification items,  
5-12etc.  
USER verb, 9-50  
utility PROCs, G-1etc.

## V

V/CONV attribute, 3-8  
V/MAX attribute, 3-5,3-9  
V/TYP attribute, 3-4  
value mark (VM), 2-12  
verb definition items, 4-6  
verb utilities, F-28etc.  
ICL READ/WRITE-1900, F-33  
ICL READ/WRITE-2900, F-45  
Table-Load/Table Check/U21AC,  
F-28etc.  
VERIFY-SYSTEM PROC, 10-40  
verifying T.E. codings, H-17  
virtual memory, 2-2

## W

WHAT verb, 9-45  
WHERE verb, 9-48  
WHO verb, 9-45  
work space (additional)  
assignment, 5-16etc.  
displaying, 9-50  
for FIX-FILE-ERRORS verb, 9-60  
for TIPH, 11-2  
parameter, 5-16  
specification, 5-14  
WORKSPACE verb, 5-18  
WRITE-1900 verb, F-33  
WRITE-2900 verb, F-47

## X

XTD verb, 9-2,9-3

