

PICK[®] REV 3.8 OPERATOR GUIDE

88A00916A01-A

Copyright © by General Automation, Inc.
1055 South East Street P.O. Box 4883
Anaheim, California 92803
(714) 778-4800 (800) 854-6234
TWX 910-591-1695 TELEX 685-513

RECORD OF REVISIONS

Title: PICK® Rev. 3.8 Operator Guide

Document No. 88A00916A01-A

Date	Revision Record
Oct 88	Original Issue
Jan 89	Final Rev A Update

NOTICE

The information contained in this document is subject to change without notice.

General Automation makes no warranty or representation with regard to this material, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. General Automation shall not be liable for errors contained herein.

Copyright 1988, General Automation, Inc. All rights reserved.
Copyright 1980, Pick Systems. All rights reserved. No part of this document may be photocopied, reproduced or transferred in any manner without the prior written consent of General Automation.

ACCESS is a trademark of Pick Systems
ACCU-PLOT is a trademark of AccuSoft Enterprises
CompuSheet+ is a trademark of Raymond-Wayne Corporation
JET is a trademark of JET Software
MC68000, MC68010, and MC68020 are registered trademarks of Motorola Corporation
PICK is a register trademark of Pick Systems
ZEBRA is a registered trademark of General Automation, Inc.

Related ZEBRA/PICK documents that are available to the user:

<u>Document Number</u>	<u>Title</u>
88A00751A	Overview of the PICK Operating System
88A00758A	ACCU-PLOT Operator Guide ZEBRA/PICK Graphics System
88A00760A	Quick Guide for the PICK Operating System
88A00774A	PICK Utilities Guide
88A00775A	ZEBRA Hardware Reference Manual
88A00776A	ACCESS Reference Manual
88A00777A	PICK SPOOLER Reference Manual
88A00778A	PICK BASIC Reference Manual
88A00779A	EDITOR Reference Manual
88A00780A	PROC Reference Manual
88A00781A	RUNOFF Reference Manual
88A00782A	Introduction to PICK TCL and FILE STRUCTURE
88A00783A	PICK JET Word Processor Guide
88A00829A	CompuSheet+ Operator Guide

TABLE OF CONTENTS

<u>Section</u>	<u>Title</u>	<u>Page</u>
1	UPGRADING TO THE 3.8 RELEASE	1-1
1.1	UPGRADING TO A NEW SOFTWARE RELEASE	1-2
2	INTRODUCTION	2-1
2.1	DYNAMIC WORKSPACE	2-1
2.2	SYS3 PRIVILEGE LEVEL.	2-1
2.3	ITEM STRUCTURE (PHYSICAL)	2-2
2.4	FILE STRUCTURE.	2-5
2.5	SUMMARY OF NEW FEATURES	2-5
3	FILE MANAGEMENT.	3-1
3.1	LISTS	3-1
3.2	FILE DEFINITION ITEMS	3-2
3.3	USING PATH NAMES.	3-2
3.4	CREATE-FILE VERB.	3-4
3.5	DELETE-FILE VERB.	3-5
3.6	RENAME-FILE VERB	3-6
3.7	CHANGING FILE TYPE (DC/DX/DY/DZ).	3-7
3.8	MODIFYING D-POINTERS WITH SYSTEM EDITOR	3-7
3.9	GROUP FORMAT ERROR HANDLER.	3-8
3.9.1	(E)XIT.	3-8
3.9.2	(I)GNORE.	3-8
3.9.3	(F)IX	3-9
3.9.3.1	Invalid Forward Link	3-9
3.9.3.2	Invalid Backward Link	3-9
3.9.3.3	Invalid Item Size	3-9
3.9.3.4	Embedded Segment Mark	3-10
3.9.3.5	Missing Ending Segment Mark	3-10
3.9.3.6	Incorrectly Terminated Item	3-10
3.9.3.7	Invalid Item Flags	3-10
3.9.3.8	Illegal Indirect Item FWD Link.	3-10
3.9.3.9	Illegal Indirect Item BWD Link.	3-11
3.9.3.10	Invalid Indirect Item Size.	3-11
3.9.3.11	Illegal Binary Item FWD Link.	3-11
3.9.3.12	Illegal Binary Item BWD Link.	3-11
3.9.3.13	Invalid Item Hash Address	3-11
3.9.3.14	Duplicate Item-id	3-11
3.9.3.15	Item-Id Longer than 50 chars.	3-12
3.9.3.16	Item-Id Contains Sys. delimiter	3-12
3.9.4	(L)IST ITEM	3-12
3.9.5	(C)UT ITEM.	3-12
3.9.6	(K)ILL GROUP.	3-12
3.9.7	(T)RUNCATE GROUP.	3-12
3.9.8	(D)EBUGGER.	3-12
3.9.9	SPECIAL CASE HANDLING OF GFE'S	3-13
3.9.9.1	Splr GFE During Error Logging	3-13
3.9.9.2	FILE-SAVE G Option	3-13
3.9.9.3	GFE's and Next Item Pointer	3-13

<u>Section</u>	<u>Title</u>	<u>Page</u>
3.10	SYSTEM BACKUP	3-14
3.10.1	SAVE.	3-14
3.10.1.1	OPTIONS	3-16
3.10.2	FULL SYSTEM SAVE.	3-18
3.10.3	INCREMENTAL SYSTEM SAVE	3-18
3.10.4	FILE-SAVE	3-19
3.10.5	UPDATE-SAVE	3-21
3.10.6	ACCOUNT SAVES	3-21
3.10.7	ERROR HANDLING & LOGGING.	3-21
3.10.8	SELECTIVE SAVE AND THE T-DUMP VERB.	3-21
3.11	SYSTEM RESTORE.	3-22
3.11.1	BOOT TIME FILE RESTORE.	3-22
3.11.2	THE :FILELOAD VERB.	3-23
3.11.3	THE RESTORE VERB.	3-23
3.11.4	THE ACCOUNT-RESTORE VERB.	3-24
3.11.5	OPTIONS	3-25
3.11.6	ADD-ACCOUNTS	3-26
3.11.7	VERIFY-SAVE VERB.	3-28
3.11.8	VERIFY-SAVE-DATA VERB	3-28
3.11.9	ERROR HANDLING DURING RESTORE	3-28
3.11.9.1	HANDLING OF ERRORS NOT STORED IN THE RESTORE-ERRORS ACCOUNT	3-29
3.11.10	SELECTIVE RESTORE AND THE T-LOAD VERB	3-30
3.11.10.1	T-LOAD OPTIONS.	3-31
4	ACCESS	4-1
4.1	ACCESS PRE-PROCESSOR.	4-1
4.1.1	PHRASES	4-1
4.1.2	PHRASE USAGE EXAMPLES	4-2
4.1.3	QUOTATION MARKS	4-5
4.1.4	OPTIONS	4-5
4.1.5	ACCESS STATEMENT EXAMPLES WITHOUT QUOTATION MARKS	4-6
4.2	ACCESS "ANY" DEFINITION	4-7
4.3	CONVERSION AND CORRELATIVE CODES.	4-7
4.3.1	Center (XC)	4-7
4.3.2	Change Conversion Direction (XT, XI, XO).	4-7
4.3.3	Running Sub-Total (XR).	4-9
4.3.4	Function Processor, F Code:	4-9
4.3.5	Function Processor, A Code:	4-12
4.3.6	File Translation Code, Tfile.	4-12
4.4	ACCESS USER-EXIT.	4-13
4.4.1	EXCHANGE CHARACTERS -- U21A1.	4-13
4.5	LIST MANIPULATION VERBS	4-14
4.5.1	CONCATENATING LISTS (GET-LIST).	4-14
4.5.2	MERGING LISTS (GET-LIST).	4-14
4.5.3	DELETING ITEMS FROM A LIST USING A SECOND LIST	4-14
4.5.4	SORTING LISTS (SORT-LIST)	4-15

<u>Section</u>	<u>Title</u>	<u>Page</u>
5	BASIC.	5-1
5.1	BASIC VERBS	5-4
5.1.1	RUNI.	5-4
5.1.2	BLIST	5-4
5.1.2.1	BLIST CUSTOMIZATION	5-5
5.1.3	BFORMAT	5-6
5.1.3.1	BFORMAT CUSTOMIZATION	5-7
5.1.4	RENUMBER.	5-7
5.2	ASSIGNING VALUES TO VARIABLES	5-7
5.3	BASIC or COMPILE.	5-8
5.4	BASIC ITEM LOCKS.	5-8
5.4.1	CHANGES TO BASIC STATEMENTS	5-9
5.4.1.1	READ, READV, and MATREAD.	5-9
5.4.1.2	READU, READVU and MATREADU	5-9
5.4.1.3	WRITE, WRITEV, MATWRITE, DELETE	5-9
5.4.1.4	WRITEU, WRITEVU and MATWRITEU	5-9
5.4.1.5	RELEASE	5-9
5.4.1.6	STOP, ABORT and CHAIN	5-9
5.4.2	DISPLAYING GROUP AND ITEM LOCKS	5-10
5.5	BASIC CONVERSION CODES.	5-10
5.5.1	Center (XCn).	5-10
5.6	BASIC SUBROUTINES (USER-EXITS).	5-11
5.6.1	GET INPUT LINE — U31A1	5-11
5.6.2	PARSE INPUT LINE — U41A1	5-11
5.7	BASIC STATEMENTS.	5-12
5.7.1	\$INSERT, INSERT, \$INCLUDE, INCLUDE.	5-12
5.7.2	CLEARFILE	5-12
5.7.3	READING AND WRITING LISTS	5-13
5.7.4	LOCATE.	5-13
5.7.5	CALL and ENTER.	5-14
5.7.6	PRINTCHAR	5-14
5.7.7	INSERT OR REPLACE VALUE WITHIN VARIABLE STRING.	5-15
5.7.8	INPUT @	5-16
5.7.9	INPUTVAR @	5-17
5.7.10	DATE INPUT	5-18
5.7.11	EDITTRAP.	5-18
5.7.12	CANCELTRAP	5-20
5.7.13	IFR	5-21
5.7.14	REMOVE.	5-23
5.7.15	ERROR	5-24
5.8	BASIC FUNCTIONS	5-25
5.8.1	SWAP()	5-25
5.8.2	INMAT()	5-25
5.8.3	SENTENCE().	5-26
5.8.4	INDEX	5-26
5.8.5	EREPLACE.	5-27
5.8.6	SCAN.	5-27
5.8.7	SUM	5-28
5.8.8	SYSTEM	5-28
5.8.9	TRIME	5-30
5.8.10	TRIMB	5-30

<u>Section</u>	<u>Title</u>	<u>Page</u>
6	MULTIPLE TAPE UNITS.	6-1
6.1	CLEAR-CASCADE	6-2
6.2	RESET-CASCADE	6-2
6.3	SET-CASCADE	6-2
6.4	SET-MT.	6-3
6.5	T-ATT	6-4
6.6	T-COPY.	6-6
6.7	T-DET	6-7
6.8	T-REW	6-8
6.9	T-STATUS.	6-8
6.10	T-UNLOAD.	6-9
7	SMA STANDARD 1/4-INCH TAPE	7-1
7.1	T-FORMAT.	7-2
7.1.1	T-ATT	7-2
7.1.2	T-CHK	7-2
7.1.3	T-EOD	7-2
7.1.4	T-EOFD.	7-2
7.1.5	T-REW	7-3
7.1.6	T-WEOF.	7-3
8	FLOPPY DISK	8-1
8.1	FORMAT-FD	8-2
8.2	SET-FADDR	8-2
8.3	SET-FD.	8-3
8.4	VERIFY-FD	8-3
9	PROTOCOL	9-1
9.1	PROTOCOL VERB	9-2
9.1.1	EXAMPLE PROTOCOL DISPLAY	9-7
9.2	BASIC PROTOCOL FUNCTIONS	9-8
9.3	PROTOCOL CABLING	9-10
10	SYSTEM VERBS AND PROCS	10-1
10.1	:BTI AND :ITB VERBS	10-1
10.1.1	:BTI VERB	10-1
10.1.2	:ITB VERB	10-1
10.2	:SET-XOVR	10-2
10.3	:SWC.	10-2
10.4	:SWD.	10-3
10.5	:SWX.	10-3
10.6	:SWY.	10-3
10.7	:SWZ.	10-3
10.8	COMPARE	10-4
10.9	RENAME-FILE	10-5
10.10	CREATE-FILE	10-6
10.11	DELETE-FILE	10-7
10.12	MESSAGE or MSG.	10-8
10.13	PEEK	10-9

SECTION	TITLE	PAGE
10.14	PEEK-STOP	10-9
10.15	PEEK-ON	10-9
10.16	PEEK-OFF	10-9
10.17	PEEK-STAT	10-9
10.18	POKE.	10-10
10.19	POKE-OFF.	10-12
10.20	POKE-ON	10-12
10.21	POKE-RESET.	10-12
10.22	T-DUMP.	10-12
10.23	T-LOAD.	10-13
10.24	VERIFY-SAVE	10-14
10.25	VERIFY-SAVE-DATA.	10-15
10.26	LIST-LOCKS.	10-15
10.27	WHAT.	10-16
10.28	CLEAR-GROUP-LOCKS	10-18
10.29	CLEAR-ITEM-LOCKS.	10-19
10.30	CLEAR-LOCKS	10-19
10.31	LIST-GFES	10-19
10.32	GFE.RECOVER	10-19
10.33	ADD.UPDATE.MD	10-19
10.34	DELETE.MD	10-21
10.35	UPDATE.MD	10-21
10.36	CONVERT.MD	10-22
10.37	CHECK-SUM, STAT, SUM	10-22
10.38	DEFAULT-MODSEP	10-23
10.39	DATE-FLAG	10-23
10.40	TANDEM	10-23
10.41	TANDEM-ON	10-24
10.42	TANDEM-OFF	10-24
10.43	TANDEM-STAT	10-24
10.44	DUMP.	10-25

LIST OF APPENDIXES

<u>Appendix</u>	<u>Title</u>	<u>Page</u>
A	LIST OF VERBS BY SYSTEM PRIVILEGE	A-1
B	LIST OF PROCS BY SYSTEM PRIVILEGE	B-1
C	LIST OF SYSPROG ONLY PROCS AND VERBS.	C-1
D	LIST OF 3.8 ERROR MESSAGE FILE.	D-1

LIST OF TABLES

5-1	BASIC STATEMENTS	5-2
5-2	BASIC INTRINSIC FUNCTIONS	5-2
5-3	RESERVED WORDS	5-3

upgrading to the 3.8 release

1

Before upgrading to the 3.8 release, it is recommended that the user become familiar with the attributes of the new release by reading carefully the following sections of this manual. Whether the user is upgrading from a previous version of the General Automation Pick Operating System or from another vendor's version of the Pick operating system, it will be beneficial to review this manual first.

The following paragraphs are points that the user should be aware of when attempting an upgrade.

There are three 1/4-inch tape formats, Old, New or Normal, and SMA standard. All GA models running the 3.8 version of the Pick Operating System can read and write "Normal" and "SMA" standard format. The 1350, 1750 and 3750 models can read "Old" format; but cannot write "Old" format. "New" format, now called "Normal" format, is the system default for 1/4-inch tape. "SMA" format will allow the user to read another vendor's single reel 1/4-inch tapes and other vendors to read GA single reel 1/4-inch tapes. There is no SMA standard for multi-reel 1/4-inch tape. Therefore, the ability to exchange multi-reel SMA standard 1/4-inch tape between vendors is not to be relied upon.

Upgrading a 1350, 1750 or 3750 to 3.8 when the user's 1/4-inch tapes are in the "Old" format requires the user to do individual account restores or T-LOADs. The ADD-ACCOUNTS program does not recognize "Old" format.

When restoring files that have a separation greater than 15, the user will be prompted for a new modulo even though the modulo may be correct, this forces the separation to 1. Therefore, if the user has files with a separation greater than 15 and would like the separation larger than 1, he should adjust the file re-allocation parameters in attribute 13 of the file D-pointer before saving the file.

Other vendors, Ultimate in particular, force a change to the item hashing algorithm on files greater than 64K. They also have an option to change the hashing algorithm on files greater than 32K. Unless there are re-sizing parameters in attribute 13 of the file D-pointer which cause the file to be re-hashed during the restore, various problems will result once the file is restored on the GA system. The re-sizing parameters must have a modulo different than the original modulo. If making a tape with new re-sizing file parameters is not an option, then the file may be restored even though it is hashed incorrectly. Before using the file, create a new file with a slightly different modulo and copy from the old file to the new file. Then delete the old file and use the RENAME-FILE verb to change the new file name to the original name. This uses up disk space temporarily and is not the most desirable option, but with careful disk management by the user it will allow the data to be restored and accessed correctly.

1.1 UPGRADING TO A NEW SOFTWARE RELEASE

If the user is running a previous version of the GA PICK Operating System, or is transferring files from another vendor's machine, following the steps below will allow a smooth transition to a new version of the operating system.

1. LOGOFF all users.
2. Do FILE-SAVE and printout STATS.
Remember to change file re-allocation parameters if necessary before the FILE-SAVE.
3. T-DUMP any user items in system type files (ERRMSG, PROCLIB or SYSPROG-PL, etc.).
4. LOAD SYSGEN tape in the tape unit.
5. Place Key in the SERVICE position and press RESET
6. The command function starts at the System Console and then control is passed to Line 0.

At the prompt type:

Ok. BOOT MT <cr>
Ok. BOOT CT <cr>

The executive will respond with:

Mount Tape 1 (y/n): Y<cr>

If your system is 1620, 1820, 2820 or 8830, the MONITOR, BOOT_STRAP and other executable files are loaded similar to the following:

0200 hh:mm:ss dd mmm yy	SYS_TEST	01
0200 hh:mm:ss dd mmm yy	README	01
0200 hh:mm:ss dd mmm yy	VR	01
0200 hh:mm:ss dd mmm yy	REPORT	01
0200 hh:mm:ss dd mmm yy	INIT	01
0200 hh:mm:ss dd mmm yy	MONITOR;\$nnnn;\$nnnnnnnn	01
0200 hh:mm:ss dd mmm yy	BOOT_STRAP;\$nnnn;\$nnnnnnnn	01
0200 hh:mm:ss dd mmm yy	CT_STRAP;\$nnnn;\$nnnnnnnn	01

Control passed to line 0

7. Put Key in RUN/ON position

On Line 0, the system will respond with:

ZEBRA ~~xxxx~~ SYSGEN LOADER

RESTORE SYSTEM - R

RESTORE ABS - A

8. ENTER OPTION ->R

The loader will display:

Loading and Verifying Pick Monitor

Pick Monitor loaded and verified

Loading and verifying ABS

Loading ABS frame -> ~~xxxx~~

When the Monitor and ABS are loaded, the loader displays:

ABS LOADED AND VERIFIED

~~xxxx~~K MEMORY

~~xx~~ COMM LINES

SPOOLER STARTED

When the File Restore from the SYSGEN tape is finished, the system will begin an automatic COLDSTART.

9. The program, ADD-ACCOUNTS, will automatically restore accounts not already resident on the system from a FILE-SAVE tape. Mount FILE-SAVE tape and logto SYSPROG. At TCL, enter ADD-ACCOUNTS. If all accounts are not needed, then individual ACCOUNT RESTORES may be faster. If ADD-ACCOUNTS program is not used to restore accounts, DO NOT restore SYSPROG as a new SYSPROG was loaded with the system. *** DO NOT DO A FILE RESTORE (F option or :FILELOAD) OF OLD DATA ***
10. Log to SYSPROG and run ADD.UPDATE.MD on all accounts.

introduction 2

This manual is a supplement to the current range of Zebra Pick Manuals. The information found in this manual supercedes coverage in other GA Pick manuals and is intended for use with the 3.8 version of the Zebra Pick Operating System.

2.1 DYNAMIC WORKSPACE

The operating system will allow the extension of process workspace as required. This feature automatically extends the process workspace for TCL, BASIC, EDITOR, and other processors when the current workspace becomes too small. The workspace is always extended by an amount equal to the system workspace size. The system workspace size is 127 frames. Therefore, as additional space is required, an additional 127 contiguous frames will be added. The extra workspace is returned to overflow at LOGOFF, or in the case of EXECUTE, when returning from a level.

2.2 SYS3 PRIVILEGE LEVEL

The SYS3 privilege level will bypass all file security. The SYSPROG account must have SYS3 privileges, as well as the dummy accounts FILE-SAVE and COLD-START. SYS3 privileges now take the place as the highest priority on the system. SYS3 privileges bypass retrieval and update locks and allow functionality of all system verbs. Prior to 3.8 only SYSPROG had such super priority.

2.3 ITEM STRUCTURE (PHYSICAL)

An item consists of one or more variable length attributes, separated by attribute marks. An attribute mark is a character with a value of FE (hexadecimal), which displays as ^. The size of an item may be as large as 2 gigabytes, but a practical limit is approximately 16 megabytes, the longest string allowed in a BASIC program. The first attribute in an item (attribute 0) is the item-id. The item-id is preceded by a two character binary count field which specifies the total number of characters in the item (including the count and flags fields), and a two character flags field which is used to indicate item type, and to track updates and deletions on an item by item basis for use by the incremental file-save. The item is terminated by an attribute mark, followed by a segment mark (hexadecimal FF). Every item is aligned on an even (word) boundary. Thus, the item size field will always be an even number. A pad character (hexadecimal 81) may follow the terminating segment mark if the item's true size is an odd number. For example, consider the following stored item:

```
.(.X123^SMYTHE^JOHN^1234 MAIN STREET^_.
```

Attribute 0 is the item-id (X123). It is preceded by .(. (actually hexadecimal 00288003) which are the item count (0028 hexadecimal, 40 decimal), and item flags fields. Attribute 1 of X123 is SMYTHE, attribute 2 is JOHN, and attribute 3 is 1234 MAIN STREET. The item ends with ^_. (actually hexadecimal FEFF81) which is attribute mark, segment mark and pad character 81.

When the length of an item reaches a pre-determined value, the item becomes "indirect". That is, the item is copied to its own set of disk frames, and a pointer to the item is stored in the file. In this case, the format of the pointer is two character binary count field (size of the pointer), two character flags field (with the "indirect" flag set), item-id, attribute mark, 6 character hexadecimal pointer, attribute mark, segment mark. A pad character (hexadecimal 81) may follow the terminating segment mark. The format of the indirect item is 4 character binary count field (true size of the data part of the item), item-id, attribute mark, item body, attribute mark, segment mark. There are no flags in the indirect element. The flags are held in the pointer only.

The pre-determined value at which an item becomes "indirect" may be set using the :SET-XOVR verb. The value specified must be between 250 and 32767 minus the data frame size. If linked frame is 512 bytes, the data frame is 500 bytes, thus 32267 is the maximum size before the item becomes indirect. If linked frame is 2048 bytes, the data frame is 2000, thus 30,767 is the maximum size before an item becomes indirect. Altering this value will effect file sizing, disk space usage, and system performance. System default is 32267 for 512 byte data frames.

Since every item is aligned on an even (word) boundary, the item size field will always be an even number, preventing a segment mark in the lower byte of the item size. Also, since the (in group) item size must be less than 32767 minus the linked frame size, the upper byte cannot contain a segment mark. The flags field is defined so that valid item flags will never contain a segment mark. Thus, segment mark remains a unique item terminator.

The item flags field is defined as follows:

bit 0 - ITM.MSB	Always set
bit 1 - reserved	
bit 2 - ITM.DPTR	Set if item is a D pointer
bit 3 - reserved	
bit 4 - ITM.DDEL	Set if D pointer has been deleted
bit 5 - ITM.DUPD	Set if D pointer has been updated
bit 6 - ITM.IDEL	Set if item has been deleted
bit 7 - ITM.IUPD	Set if item has been updated
bit 8 - reserved	
bit 9 - ITM.BIN	Set if item is binary (linked or contiguous)
bit A - ITM.LNK	Set if binary item is linked
bit B - ITM.IND	Set if item is indirect
bit C - reserved	
bit D - reserved	
bit E - ITM.ODD	Set if item is odd
bit F - ITM.LSB	Always set

Additionally, each group contains a group flags field, usually the first byte of the link field of the first frame of each group. This field contains two flags:

- bit 0 - file updated flag
- bit 1 - group updated flag

The "file updated" flag is valid only in the first frame of the file (BASE FID), and indicates whether any updates to the file have occurred since the last full file-save. The "group updated" flag is set on a group by group basis, and indicates whether any items in the group have been updated since the last full file-save.

Since the item size is always an even value, and an actual item may be any size (even or odd), a bit in the flags (ITM.ODD) indicates if an item size is actually odd. In this case, a pad character (X'81') follows the terminating segment mark.

Two classes of items are supported: Normal items and Binary items. Normal items are stored directly in the group or indirectly out of the group. If the item size is larger than a pre-determined size, then it is stored indirectly, otherwise it is stored directly in the group. Binary items may be either contiguous (BASIC object code), or linked (Unused at this time). Binary items may not be accessed by BASIC or the EDITOR.

The item flags field has four bits defined to control item storage and access. These bits are ITM.IND, set if an indirect item is accessed; ITM.BIN set if a binary item is accessed; ITM.LNK set if a binary item is linked (reset if it is contiguous); and ITM.DPTR set if an item is a File Definition Item (D pointer).

The system routines RETIX and GETITM update the ITM.FLGS element in the PCB. The SIZE element of the PCB has been moved and expanded to 4 bytes.

Indirect and Binary items contain "pointers" in the body of the group which point to the actual item. The "pointers" consist of a count field, flags field, item-id, attribute mark, 6 byte hexadecimal FID, attribute mark, segment mark. If the item is a binary contiguous item, then a 4 byte hexadecimal frame count and attribute mark follows. The item is terminated with a segment mark.

The FID for an indirect item points to the first frame of a linked set which contains the referenced item. The structure of the indirect item is similar to the item in the group: 4 byte binary count field, item-id, attribute mark, item body, attribute mark, segment mark.

The FID for a binary item points to the first frame of the binary data.

Certain item flags are used to control the incremental file-save. These are ITM.DDEL (set when a D pointer is deleted), ITM.DUPD (set when a D pointer is created), ITM.IDEL (set when a normal item is deleted and not removed from the group), and ITM.IUPD (set when a normal item is updated or created). These flags are modified by the system routines UPDITM and UPDDITM; and the FILE-SAVE process.

Note: When updates for a particular file are required to be "flagged" by the system routine UPDITM, and an item is deleted from the file, the body of the item is removed from the group, leaving the item-id in the group. The flags are marked appropriately, and the item will not be on file when accessed by the system routine RETIX, but the incremental FILE-SAVE will be able to record the deleted item. These items are refiled in place, taking room only for item-id, and the reset of the group is shuffled up. Also, this means deleted items can be anywhere in a group.

2.4 FILE STRUCTURE

The file structure is unchanged except that a file does not have to be DC type to contain lists or BASIC object code, except where required for compatibility with older systems. A file type of DC is recognized by the FILE-SAVE and T-DUMP processes when producing a tape that is compatible with older systems. In this case, all items (except those in a dictionary level file with a single alpha character on line 1) are converted to list (CL) format when written to the tape. When these items are restored on older machines, they will be list (CL) items and must be restored into a DC type file. When restored on this implementation, they will be re-converted to normal items.

Maximum modulo is 1048573 with a maximum separation of 15.

2.5 SUMMARY OF NEW FEATURES

The following is a brief summary of the new features included in the 3.8 version of the Zebra Pick Operating System. Complete explanation of the new features are found in separate sections of this manual.

- Workspace frames are automatically extended as needed for the TCL, JET, BASIC, and EDITOR processors.
- Length of Basic strings have been increased to approximately 16 megabytes.
- Item length has been increased to 2 gigabytes.
- "DC" type files are no longer required for pointer-items or Basic object code, except where required for compatibility with other systems.
- Items of same size are updated in place.
- Basic item locks.
- Simultaneous use of 1/2" tape, 1/4" tape and floppy disk by multiple users.
- Addition of Floppy Disk.
- Addition of SMA standard 1/4" tape.

ACCESS UTILITIES

- Pre-defined phrases that contain parts of Access statements
- Quotation marks are no longer required for explicit values
- "ANY" attribute item allows Access the ability to search entire items value by value, attribute by attribute.
- Additional conversions to center a field, force conversion to input or output and to handle running sub-totals.

BACKUP FEATURES

- Incremental FILE-SAVE -- ability to backup new or changed data.
- Perform FILE-SAVE of selected accounts.
- Perform ACCOUNT-SAVE of selected files.
- Ability to backup a file dictionary with all data files belonging to that dictionary.

RESTORE FEATURES

- Ability to restore selected accounts from a FILE-SAVE tape.
- Ability to restore shared dictionary and associated data files.
- Ability to restore selected files.
- Ability to do a complete "DUMMY" restore to verify data integrity of FILE-SAVE tape. -- No data is actually restored.
- Ability to compare data on tape to data on disk.
- T-LOAD verb will restore from a FILE-SAVE tape.
- Ability to restore changes or new items from incremental FILE-SAVE tape.

FILE FEATURES

- CREATE-FILE verb will create "DX", "DY" and "DC" pointers.
- CREATE-FILE in different account with SYS3 privileges.
- DELETE-FILE verb can delete a file in a different account with SYS3 privileges.
- A file D-pointer cannot be edited without special SYS3 privileges.
- A file D-pointer may be moved or renamed using the RENAME-FILE verb.
- New Group Format handler that will handle GFES intelligently.
- File Q-pointers may point to other Q-pointers up to 255 levels of Q-pointers.

file management 3

3.1 LISTS

Since normal items may be of (almost) any size, no special format is required for lists. The list handling verbs (SAVE-LIST, GET-LIST, EDIT-LIST, DELETE-LIST, RENAME-LIST) still assume the POINTER-FILE is used to store lists, but does not have to be a DC type file. As with previous versions of the operating system, an optional file name may be specified for all list handling verbs. The optional file is no longer required to be a "DC" type file. If no filename or no listname is specified, defaults are used based on the contents in attribute 6 of the verb definition item.

Lists are stored as normal items allowing access by BASIC programs through READ and WRITE statements.

In order to remain compatible with older systems, the SAVE and T-DUMP processors, when used with the "C" option, will convert certain items to "list" format on the tape. Items to be converted must be in a "DC" type file. If the file is a dictionary level file, items that have only a single alphabetic character on line 1 are not converted. See SAVE verb as described later in this section.

When loading a tape produced on an older system, lists are automatically converted to normal items. In order to inhibit this conversion when using the RESTORE verb, use the "Y" option. See RESTORE verb as described later in this section.

3.2 FILE DEFINITION ITEMS

File definition items (D-pointers) may only be created by the CREATE-FILE verb or the file restore process. They may only be deleted by the DELETE-FILE verb. The first four attributes of a file definition item Code, Base, Modulo and Separation may not be modified by the user (with the exception of the Code, which may be changed with the :SWx verbs). Duplicate file definition items may not exist.

Synonym file definition items (Q-pointers) may now refer to other synonym file definition items. This is valid as long as no circular definitions exist. The exact format of the Q-pointer depends on the level of the dictionary which contains it as defined below:

SYSTEM Dictionary

```
001 Q
002 account-name    (must be in SYSTEM)
003                 (must be null)
004...              (Same as for D pointer)
```

ACCOUNT Master Dictionary

```
001 Q
002 {account-name} (if null, current account assumed)
003 {dict{,data}} (if null, account MD assumed)
004...              (Same as for D pointer)
```

FILE Dictionary

```
001 Q
002                 (must be null)
003 {data}           (if null, file dictionary assumed)
004...              (Same as for D pointer)
```

3.3 USING PATH NAMES

Whenever a file-name is required by the system (TCL-II verbs, BASIC OPEN, CALL and ENTER statements, etc.), a path-name may be used instead. The path-name specifies a file by means of a "path", where the first file named in the path-name is defined in the user's master dictionary. The path-name syntax is:

SYSTEM

Accesses the SYSTEM file.

SYSTEM,accountname

Accesses the master dictionary of accountname.

SYSTEM,accountname,dictname

Accesses the data section of dictname in account accountname.

SYSTEM,accountname,dictname,dataname

Accesses the data section dataname of file dictname in account accountname.

These forms assume the user's master dictionary contains a Q-pointer to SYSTEM, and that the user has access privileges to each of the files along the path, including SYSTEM.

Accountname

Accesses the master dictionary of accountname.

Accountname,dictname

Accesses the data section of dictname in account accountname.

Accountname,dictname,dataname

Accesses the data section dataname of file dictname in account accountname.

These forms assume the user's master dictionary contains a Q-pointer to accountname, and that the user has access privileges to each of the files along the path.

Note that any of the above forms may be preceded by the word DICT to access the file dictionary. If the access is to a dictionary already, DICT has no effect.

It is recommended that the Q-pointer to SYSTEM in the user's master dictionary be a full 10 line Q-pointer. Therefore, when using ACCESS, the type and length attributes (lines 9 and 10) will be taken from the Q-pointer and not the D-pointer in SYSTEM. If an abbreviated Q-pointer (2 lines) is used, then the type and length attributes will be taken from the D-pointer. When accessing an account master dictionary using the abbreviated Q-pointer to SYSTEM, the password (if any) from line 7 of the D-pointer will be assumed to be a conversion by ACCESS, resulting in an "Illegal conversion code" error.

3.4 CREATE-FILE VERB

The CREATE-FILE verb is used to create new accounts, dictionaries, and data files in the system. It creates a file by reserving disk space for the file and inserting a file definition item (D-pointer) into the parent file (SYSTEM for an account, MD for a dictionary, dictionary for a data file). The general forms of the CREATE-FILE verb are:

CREATE-FILE file-name m1{,s1} m2{,s2} (options)

CREATE-FILE dict-name,data-name m1{,s1} m2{,s2} (options)

CREATE-FILE DICT dict-name m1{,s1} (options)

CREATE-FILE DATA file-name m2{,s2} (options)

CREATE-FILE DATA dict-name,data-name m2{,s2} (options)

where:

file-name	is the name of the file
dict-name	is the name of the dictionary
data-name	is the name of the data section
m1	is the modulo of the dictionary
s1	is the separation of the dictionary
m2	is the modulo of the data section
s2	is the separation of the data section

Forms which do not specify DICT or DATA create both the dictionary and data sections of the file. If DICT is specified, only the dictionary section is created. If DATA is specified, the dictionary must already exist, and only the data section is created.

Options:

C	create DC file (only DICT file pointer will be DC)
Y	create DY file (only DATA file pointer will be DY)
X	create DX file (both DICT and DATA file pointers will be DX)
R	create file in another account

Either X or Y may be specified, but not both.

The R option is used to specify another account as the destination for the file being created. SYS3 privileges are required when the R option is specified. If the R option is specified, you will be prompted:

Account ?

A valid account definition (in SYSTEM) must be entered. The account definition may be either a D-pointer or a Q-pointer. Note that specifying SYSTEM as the account will create the file in the SYSTEM dictionary. In this case, a dictionary level file only must be created.

3.5 DELETE-FILE VERB

The DELETE-FILE verb is used to delete an account (including all dictionaries and data level files), dictionary (including all data level files) or data file. The verb operates by returning all space owned by the file to the available space pool, and deleting the file definition item in the parent file. The general forms of the DELETE-FILE verb are:

DELETE-FILE file-name (option)

DELETE-FILE DICT dict-name (option)

DELETE-FILE DATA data-name (option)

DELETE-FILE DATA dict-name,data-name (option)

where:

file-name	is the name of the file
dict-name	is the name of the dictionary
data-name	is the name of the data section

The form which does not specify either DICT or DATA deletes both the dictionary and data level files.

Option:

R delete a file from another account. This option requires SYS3 privileges.

If the R option is specified, system will prompt:

Account ?

A valid account definition (in SYSTEM) must be entered. The account definition may be either a D-pointer or a Q-pointer.

To delete an account, specify the desired account as the file-name in the DELETE-FILE command (with the R option) and specify SYSTEM at the "Account ?" prompt. This will delete the desired account including all dictionaries and data level files. Note that the DICT or DATA modifiers may not be specified when using this verb to delete an account, or use DELETE-ACCOUNT.

Example: DELETE-FILE account-name (R)
 "Account ?" SYSTEM

3.6 RENAME-FILE VERB

RENAME-FILE is a TCL-II verb used to move or rename file definition pointers. Only file definition items may be operated on by this verb (not synonym definitions). The source file may be renamed or moved (and renamed) to a new dictionary. The RENAME-FILE verb always deletes the source file definition after updating the destination file definition. The source file definition is not deleted if the pointer is copied to the terminal or printer. RENAME-FILE may only move file definition items to dictionaries which are at the same level as the source dictionary. That is, a file definition item in the master dictionary may not be copied to the SYSTEM dictionary or to a file dictionary, but could be copied to another account's master dictionary. This is due to both file and item flags containing information as to the level of file they are in. It may be necessary to use the RENAME-FILE verb twice to completely rename a file, if the user desires to keep the data section name the same as the dictionary name.

```
RENAME-FILE {DICT} file-name file-definition-item {(options)}
TO: {( ) {DICT} file-name {file-definition-item}}
```

Options:

- F Form-feed. Display each item on terminal or printer onto new page.
- I Item-ids suppressed during file rename.
- N No page wait on terminal.
- P Printer is destination.
- T Terminal is destination.
- X Hexadecimal format used on terminal or printer copy

Example: (File with dict and data section of same name of FILE1 to be changed to FILE2.)

```
>RENAME-FILE MD FILE1
TO: FILE2
```

```
>RENAME-FILE DICT FILE2 FILE1
TO: FILE2
```

3.7 CHANGING FILE TYPE (DC/DX/DY/DZ)

A set of ACCESS verbs :SWC, :SWD, :SWX, :SWY, and :SWZ exist to change the file type code of a file definition item. These verbs make sure each item is a valid D-pointer. If 'D' in attribute 1 and :SWn is used, item is updated correctly. If 'D' in attribute 1 but item is not a valid D-pointer, no update occurs. Verb format is:

```
:SWD {DICT} filename item-list {(I}
```

where: I = Inhibit error message [782] 'A' updated.

The :SWC verb is used to change the file type to DC. If the file type was DX, DY or DZ, then it is changed to DCX, DCY or DCZ respectively.

The :SWD verb is used to change a DX, DY or DZ file to a D file. If the file type was not DX, DY, or DZ (or DCX, DCY, or DCZ), then no operation takes place. If the file type was DCX, DCY or DCZ, then the file type is changed to DC.

The :SWX, :SWY and :SWZ verbs change the file type to DX, DY or DZ respectively. If the file type contained X, Y or Z, it is removed first, then the new code added.

3.8 MODIFYING D-POINTERS WITH THE SYSTEM EDITOR

If a user has SYS3 privileges, the system EDITOR may be used to modify file definition items. In order to update a file definition item after changing lines 1, 2, 3, or 4, the FID, FSD or FDD editor command must be used. A warning will be issued about corruption of the file system and the operation must be verified before the update takes place. This method of modifying D pointers is not recommended. Use RENAME-FILE or :SWn verbs.

3.9 GROUP FORMAT ERROR HANDLER

The Group Format Error Handler is entered whenever the operating system detects an error in the physical format of a group (of items). The GFE handler locates the error, reports it, logs it in the SYSTEM-ERRORS file, and prompts the user for action. The errors may be displayed by using the LIST-GFES proc on the SYSPROG account.

There are 16 separate types of GFES that may be detected. The first two tests applied are for illegal forward and backward links in an entire group, and after any linkage errors have been fixed, the other 14 tests are applied to each individual item of the group. Before performing the item tests, the physical end of group is determined, and the last two bytes of the last frame are set to X'FFFF' to prevent a forward link zero error condition while in the debugger. At the same time, if the calling process was "SELECT" or "FILE-SAVE", the displacement of the next item into the group is determined so that it may be reset if the group is "fixed".

After an error has been "fixed", the group is checked again for errors. If the error was a group linkage error, then the checking begins with the test for illegal forward or backward link. If the error is with a specific item, the checking begins with the first item in the group.

When the GFE handler locates an error, a message similar to:

Format error at Group 34992, Frame 34643, File Base 34901.
Invalid item size during Update.

Action: (E)xit, (I)gnore, (F)ix, (L)ist item, (C)ut item,
(K)ill group, (T)runcate group, (D)ebugger ?

will be displayed. Only users with SYS3 privileges will receive the "Action:" part of the message. All other users will receive the informative part of the message and be forced into the system debugger. The action taken by the GFE handler is determined by the user response to this prompt.

3.9.1 (E)XIT

The (E)xit response terminates the current process and exits to TCL. The format error remains in the file. No data is altered.

3.9.2 (I)GNORE

If the error occurred during a SELECT (verb performing a full file retrieval) or FILE-SAVE operation, the (I)gnore response will ignore the error in the group, and proceed to the next group. Some items will be lost in this case, but the process can continue without fixing the error. If not a SELECT or FILE-SAVE, the GFE handler will indicate that this option is not permitted and will reprompt the user.

3.9.3 (F)IX

The (F)ix response attempts to repair the group. The exact action depends on the type of error (defined below):

3.9.3.1 Invalid Forward Link

The forward link to the invalid frame is cut, and the group is terminated at the last valid frame.

The GFE handler examines the linkages for valid forward link frame number (FRMN), number of next contiguous frames forward linked from this FID (NNCF), backward link frame number (FRMP), and number of previous contiguous frames backward linked to this FID (NPCF). A valid FRMN is either zero or between SYSBASE and MAXFID. A valid FRMP is zero (first frame), or FID of previous frame in chain. Valid NNCF is one less than previous NNCF if previous NNCF not zero, otherwise zero. A valid NPCF is zero (first frame), or one more than previous NPCF. Since both forward and backward links are tested in a single pass through the group, circular linkages are detected. A valid circular linkage requires non-zero FRMN and FRMP for each frame, since the first frame of the group must have a zero FRMP, circular linkage will be detected as a backward link error.

3.9.3.2 Invalid Backward Link

The forward link to the frame which has an invalid backward link is cut, and the group is terminated at the last valid frame.

3.9.3.3 Invalid Item Size

Invalid item size is detected if the item size is less than 6, odd, or greater than 32767 minus linked frame size. It is also detected if either byte in the size field contains a segment mark.

If the segment mark is found within 32767 minus linked frame size bytes, the segment mark is assumed to terminate the item, and the size field is updated accordingly. The ODD bit in the flags field is set appropriately, and a pad byte (X'81') added if necessary to ensure alignment. If a segment mark is not found within 32737 - linked frame size bytes, the item is truncated, and the data between maximum item size (32767 - linked frame size) and the next segment mark is discarded.

3.9.3.4 Embedded Segment Mark

The embedded segment mark is changed to hexadecimal DF. An embedded segment mark is indicated when the item count field points to the item terminating segment mark, but a segment mark exists between the count field and the terminating segment mark.

3.9.3.5 Missing Ending Segment Mark

If the end-of-item segment mark is not found at the expected end of item based on the count field, the item is terminated at either the first segment mark found, or the maximum item size, whichever is smaller. Fixing this error is identical to fixing an item size error.

3.9.3.6 Incorrectly Terminated Item

The last character of the item is overwritten with an attribute mark to force proper termination. The error is not logged in the SYSTEM-ERRORS file. An error is detected if item does not end with the sequence: AM SM (even size) or AM SM X'81' (odd size).

3.9.3.7 Invalid Item Flags

The flags field is corrected based on the format of the item. If the item is in a dictionary level file and the item is a valid D pointer, then the D pointer flag is set.

Invalid item flags error is detected when the item flags are not consistent with the data. The following conditions constitute an error: ITM.MSB reset, ITM.LSB reset, ITM.DDEL and ITM.DUPD both set, ITM.IDEL and ITM.IUPD both set, ITM.BIN and ITM.IND both set, ITM.BIN and invalid hex pointer, ITM.IND and invalid hex pointer, valid D pointer in dictionary level file and ITM.DPTR reset, ITM.DPTR set and invalid D pointer or not in dictionary level file.

When invalid item flags are fixed, bits ITM.MSB and ITM.LSB are set. If bit ITM.IDEL was set, and an item body exists, ITM.IDEL is reset; if no item body exists, ITM.IUPD is reset. If item body is a valid hex fid, followed by a valid hex frame count, ITM.BIN is set, ITM.LNK and ITM.IND are reset. If item body is a valid hex fid, and if the item-id matches a string 4 bytes from the frame pointed to by the fid, then ITM.BIN and ITM.LNK are set and ITM.IND is reset. If the item is in a dictionary level file and if the item appears to be a valid D pointer, ITM.DPTR and ITM.DUPD are set and ITM.DDEL is reset otherwise ITM.DPTR and ITM.DUPD are reset.

3.9.3.8 Illegal Indirect Item Forward Link

The forward link to the invalid frame is cut, and the item is terminated at the last valid frame.

3.9.3.9 Illegal Indirect Item Backward Link

The forward link to the frame which has an invalid backward link is cut, and the item is terminated at the last valid frame.

3.9.3.10 Invalid Indirect Item Size

The indirect item size is corrected assuming the first segment mark found in the indirect item terminates the item and the termination sequence is unconditionally changed to attribute mark, segment mark.

The size of the indirect item is computed and compared to the size stored in the first 4 bytes of the first frame of the indirect item. If there is a difference, an error is detected. Also, if the indirect item is not terminated with the sequence attribute mark, segment mark and error is detected.

3.9.3.11 Illegal Binary Item Forward Link

The forward link to the invalid frame is cut, and the item is terminated at the last valid frame. If the item was a contiguous binary item, it is changed to a linked binary item. Basic object code has been corrupted in this case.

3.9.3.12 Illegal Binary Item Backward Link

The forward link to the frame which has an invalid backward link is cut, and the item is terminated at the last valid frame. If the item was a contiguous binary item, it is changed to a linked binary item. Basic object code has been corrupted in this case.

3.9.3.13 Invalid Item Hash Address

A pointer to a copy of the item is stored in the log entry, and the item is removed from the group. To recover the item, use the GFE.RECOVER program.

Note: This error may be reported for two reasons. First, if the item is not indirect, and the item-id hashes to another group in the file, and second, if the item is indirect, and both copies of the item-id (one in the group and one in the out-of-group copy) are not the same.

3.9.3.14 Duplicate Item-Id

A pointer to a copy of the first item is stored in the log entry, and the first item is removed from the group. The second item remains. To recover the item, use the GFE.RECOVER program.

3.9.3.15 Item-Id longer than 50 characters

A pointer to a copy of the item is stored in the log entry, the item-id is truncated at 50 bytes, the remainder is inserted before the first attribute of the item, and the item is removed from the group. To recover the item, use the GFE.RECOVER program.

3.9.3.16 Item-Id contains system delimiter character

A pointer to a copy of the item is stored in the log entry, the item-id is truncated at the delimiter, the remainder of the item-id is inserted before the first attribute of the item, and the item is removed from the group. To recover the item, use the GFE.RECOVER program.

3.9.4 (L)IST ITEM

The (L)ist item response is valid only for errors which are not linkage errors. The item is displayed on the terminal a page at a time. Pressing any key displays the next page. A control-X will terminate the listing and return to the prompt. Only the first 400 bytes of the item are listed. Upon completion of the listing, the operator is prompted for the desired action to take.

3.9.5 (C)UT ITEM

The (C)ut item response will "cut" the offending item from the group. The rest of the group is shifted, overwriting the offending item.

3.9.6 (K)ILL GROUP

The (K)ill Group response cuts all forward and backward links from the primary file space. The group is marked as empty. All data in the group is lost!

3.9.7 (T)RUNCATE GROUP

The (T)runcate Group response may act in two different manners. First, if the error was an illegal forward or backward link (not in binary or indirect items), the group is truncated at the last valid frame (same as (F)ix). Otherwise, the group will be truncated after the last valid item; that is, the item reported in error and all items stored after that item (physically) are removed from the group.

3.9.8 (D)EBUGGER

The (D)ebugger response enters the system debugger. The operator examines or repairs the GFE manually. The error will be logged before entering the system debugger. If the debugger is exited with a "G" command or line feed, control returns to the calling process which normally retries the desired operation, possibly detecting the same or another GFE.

3.9.9 SPECIAL CASE HANDLING OF GFE'S

There are a number of special cases requiring further explanation of the GFE Handler. They are: Spooler encountering GFE while logging errors, GFE during FILE-SAVE with G-option in effect and handling of next item.

3.9.9.1 SPOOLER GFE DURING ERROR LOGGING

If the spooler encounters a GFE during error logging, the spooler unlocks the group it was attempting to update and continues as if the error had been logged.

3.9.9.2 FILE-SAVE G OPTION

The GFE is logged, and the remainder of the group bypassed the same as the Ignore option of the GFE query message.

3.9.9.3 GFE'S AND THE NEXT ITEM POINTER

If the calling process is "SELECT" or "FILE-SAVE", then the next item pointer is assumed to point to the item currently being accessed. The GFE handler attempts to reset the pointer to the same item when exiting if possible, even though the group may be shifted when errors are fixed. If for some reason, the pointer could not be correctly reset, a warning message is printed, and the pointer is reset to the beginning of the group.

3.10 SYSTEM BACKUP

The format of a FILE-SAVE tape has been expanded to handle large items. "L" segments are exactly like "I" segments, except that the count field consists of eight hexadecimal digits, instead of four. Additionally, a header ("H" segment) exists at the beginning of every physical tape file. The header segment defines the tape type (System, Account, Dict, Data, Updates), the data frame size of the system producing the tape, time and date used to match full file-save tapes with incremental tapes, and option flags. An end-of-file segment ("XF") is placed on the tape after all items from a file have been written to the tape.

The UPDATE-SAVE tape format is similar to the FILE-SAVE format, except "K" segments denote a deleted item. Also, "D" segments have been expanded. A secondary segment type must be specified: "DU" for updated file definition (newly created file), "DD" for deleted file definition (deleted file), "DI" for updated file-definition item body (attribute 5 and above), and "DZ" for unchanged file definition.

The order in which items are saved has been changed. First, the D pointer is saved, then all of the items in the file (except other D pointers), then the D pointer for a sub-file, followed by all of the sub-file's items, etc.

The data stored in the STAT-FILE has been modified to a small degree. SIZE MAX/I and MIN/I refer to the actual item size. RAWSIZE, MX/GP and MN/GP refers to the bytes in the group, not the entire item size for indirect items. DELITMS and DELSIZE contain the number and size of deleted items. INDITMS and INDSIZE contain the number and size of indirect items. FRAMES counts primary file space, linked space, indirect and binary item space. The reel number has been moved from the Item-Id to attribute 28.

Completion of the save process outputs the error message:

[8000] Save complete (nn Files, mmmm Items.)

It is possible for nn or mmmm to be zero or a negative number as files/items flagged as deleted are subtracted from files/items saved.

3.10.1 SAVE

SAVE is a TCL-II verb used to produce formatted system, account, dictionary, and data save tapes. The SAVE verb may also be used to produce an "updates only" system save tape providing an incremental backup facility.

The general form of the SAVE verb is:

SAVE parent-file-name sub-file-name sub-file-name... (Options)

The parent-file-name is the dictionary level file which contains the sub-files which are to be saved. The sub-file-name is the name of the file defined in the parent file dictionary to be saved. For example, to save an account, the parent-file-name would be SYSTEM and the sub-file-name would be the account-name. Multiple sub-file names may be specified in the statement. Some examples of this relationship are:

SAVE SYSTEM SYSTEM (T

This command saves the entire system. The SYSTEM dictionary and all of its items (including all accounts) are saved. This tape could be used to restore the system using an F option at boot time.

SAVE SYSTEM SYSPROG MAIL DEMO (T

This command saves the accounts SYSPROG, MAIL, and DEMO as accounts. Note that this command does not save the SYSTEM dictionary or any of its items. Another way would be to select the accounts to be saved.

SELECT SYSTEM "SYSPROG""MAIL""DEMO"
SAVE SYSTEM (T

This command saves the accounts SYSPROG, MAIL and DEMO as accounts. Note that the SYSTEM dictionary is not saved.

SAVE MD CUSTOMERS (T

This command saves the dictionary of the CUSTOMERS file, and any data sections in the CUSTOMERS dictionary. This tape could be used for archival purposes, or used to move the file and its dictionary to another system.

SAVE DICT CUSTOMERS INACTIVE (T

This command saves the INACTIVE data section of the file CUSTOMERS.

3.10.1.1 OPTIONS

- T** In order to save the specified files to magnetic tape (or other selected media), the T option must be specified in the command. If the T option is specified, a tape label prompt will be issued.
- S** If statistics of the files saved are desired, the S option must be specified. The statistics include the file name, base, and modulo, number of items, number of bytes, number of frames, number of groups containing format errors, and other statistical information. The statistics are accumulated in a sub-file of STAT-FILE. The sub-file name is STAT-FILE*p, where 'p' is the number of the port running the SAVE. Note that it is possible to use the SAVE verb with the S option and without the T option to gather statistical information only. To avoid contention for the STAT-FILE when multiple users are saving data and requesting statistics, a sub-file will be created that concatenates an asterisk and the user's port number to the file. Thus a user on port 2 requesting statistics would cause the sub-file STAT-FILE*2 to be created. The default modulo and separation are determined by the Sysprog verb DEFAULT-MODSEP defined in the SYSTEM VERBS and PROCS section of this manual. In addition, the system utilities LIST-FILE-STATS, FILE-SAVE-STATS, MODSEP and GFE.RECOVER display the sub-files in the STAT-FILE and prompt the user for the port number from which the SAVE was done.
- C** The C option is specified when a backward compatible tape is to be produced. When this option is specified, if the tape is positioned at BOT (beginning of tape), a file mark (EOF) is written before the save begins. Also, data which the old RESTORE processor would reject, such as header segments, linked binary items, and large items (size greater than 32767 minus linked frame size bytes) are not written to the tape.. This option may not be used with the U option, nor for dictionary or data level saves. Does not clear update flags when full system save is executed with this option.
- Normal items stored in DC files are saved as list (CL) items.
Contiguous binary items are saved as CC items.
- U** The U option is used to produce an "updates only" tape. This option may only be used when the entire system is being saved. All files and items which have been updated or deleted are recorded on the tape. This option may only be used after the successful completion of a full system save (without the C or X options). An identifier which matches the corresponding full system save tape is recorded on the updates only tape. This option may not be used in conjunction with the C or X options.
- G** The G option is used to prevent the SAVE process from stopping at all group format errors. If this option is specified and a format error is detected, the error is logged in the SYSTEM-ERRORS file, and the rest of the group is skipped.

- L** The L option is used when only some of the sub-files are to be saved, as when producing a skeleton system tape. When the L option is used, the user will be prompted for the name of a saved list which contains the names of the sub-files to be saved. This list is used in place of files found in the parent dictionary. The list name may be a list stored in the POINTER-FILE, or if the form "(file-name list-name" is used, any file may contain the list. For example,
- SAVE SYSTEM SYSTEM (LT)
- File-Save tape label = ABC CO
List name = MYLIST
- Will save the SYSTEM dictionary, all the items in the SYSTEM dictionary, but only the accounts specified in the saved list MYLIST.
- SAVE SYSTEM SYSTEM (LT)
- File-Save tape label = IPL FILES
List name = (IPL SYSGEN
- Will save the SYSTEM dictionary, all the items in the SYSTEM dictionary, but only the accounts specified in the list SYSGEN in the file IPL.
- Note that when both the T and L options are specified, the Tape Label prompt will be issued first, followed by the List name prompt.
- F** When the F option is specified, all file names are displayed on the terminal (or printer) as they are saved. Otherwise, only account names will be displayed.
- X** When performing a full system save, the X option causes the updated item and deleted item flags (used for the Updates Only Save) not to be cleared. The tape produced could not be used as the basis of an Updates Only Save tape.
- Y** The Y option causes each group of the database accessed to be checked for validity by the Group Format Error Handler. If any format errors are encountered, the Group Format Error Handler will be entered. This option is useful when it is desired to validate the integrity of the database. This option slows the file-save operation.
- P** The P option will direct the file listing to the printer instead of to the terminal.
- R** The R option allows saved files to be re-named on the tape. If this option is specified, the prompt TO: will be issued. The response must be a list of file-names corresponding to the names of the files specified at TCL. This option may be used to re-name accounts, dictionaries or data level files.

3.10.2 FULL SYSTEM SAVE

The entire system is backed up by issuing the following TCL command:

```
SAVE SYSTEM SYSTEM (T)
```

Allowable options are F (to list all file names), S (to accumulate statistical data), G (skip group format errors), X (don't clear update/delete flags), Y (validate database integrity), and C (compatible format). Unless the C or X options are specified, updated and deleted item and file flags are cleared as the data is saved. These flags are used to determine what data are saved on an "updates only" tape. Once the save has begun, it must terminate normally if an "updates only" save is to be created later. If the save is interrupted and terminated (break key and END), or if the system debugger is entered (Abort!), an "updates only" save may not be created. If the C or X options are specified or the T option is absent, the ability to create an "updates only" save is not impaired, whether or not the save terminates normally. However, the full save tape created with the C or X option may not be used as the basis for any "updates only" save.

3.10.3 INCREMENTAL SYSTEM SAVE

An "updates only" save tape may be created only after the successful completion of a full system save (without the C or X option). To create an "updates only" tape, issue the following TCL command:

```
SAVE SYSTEM SYSTEM (TU)
```

Allowable options are F (to list all file names), S (to accumulate statistical data), and G (skip group format errors). Note that the S option will force the save to "visit" each item of each file, thereby slowing its progress. Without the S option, the save processes only updated files, updated groups within those file, and updated items, increasing its speed. This save proceeds more rapidly than the full system save since only updated or deleted items and files are recorded on the tape. If a file has been deleted, the displayed name (printed on the terminal) will be shown with a hyphen (-) instead of a file number, and only the file descriptor segment is saved. This tape is synchronized to the most recent full system save made (without the C or X option). If a restore is required, the full save is restored first, then the matching "updates only" save is restored. Note that only the most recent "updates only" save tape is required!

3.10.4 FILE-SAVE

FILE-SAVE executes a Basic program in the SYSPROG account that saves the entire system to tape, with the exception of those accounts and files that have been DX'ed. The SAVE verb is automatically executed by the program. If a GFE is encountered during the SAVE then it will be logged in the SYSTEM-ERRORS file and the rest of the group will be skipped. File-stats are always saved and T-DUMP'ed on the end of the save tape. The user may enter FILE-SAVE at TCL when logged on to the SYSPROG account or logon to FILE-SAVE which is a Q-pointer to the SYSPROG account. Either way the FILE-SAVE program will prompt the user for information that is used as options to the SAVE verb. Entering an "X" at any prompt will log the user off. Once FILE-SAVE is entered, it will prompt:

(M)ag tape, (C)artridge tape or (F)loppy disk (M,C,F,X)?

Enter block-size <4096>:

If <cr> entered, then default is 4096

If M entered to previous prompt then must be > 2 or <= 16384.

If M entered to previous prompt then user is queried re: 1/2" tape.

Do you have multiple 1/2" tape units (Y/<N>/X)?

If X, executes OFF.

If N, executes SET-MT block-size.

If Y, T-STATUS verb is executed and the query:

(C)ascade the save or use a (S)ingle drive (C,<S>,X)?

If X, executes OFF.

If S, then query:

Enter ONLY tape unit number (0-3) - <CR> to end:

Using unit n OK? (<Y>,N)?

If N, return to Cascade/Single drive query.

If Y, go to printer listing query.

If C, then query:

Enter FIRST tape unit number (0-3) - <CR> to end:

Enter NEXT tape unit number (0-3) - <CR> to end:

Enter NEXT tape unit number (0-3) - <CR> to end:

Cascade sequencing: Units n,n,n,n OK? (<Y>,N)?

If N, return to Cascade/Single drive query.

If Y, tape units are attached and queries:

If C entered to previous prompt then block-size must be 512 or 4096.

If C entered to previous prompt then the user is queried:

Desired tape format (S)MA or (N)ormal: (S,<N>,X)?

If X, executes OFF.

If N, executes SET-CT block size.

If S, executes SET-CT 8192 regardless of block-size specified.

If F entered to previous prompt, the user is queried:

Desired diskette density: (H)igh-1.2MB or (L)ow-360KB: (<H>,L,X)?

If X, executes OFF.

If H, executes SET-FD.

If L, executes SET-FD (X

Do you want a listing to the printer (Y,<N>)?

If Y, then P option added to SAVE verb

Do you want a file-stat report (Y,<N>)?

If Y, file-stats are output to printer at completion of save.

Do you want to display filenames (<Y>,N)?

If Y, then F option added to SAVE verb

Enter time to start save or <RETURN>

Format is: hh:mm or hh:mm:ss

If <RETURN>, then start save immediately.

If time is entered, then port will SLEEP until time to start.

3.10.5 UPDATE-SAVE

UPDATE-SAVE executes a Basic program in the SYSPROG account that produces an "updates only" tape. It can be used only after a complete system save was successfully completed. All files and items which have been updated or deleted are recorded on the tape. The SAVE verb is automatically executed by the program. If a GFE is encountered during the SAVE then it will be logged in the SYSTEM-ERRORS file and the rest of the group will be skipped. The user may enter UPDATE-SAVE at TCL when logged on to the SYSPROG account or logon to UPDATE-SAVE which is a Q-pointer to the SYSPROG account. Either way the UPDATE-SAVE program will prompt the user for information that is used as options to the SAVE verb. Entering an "X" at any prompt will log the user off. Once UPDATE-SAVE is entered, it will prompt the user in exactly the same way as prompted by FILE-SAVE:

(M)ag tape, (C)artridge tape, or (F)loppy Disk (M,C,F,X)?

3.10.6 ACCOUNT SAVES

Accounts may be saved using the SAVE verb, as described above. However, since a user may not have access to the SYSTEM file, the ACCOUNT-SAVE verb may be used to save single accounts. The general form of the ACCOUNT-SAVE verb is:

ACCOUNT-SAVE account-name (options)

The specified account-name must exist in the SYSTEM dictionary. The T and F options are standard options included in the verb. Options S, R, P, G, Y and C, as described for the SAVE verb, may be selected. Does not clear update flags in the account.

3.10.7 ERROR HANDLING & LOGGING

Any errors (other than group format errors) encountered during the save process will be reported and logged into the SAVELOG file. Also, the names of all accounts saved will be logged in the SAVELOG file. The item-id used for the log item is current system date (internal format) concatenated with an asterisk (*), concatenated with the current system time (internal format, milliseconds), concatenated with an asterisk (*), concatenated with the port number doing the save. Also, an item named SAVELOG is written into the SAVELOG file whose first attribute contains the item-id of the most recent log entry. If the SAVELOG file cannot be opened, the log item will not be written. This file should be cleared out periodically. The default modulo and separation for the SAVELOG file is set by the SYSPROG verb DEFAULT-MODSEP, described in the SYSTEM VERBS and PROCS section of this manual.

3.10.8 SELECTIVE SAVE AND THE T-DUMP VERB

An option has been added to T-DUMP to dump a compatible tape for moving data to older systems. The "C" option will dump a compatible tape, converting items in a DC type file to list (CL) items. Linked binary items and non-DC type items larger than 32767 minus linked frame size bytes may not be dumped to a compatible tape (a warning will be printed if this is attempted). No SAVELOG entries are initiated.

3.11 SYSTEM RESTORE

System restores may be initiated in several ways: 1) a full system restore may be initiated at boot time by using the F option; 2) a verb is provided to restore the entire system from a running process, and; 3) accounts, dictionaries, and data level files may be restored by using the TCL-II verb RESTORE.

When restoring from a tape made under the old file structure, list (CL) items are automatically converted into normal items. The "Y" option may be used to inhibit this conversion.

3.11.1 BOOT TIME FILE RESTORE

The F option at boot time invokes the restore process after the system boot has completed. The user will be prompted to select the restore device:

(M)ag tape, (C)artridge tape, or (D)isk cartridge or (F)loppy disk?

If (C)artridge tape is selected, the cartridge format is selected:

(N)ormal format, (S)MA format, or (Q) restart media select?

If N, then query

(A) QIC-24, (AL) QIC-24 long, or (Q) restart media select?

If (F)loppy disk is selected, then

FD density (H)igh - 1.2MB, (L)ow - 360KB, or (Q) restart media select?

The system will check for a valid device selection, and prompt:

Current disk file system will be overlaid: (C)ontinue or (Q)uit? (C/<Q>):
Mount data tape/diskette and press RETURN

The selected device will be read, and the type of tape mounted will be checked. If the tape is not a full file-save tape (made with or without the C option), the following message will be displayed:

Incorrect tape format!

(F)orward space to next file

(R)ewind tape and try again

(M)ount different tape or change device

(Q)uit (will jump into the COLDSTART Proc if a system exists on disk)

Enter selection :

Once the correct tape format has been identified, the restore process will begin. After the full file-save tape has been restored, and if the save tape was created without the C or X options, the user will be prompted:

Restore updates (Y/N) ?

If the response is Y (yes), the user will be prompted to select the restore device, then:

Mount update tape/disk and press RETURN

After the tape format has been verified and the update tape identifier verified against the full file-save tape identifier, the updates will be restored.

Upon completion of the restore, the COLD-START procedure will be executed.

Note that during the restore process, all terminals will be inhibited from logging on to the system.

3.11.2 THE :FILELOAD VERB

The :FILELOAD verb may be used by a running process (with SYS3 privileges) to initiate a full system restore from TCL. In order to execute this verb, all other users must be logged off. The restore begins from the currently selected tape device, and proceeds just as described above, except for device selection and "Mount data tape" prompt.

3.11.3 THE RESTORE VERB

RESTORE is a TCL-II verb which is used to restore specified accounts, dictionaries and data level files from a save tape. The general form is:

RESTORE parent-file-name sub-file-name sub-file-name... (Options)

where parent-file-name is the name of the dictionary into which the sub-files will be restored, and sub-file-name is the name the restored file is to be given (this may be different than the name it was saved as). Depending on the level of the first file on the save tape, one or more of the following prompts will be issued:

Account name on tape: For full-system save tapes or account level save tapes, for example:

SAVE SYSTEM SYSTEM or SAVE SYSTEM account-name
FILE-SAVE or ACCOUNT-SAVE account-name

Dictionary name on tape: For dictionary level save tapes, for example:

SAVE MD file-name or SAVE SYSTEM,account-name file-name

File name on tape: For data level save tapes, for example:

```
SAVE DICT file-name data-name
SAVE DICT SYSTEM,account-name,file-name data-name
```

The name (or names) entered in the final response correspond to the list of sub-file-names entered in the TCL command. Any missing names (null is an acceptable response to the last prompt) will assume the corresponding sub-file-name specified in the TCL command line.

An asterisk may be used in place of the list of sub-file-names in the TCL command. In this case, any file found on the tape which does not exist in the parent dictionary will be restored.

If the tape restored was a full save tape made without the C or X options, then upon completion of the restore, the user will be asked if updates are to be restored. In this case, only updates to the files which have been restored by this TCL command will be restored.

Completion of the restore process outputs the error message:

```
[8020] Restore complete (nn Files, mmmm Items.)
```

It is possible for nn or mmmm to be zero or a negative number as files/items flagged as deleted are subtracted from files/items restored.

Note that the RESTORE verb honors the retrieval and update restrictions of the file on the tape. That is, you may not restore an account, dictionary or data file which you are not authorized to access.

Note: T-DUMP tapes cannot be loaded using the RESTORE verb as it does not contain sufficient information on the tape.

3.11.4 THE ACCOUNT-RESTORE VERB

The ACCOUNT-RESTORE verb may be used to restore one or more accounts from a file-save tape. This verb is provided to allow users who do not have access to the SYSTEM file to restore accounts. The account name (or names) on the file-save tape must be specified.

3.11.5 OPTIONS

The following options may be used with the verbs :FILELOAD, RESTORE and ACCOUNT-RESTORE unless a restriction is indicated in the option explanation.

- N The N option inhibits the automatic re-sizing of files when restoring a tape made on a machine with a different data frame size. The default is to automatically adjust the file modulo based on the change in data frame size. If the save tape was made under the old format or with the C option, no automatic re-sizing takes place anyway. Also, the N option inhibits re-sizing of files based on the reallocation specification in the file definition item (attribute 13).
- R The R option enables automatic re-sizing of files when restoring a tape made under the old format or with the C option. If this option is specified, the user will be prompted:

Tape producer data frame size =

The frame size must be a multiple of 500. This option is only available with the RESTORE and ACCOUNT-RESTORE verbs.
- P Output list to printer.
- S The S option suppresses the display of the list of file names being searched for on the tape.
- Y The Y option inhibits the conversion of "CL" items on old format tapes into normal items, restoring them as linked binary items instead.
- V The V option is used to perform a "verify" operation, not a restore. This option causes the tape to be read as if it were being restored, but discards the data after reading it. This option will detect defective D-pointers, invalid item sizes, unrecognizable tape segments, and other tape format errors. See the verb VERIFY-SAVE below.
- X The X option may be used with RESTORE or :FILELOAD when restoring from a full file-save tape, and when it is not desired to restore a corresponding update tape. This option by-passes the "Restore updates?" prompt, allowing for automation of save/verify/restore procedures.
- B The B option is used when restoring files from a tape made under Open Architecture releases of the Pick System. Item size errors are ignored when using the B option, and binary items requiring more than 255 frames are skipped.

- D The D option, used with the V option, verifies that the data on the tape is the same as the data on the system. This option actually opens the files as they are encountered on the tape, and as each item is read from the tape, it is compared to the item on the system. This option may be used to verify the integrity of the save tape. Note: data read from the tape is compared to data stored on the system, not vice-versa. See the verb VERIFY-SAVE-DATA below.

To verify a full save tape, use the following command:

```
:FILELOAD (VD
```

To verify an account save tape, use the following command:

```
RESTORE SYSTEM account.name (VD
```

answering the "Account name on tape" prompt with a carriage return.

WARNING: Forgetting to include the V and D options with :FILELOAD and ignoring the "Continue or Quit" prompt, will destroy current file data. Therefore, it is suggested that the V and D options be used only in Procs for specific applications and that the verbs VERIFY-SAVE and VERIFY-SAVE-DATA be used at the TCL prompt.

3.11.6 ADD-ACCOUNTS

Automatically adds new accounts from a FILE-SAVE tape or multi ACCOUNT-SAVE tape, that do not already exist on the system.

(M)ag tape, (C)artridge tape or (F)loppy disk (M,C,F,X)?

If M, then query:

Do you have multiple 1/2" tape units (Y/<N>/X)?

If X, returns to TCL.

If N, SET-MT 4096 is executed and restore begins.

If Y, T-STATUS verb is executed and the query:

Would you like to (C)ascade the restore or use a (S)ingle drive
(C,<S>,X)?

If X, executes OFF.

If S, then query:

Enter ONLY tape unit number (0-3) - <CR> to end:

Using unit n OK? (<Y>,N)?

If N, returns to Cascade/Single driver query.

If Y, begins restore.

If C, then query:

Enter FIRST tape unit number (0-3) - <CR> to end:

Enter NEXT tape unit number (0-3) - <CR> to end:

Enter NEXT tape unit number (0-3) - <CR> to end:

Cascade sequencing: Units n,n,n,n OK? (<Y>,N)?

If N, returns to Cascade/Single drive query.

If Y, tape units are attached and starts restore.

If C, then query:

Desired tape format (S)MA or (N)ormal: (S,<N>,X)?

If S, executes T-FORMAT (S and SET-CT 8192 and starts restore.

If N, then query:

Please specify tape length: (A)-QIC 24, (AL)-QIC 24 long:

Then executes T-ASSIGN and SET-CT 4096 and starts the restore.

If F, then query:

Desired diskette density: (H)igh-1.2MB or (L)ow-360KB: (<H>,L,X)?

If X, executes OFF.

If H, executes SET-FD.

If L, executes SET-FD (X

If X, returns to TCL.

3.11.7 VERIFY-SAVE VERB

Verifies the currently attached tape as if it were being restored to disk, but discards the data after reading it. This operation will detect defective D-pointers, invalid item sizes, unrecognizable tape segments, and other tape format errors. Verb format is:

VERIFY-SAVE {account-name (A)}

Option: A Will prompt Account name on tape:

User may enter a return or an account-name, this overrides account-name given at TCL command.

3.11.8 VERIFY-SAVE-DATA VERB

Verifies that the data on the tape is the same as the data on the system disk(s). The files on the disk are opened as they are encountered on the tape, and as each item is read from the tape, it is compared to the item on the system. Note any items that change on disk since the save will be reported as an error. Therefore, for the verification to be useful, the system should not be in use while the verification is in progress. Verb format is:

VERIFY-SAVE-DATA {account-name (A)}

Option: A Will prompt Account name on tape:

User may enter a return if account-name on disk is the same as account-name on tape, or enter account-name on tape.

3.11.9 ERROR HANDLING DURING RESTORE

Once an F-level restore is initiated via the boot and after the SYSTEM file has been created, all errors are logged in the RESTORE-ERRORS account (the account will be created by the restore process and may not exist prior to the restore). An item, RESTORE-ERRORS, will be created in the account level file RESTORE-ERRORS; this list contains all of the error messages reported during the restore. This item is written after the restore is completed. All other restore processes initiated will log errors in an account called RESTORE-ERRORS*p, where 'p' is the port number that initiated the restore.

Additionally, any item on the tape whose file cannot be identified (in the case of damaged file definition segments or missing reels) will be stored in a file contained in the RESTORE-ERRORS{*p} account. The default modulo and separation for RESTORE-ERRORS{*p} is defined by the Sysprog verb DEFAULT-MODSEP described in the SYSTEM VERBS and PROCS section of this manual. The dictionary where the item is filed depends on the type of restore in process: RESTORE if not processing an updates-only tape, UPDATE if processing an updates-only tape, and the item is new or updated, and DELETE if processing an updates-only tape, and the item is deleted. The data level file name is simply the file number (from the tape) of the file where the item belongs.

Thus, after a restore, it is important to examine the contents of the RESTORE-ERRORS{*p} account (if it exists) to determine if the restore was successful, i.e., LIST-RESTORE-ERRORS.

If the restore process should enter the system debugger during the restore, it is generally possible to continue the restore with the next segment from the save tape. Entering "END" in response to the debug (!) prompt, should prompt:

(C)ontinue with next segment or (Q)uit ?

Entering C should continue the restore with minimal data loss.

3.11.9.1 HANDLING OF ERRORS NOT STORED IN THE RESTORE-ERRORS ACCOUNT

If no separation is given on the tape being restored, separation is forced to 1 and does not go through error processing.

If modulo on tape is greater than 1,048,573, less than or equal to zero, or there is not enough disk space, the user is prompted for a new modulo and the separation is set to 1. The old modulo is displayed for the users information.

When the user is prompted for modulo, the number is forced to odd and validated. If the number fails, the user is prompted again.

When restoring a tape from a system with a different frame size and the system calculates the modulo (R-option), the modulo is set to a number that is odd and not divisible by five.

If attribute 13 of the file D-pointer is (n), then n=MODULO and SEPARATION=1. If it is (n,m), then n=MODULO, m= SEPARATION. If attribute 13 is empty, does not begin with "(", or specified modulo and separation are the same as the current D-pointer, no action is taken.

3.11.10 SELECTIVE RESTORE AND THE T-LOAD VERB

The T-LOAD verb will accept a "save" format tape as input. If a tape is determined to be a "save" format tape, one or more of the following prompts will be issued:

Account name on tape: For full-system save tapes or account level save tapes, for example:

SAVE SYSTEM SYSTEM or SAVE SYSTEM account-name
FILE-SAVE or ACCOUNT-SAVE account-name

Dictionary name on tape: For dictionary level save tapes, for example:

SAVE MD file-name or SAVE SYSTEM,account-name file-name

File name on tape: For data level save tapes, for example:

SAVE DICT file-name data-name
SAVE DICT SYSTEM,account-name,file-name data-name

Normal ACCESS selection criteria may be used to specify items which are to be loaded from the tape.

T-LOAD may be used with either full or updates-only save tapes as well as account, dictionary, and data file saves. If an updates-only save tape is used, only new or updated items will be restored. Deleted items will not be deleted from the file. RESTORE must be used to delete items from a file.

When using T-LOAD with a "save" format tape, the tape must be positioned at the beginning of a tape file before executing the verb, unless the A option is specified. If data is being restored from an account which begins on a reel other than the first, then a T-FWD should be issued to position the tape at the beginning of the next tape file prior to executing the T-LOAD. If this is not done, then the tape will probably not be identified as a "save" format tape, and erroneous data may be loaded.

Note that the T-LOAD verb used with a "save" format tape honors the retrieval and update restrictions of the file on the tape. That is, you may not load from account, dictionary or data file which you are not authorized to access. Note: SEL-RESTORE is the same as T-LOAD command. It is converted internally to the T-LOAD command.

3.11.10.1 T-LOAD OPTIONS

- S The S option will suppress the list of file names displayed during the search for the file to be restored.
- N The N option will search for a specified file number rather than a file name. If the N option is specified, the following prompt will be issued:

File number on tape :

- A The A option will cause T-LOAD to assume a save format tape, at a position other than BOT (beginning of tape) or EOF (end of file). Since the logical location cannot easily be determined, the following prompt will be issued:

Level: 0-System 1-Account 2-Dictionary 3-Data File :

The levels are interpreted as follows: 0 - tape is positioned somewhere within the SYSTEM file. The first file prompt will be for Account name. 1 - Tape is positioned within the desired account. The first file prompt will be for Dictionary name. 2 - tape is positioned within the desired dictionary. The first file prompt will be for File name. 3 - Tape is positioned within the desired data file. No file name prompt will be issued. The A option is useful when T-LOADing from volumes other than the first volume in a multi-volume set, or when performing multiple T-LOADs from a single account.

- C The C option will allow old ULTIMATE T-DUMPS to be read correctly.
- P Send output to printer.

ACCESS 4

4.1 ACCESS PRE-PROCESSOR

The ACCESS Pre-Processor enhances the Pick operating system Access query language. The pre-processor provides two major functions. First, it adds the ability to use and define phrases which contain parts of Access statements. Second, it eliminates the requirement that explicit values be enclosed in quotation marks, except to resolve ambiguities.

The ACCESS Pre-Processor is enabled by changing the Access verb definition in the master dictionary for those verbs where pre-processing is desirable. The pre-processor scans the input sentence, expands phrases and determines which words are values, then passes the sentence to the standard Access compiler.

In normal operation, the ACCESS Pre-Processor processes the input statement before the Access compiler. An option (Z) may be used to bypass the pre-processing.

4.1.1 PHRASES

Phrases must be defined in the dictionary of the file for which the phrase will be used (or in the account's master dictionary). Line one of the phrase definition must be an H. Line two of the phrase definition contains the words which make up the phrase.

Phrases may contain any legal Access words and constructions except verbs and file names. This includes output attributes, headings, footings, selection criteria, modifiers (ID-SUPP, DBL-SPC, etc.), even explicit item-id's. Phrases may contain other phrases. Circular phrase references are not permitted and result in an error message.

It is possible to store a collection of phrases in a file and reference them by means of a "remote" phrase definition. A remote phrase definition contains an R on line 1, the phrase file name on line 2, and the phrase item-id on line 3. The item in the remote phrase file must be a phrase definition (H on line 1), and the file and item must exist.

A special phrase may exist in the file dictionary (or in the account's master dictionary if the DICT modifier is specified) which defines the default output attribute list when no explicit attribute names have been included for output. The item-id of this special phrase is the at sign followed by the file name (@filename) or, optionally, just the at sign (@). This is used in place of the conventional numbered output attributes (1,2,3...). If the @filename phrase is missing, then the @ phrase is assumed to define the default output attributes. Using this method, dictionaries which define several data files may use different output attribute list phrases. Also, if Q-pointer(s) exist for the file, the @filename for the actual file will be used to determine the output attributes.

If line 1 of the @ phrase is an H or R, all numerics separated by spaces and not enclosed in quotes are treated as values. That is, if a user's master dictionary contained the @ phrase with an H or R on line 1, the statement "LIST MD 1 2 3" would list only items '1' '2' and '3' using the default output attribute list (from the @ phrase).

Alternatively, if line 1 of the @ phrase contained an HZ or RZ, numbered attribute names are allowed and will only be considered as values if they do not exist as attribute definitions. For example, if a user's master dictionary contained the @ phrase with an HZ or RZ on line 1, the statement "LIST MD 1 2 3" would list the entire master dictionary displaying attributes 1, 2, and 3 (provided that 1, 2, and 3 are attribute definitions in the master dictionary).

4.1.2 PHRASE USAGE EXAMPLES

By definition a phrase may be any syntactically complete portion of an ACCESS statement except verbs and file names. For example, a phrase named ADDRESS.LIST in the file CLIENT could be used to display NAME and ADDRESS:

```
ADDRESS.LIST
001 H
002 NAME STREET CITY STATE ZIP
```

Once defined in the DICT of the file CLIENT this phrase may be used in an ACCESS query such as:

```
>SORT CLIENT WITH STATUS = ACTIVE BY ZIP ADDRESS.LIST LPTR
```

This statement would produce a report of clients defined as ACTIVE in the data file and using the phrase definition ADDRESS.LIST as the output specifications. Please note that the selection value ACTIVE is not enclosed in quote marks; another feature of the ACCESS Pre-Processor.

A HEADING statement could have been included in the phrase ADDRESS.LIST, however, a more effective method may be the use of a "remote phrase." For example a remote phrase file called STAFF could be created containing HEADING phrases for each staff member. To use remote phrases an 'R' type phrase is required in the DICT of the file CLIENT, as well as, a phrase in the data file STAFF.

DICT of CLIENT

JIM
001 R
002 STAFF
003 JIM.JONES

DATA of STAFF

JIM.JONES
001 H
002 HEADING "Page 'P' PREPARED FOR JIM JONES 'CTLL'"

Now the ACCESS statement:

>SORT CLIENT WITH STATUS = ACTIVE BY ZIP JIM ADDRESS.LIST LPTR

Would produce the same report as in the previous example, however, the 'R' type phrase 'JIM' in the DICT of CLIENT will capture the HEADING from the STAFF file. Multiple remote phrase files may be used (i.e. FROM could be a file of desired FOOTINGS).

NOTE: If a code of 'A' was stored in the data file of CLIENT as the method of indicating a client as ACTIVE the above example would require the statement be entered as:

>SORT CLIENT WITH STATUS = "A" BY ZIP JIM ADDRESS.LIST LPTR

This is due to the fact that "i" is defined in the MASTER DICTIONARY as a throw away CONNECTIVE and the quote oarks are needed to resolve any ambiguity. This would also apply to other words which are defined in the DICT of the file being processed or in the MD.

Another type of phrase which may be used is the @filename phrase. The @filename phrase is used to define the default output attributes when no output specifications are included in the ACCESS statement. For example, the @filename phrase for the file CLIENT would be created in the DICT of CLIENT under the name @CLIENT:

@CLIENT
001 H
002 NAME CITY PHONE CONTACT

Now if the LIST or SORT verbs are used to query the file CLIENT the attributes NAME, CITY, PHONE and CONTACT will be the output for statements such as:

```
>LIST CLIENT HEADING "Page      'P'      CLIENT DEFAULT LIST'CTL'"
```

or

```
>SORT CLIENT BY NAME WITH TYPE = VAR
```

The above statements contain no output specifications, therefore, the @CLIENT phrase is used to determine the output.

If the statement was entered as:

```
>SORT CLIENT BY NAME NAME STREET CITY STATE ZIP
```

Then the default output attributes defined in @CLIENT would be ignored and the output list requested would be used.

NOTE: If line 001 of @CLIENT is defined as HZ or RZ (remote) then attribute numbers (i.e. 1 2 5 7 8) may be used on line 002 to extract the data in those attribute AMC id's, provided that those AMC numbers are defined in the file dictionary.

The @filename phrase may also be just the @ sign and the filename will be assumed. Also, if Q-pointer(s) exist for the file, the @filename for the actual file will be used to determine the output attributes. For example, if CUST is a Q-pointer to the file CLIENT and the ACCESS statement is entered:

```
>LIST CUST
```

Then the @CLIENT default attribute phrase would be used. In other words, @CUST is not required and is not valid!

A special use of the @filename phrase occurs in the case when multiple data sections are defined in a dictionary. For example, if in the DICT of our CLIENT file a data section for ACTIVE and another for INNACTIVE are defined then a separate @filename phrase may also be defined for each data section.

```
DICT of CLIENT
  @ACTIVE
001 H
002 NAME CITY PHONE
```

```
DICT of CLIENT
  @INNACTIVE
001 H
002 NAME CITY STATE ZIP
```


Now if the CLIENT file is queried:

```
>LIST CLIENT,ACTIVE
```

the pre-processor will use the @ACTIVE filename phrase to determine the default output attributes. And if the file is queried:

```
>LIST CLIENT,INNACTIVE
```

the pre-processor will use the @INNACTIVE filename phrase to determine the default output attributes.

4.1.3 QUOTATION MARKS

With the ACCESS Pre-Processor, quotation marks are not required to distinguish values from other parts of an Access sentence. Any word which is not in the file dictionary (or M/DICT), is assumed to be a value, provided that it makes sense to use a value in that part of the sentence. If it does not make sense, then an error message is printed and the sentence is aborted.

It is still necessary to use quotation marks when there is an ambiguity in the meaning of a word. For example,

```
>LIST MD D/CODE D/CODE
```

could mean to list the item 'D/CODE' showing the attribute D/CODE, or it could mean list all items showing the attribute D/CODE in two columns. In this case, quotation marks are required to distinguish between the first and second meaning. Also, if a value contains blanks, quotation marks are required to distinguish between a single value with embedded blanks, and multiple values separated by blanks.

Headings, footings, BREAK-ON options and GRAND-TOTAL headings must still be enclosed in quotation marks.

The Access verbs CHECK-SUM, STAT, and SUM do not use the pre-processor, therefore, explicit item-ids should be enclosed in quotes.

4.1.4 OPTIONS

There are two new options for use with the ACCESS Pre-Processor. The first is "Y", which causes a message to be displayed whenever the pre-processor assumes an undefined word is a value. The message shows the undefined word and the context it is assumed to be used in. The processed sentence is then displayed on the screen before entering the Access compiler. The terminal will wait for a key to be pressed before proceeding. If a control-X is entered, then the sentence will be aborted.

```
>SORT ABC BY AA WITH AA = ABC (Y
[2393] Assumed 'ABC' is a value used as selection criteria on attribute AA.
SORT ABC BY AA WITH AA = "ABC"
```

The second option, "Z", is used to override the ACCESS Pre-Processor. If the Z option is present, the pre-processor is bypassed, and the Access compiler is entered directly.

```
>SORT ABC BY AA WITH AA = ABC (Z
[5] The word "ABC" is illegal.
```

4.1.5 ACCESS STATEMENT EXAMPLES WITHOUT QUOTATION MARKS

The following examples show typical ACCESS statements in which quote marks are not required. Each of these examples will utilize the 'Y' option which will display a message defining how the pre-processor interprets the statement.

```
>SORT SALES BY DATE WITH DATE >= 1/1/87 AND <= 12/31/87 (Y
[2393] Assumed '1/1/87' is a value used as selection
        criteria on attribute 'DATE'.
[2393] Assumed '12/31/87' is a value used as selection
        criteria on attribute 'DATE'
SORT SALES BY DATE WITH DATE >= "1/1/87" AND <= "12/31/87"
```

In the example above, the ACCESS Pre-Processor identified one selection clause with two values and then showed the ACCESS statement complete with quote marks.

```
>LIST ORDERS 1234 1235 (Y
[2391] Assumed '1234' is an item-id, or modifies the item-id.
[2391] Assumed '1235' is an item-id, or modifies the item-id.
LIST ORDERS '1234' '1235'
```

In the example above, the pre-processor correctly identified two item-ids and then showed the ACCESS statement including quote marks.

```
>SORT G/L BY-EXP BALANCE > 1000.00 (Y
[2395] '1000.00' is a value used to modify the sort clause
        on attribute 'BALANCE'.
SORT G/L BY-EXP BALANCE > "1000.00"
```

The example above shows that the pre-processor is able to identify that 1000.00 is a value used to modify the BY-EXP sort clause.

```
>LIST G/L BALANCE > 1000.00 (Y
[2392] '1000.00' is a print limiter which modifies 'BALANCE'.
LIST G/L BALANCE > "1000.00"
```

The example above shows that the ACCESS Pre-Processor identified the value '1000.00' as a print limiter modifying the output attribute 'BALANCE'. The statement will list all items but only values greater than 1000.00 will be printed for the column BALANCE. Note that BALANCE is an output attribute. If the word 'WITH' was included in this statement then 'BALANCE > 1000.00' would be a selection criterion.

4.2 ACCESS "ANY" DEFINITION

The master dictionary entry "ANY" is now an attribute definition that gives Access the ability to search entire items on a value by value, attribute by attribute basis. Each value is compared to the selection criteria and if the value passes the criteria, then the item is selected for further processing. This feature is useful for searching non-attribute oriented files for specific words or phrases, such as word processor documents, source programs, etc.

The word ANY may be used to mean ANY attribute, value, or sub-value in the item. ANY may be used in selection criteria (WITH, IF), or specified for output. If ANY is specified for output, it must also be specified in selection criteria. The output of the ANY attribute definition will be any sub-value, value or attribute which passes the selection criteria.

The ANY attribute definition may be used with all Access verbs: LIST, SORT, SELECT, T-DUMP, T-LOAD, etc.

4.3 CONVERSION AND CORRELATIVE CODES

The following conversion codes may be used in Access by placing them in attribute definition items (lines 7 or 8).

4.3.1 Center (XC)

The CENTER conversion is used to produce a centered field on an Access report. The conversion is specified on line 7, and may contain a column width specification. If no column width is specified, the column width for the attribute is assumed. To use the CENTER conversion, place XC (or XCn, where n is the column width to center in), on line 7 of the attribute definition item which requires centered output.

4.3.2 Change Conversion Direction (XT, XI, XO)

The direction of conversion (input / output) may be changed (toggled), forced to input, or forced to output by using one of three conversion codes.

- XT - Toggle conversion direction.
- XI - Force to Input Convert.
- XO - Force to Output Convert.

These conversions are especially useful in 'F' and 'A' correlatives when some intermediate value is in output format, and more processing is required, but the internal format is easier to deal with. They are also useful in the case where a file contains numeric data with decimal points, and standard masks are desired for output. The MR conversion may be used as an input conversion to remove the decimal points and scale the number to the proper range.

For example, suppose an attribute contains data which may include a decimal point (23.4), and decimal points are to be aligned with two places to the right of the decimal point. The following conversions could be used in the attribute definition item:

```
007 MR2
008 XI]MR2]XO
```

In this example, the attribute could be TOTALed, used in A correlatives, etc., and would return the correct value.

Example: Changing Conversion Direction

Attribute defining items in DICT TEST

A1	CA1	A2	CA2
001 A	001 A	001 A	001 A
002 1	002 1	002 2	002 2
003 RAW]ATT1	003 CONVERTED]ATT1	003 RAW]ATT2	003 CONVERTED]ATT2
004	004	004	004
005	005	005	005
006	006	006	006
007	007 MR2	007	007 MR2
008	008 XT]MR2]XT	008	008 XI]MR2]XO
009 R	009 R	009 R	009 R
010 5	010 9	010 5	010 9

SUM	CS
001 S	001 S
002 0	002 0
003 RAW]SUM	003 CONVERTED]SUM
004	004
005	005
006	006
007	007 MR2
008 A1 + 2	008 A1(XI]MR2) + 2(XI]MR2)(XO)
009 R	009 R
010 5	010 9

Items in data section of TEST

ITEM.1	ITEM.2	ITEM.3
001 1.11	001 2.22	001 4.44
002 3.33	002 3.33	002 3.33

>SORT TEST TOTAL A1 TOTAL CA1 TOTAL A2 TOTAL CA2 TOTAL SUM TOTAL CS

TEST..	RAW.. ATT1	CONVERTED ATT1	RAW.. ATT2	CONVERTED ATT2	RAW.. SUM	CONVERTED SUM
ITEM.1	1.11	1.11	3.33	3.33	4	4.44
ITEM.2	2.22	2.22	3.33	3.33	5	5.55
ITEM.3	4.44	4.44	3.33	3.33	7	7.77
***	7	7.77	9	9.99	16	17.76

4.3.3 Running Sub-Total (XR)

The Running Sub-Total conversion produces a running sub-total of an output column on an Access report. The attribute definition must have XR specified as a conversion (line 7), never as a correlative. In order to get a running total, the attribute containing the XR conversion must be TOTALed; if not, the conversion has no effect.

The sub-total may be restricted to any number of control break levels. If, for example, the running total was only to be shown for the first control break level, the conversion could be specified as XR1 to indicate only one break level. The running total would start over from 0 after the first and higher break levels.

Other output conversions are possible in addition to the running total conversion; just separate them with value marks.

4.3.4 Function Processor, F Code:

The 'F' code processor uses a 15 element post-fix push-down stack for storing values. An operation specified by an F-code operates on the last one or two entries pushed onto the stack. Entries are removed from the stack as they are used in the operation. The results of the operation is pushed onto the stack. This continues for each operator until the entire F-code is processed. The final result is then the value on the top of the stack. The form is:

```
FS;elm{ ;elm. . .}
```

Note that this form of the 'F' code processor differs from previous implementations in the use of the S to designate standard form. This standard form includes all ordered binary operations in classic reverse Polish ordering, including the comparison operations.

where 'elm' may be any of the following:

amc{R{R}} A numeric Attribute Mark Count specifying the element to be pushed onto the stack. If the AMC is followed by R, it specifies that the first value of an attribute is to be used repeatedly when evaluating with a multi-valued attribute. If the second R is present, it specifies that the first subvalue of a value is to be used repeatedly.

Cn A capital 'C' followed by a string, specifies that a string is to be pushed onto the stack. The string is ended by the next semicolon.

D specifies that the current date is to be pushed onto the stack (internal format).

literal The literal string enclosed in either single or double quotes is pushed onto the top stack entry.

T specifies that the current time is to be pushed onto the stack (internal format).

NA specifies that the number of attributes in the item is to be pushed onto the stack.

NB specifies that the current Break level number is to be pushed on to the stack. 1 = the lowest level break and 255 = the grand-total line.

ND specifies that the number of items since the BREAK on the break line is to be pushed onto the stack. If on a GRAND-TOTAL line, it equals the item count.

NI specifies the value of the current item counter is to be pushed onto the stack (number of items listed or selected).

NL specifies that the length of the item is to be pushed onto the stack.

NS specifies the current subvalue counter, for columnar listing only, is to be pushed onto the stack.

NV specifies the current multi-value counter, for columnar listing only, is to be pushed onto the stack.

LPV specifies the loading of the value from the previous processing code.

{n} Multiplication of the top two stack entries. If 'n' is specified, the result is divided by 10 raised to the power of n.

- / Divide the second stack entry by the top stack entry and replace the top stack entry with the quotient.
- R Divide the second stack entry by the top stack entry and replace the top stack entry with the remainder.
- + Add the second stack entry to the top stack entry and replace the top stack entry with the sum.
- Subtract the top stack entry from the second stack entry and replace the top stack entry with the difference.
- : The top stack entry is concatenated onto the end of the second stack entry, and the resulting concatenated string replaces the top stack entry.
- [] A subset from the third stack entry is extracted, using the second stack entry as the starting character position, and the top stack entry as the number of characters to be extracted; the result is placed in the top stack entry.
- S A total sum of all previous computation is placed on the top of the stack. The sum operator is used with multi-valued or multi-subvalued elements to produce a single value. Multiple S operators may be present within a function. The domain of a function begins at either the start of the function or immediately following the previous S operator. At the conclusion of the S operator, a single value is present on the stack.
- _ Exchange the top two stack entries.
- P Duplicates the top stack entry back on to the stack.
- (...) A standard conversion operator, enclosed in parentheses, will operate on the top stack entry and the result will replace the original top stack entry.

The following relational 'elm's operate on the top two stack entries, and a result of zero or one is placed in the top stack entry. Depending on whether the condition is not or is satisfied.

- = Stacks a one if the two top stack entries are equal, and a zero is stacked if they are unequal.
- # Stacks a one if the two top stack entries are unequal, and stacks a zero if they are equal.
- > Stacks a one if the second stack entry is greater than the top stack entry, stack zero otherwise.
- < Stacks a one if the second stack entry is less than the top stack entry, a zero otherwise.

-] Stacks a one if the second stack entry is greater than or equal to the top stack entry, a zero otherwise.
- [Stacks a one if the second stack entry is less than or equal to the top stack entry.
- ! Stacks a one if either stack one or two is true, a zero if false.
- & Stacks a one if both stack one and two are true, a zero if not.

4.3.5 Function Processor, A Code:

The NA and NL operands function for A correlative as follows:

- NA specifies that the number of attributes in the item is to be pushed onto the stack.
- NL specifies that the length of the item is to be pushed onto the stack.

4.3.6 File Translation Code, Tfile

The Tfile code provides a facility for converting a value by translating through a file. The value to be translated is used as an item-id for retrieving an item from the defined translation file. The translation value is retrieved from the specified attribute of the item. The general form of the Tfile (Translate) code is:

`T{DICT}file;c{n{,m}};i-amc;o-amc;b-amc`

where:

- T is the code name
- file is the name of the file through which translation takes place, the file name may be preceded by "DICT" to indicate a dictionary.
- c is the translate subcode, which may be one of the following:
 - V Conversion item must exist on file, and the specified attribute must have a value. Aborts with an error message if translation is impossible.
 - C Convert if possible; use original value if item in translate file does not exist or has null conversion attribute.
 - I Input verify only (functions like 'V' for input and like 'C' for output).
 - O Output verify only (functions like 'C' for input and like 'V' for output).
 - X Convert if possible; otherwise return a null value.

- n is an operational value mark count specification. If the c element is followed by a number, the translate will return only the value in VMC n, instead of the complete collection of values concatenated together with blanks. Subvalues will be returned with included blanks.
- m is an operational subvalue mark count specification. If the c element is followed by a number, the translate will return only the subvalue in VMC n, SVMC m, instead of the complete collection of subvalues concatenated together with blanks.
- i-amc is the decimal attribute number for input conversion in Basic. The input value is used as an item-id in the specified file, and the translated value is retrieved from the attribute specified by the i-amc. If the i-amc is omitted, no input translation takes place.
- o-amc is the attribute mark count for output translation. When ACCESS creates a listing, the attribute values will be looked up on the specified file, and the attribute specified by the o-amc will be listed instead of the original value.
- b-amc if specified, will be used instead of o-amc during the listing of break-on and total lines.

4.4 ACCESS USER-EXIT

This user-exit has been added for compatibility with other vendors and is similar to the Basic EXCHANGE function.

4.4.1 EXCHANGE CHARACTERS -- U21A1

The EXCHANGE subroutine swaps one character for another. It may also be used to delete a specific character. System delimiters may not be specified as the source or replacement characters. Format of the EXCHANGE conversion is U21A1,xxyy. Where, if conversion is placed at AMC 7, xx is a two character hexadecimal representation of the character to be replaced, and yy is a two character hexadecimal representation of the replacement character. If yy is FF, then all occurrences of xx will be deleted. The conversion direction may be changed if the conversion is placed at AMC 8, then all occurrences of yy will be replaced by xx. If the character to be replaced is FF, no operation occurs.

4.5 LIST MANIPULATION VERBS

4.5.1 CONCATENATING LISTS

The GET-LIST verb will accept multiple item-ids (lists) resulting in a new list that is a concatenation of the specified lists in the order specified.

```
GET-LIST {filename} {list-name} {list-name} {list-name}
```

If filename is not specified, default is the filename specified in attribute six of the GET-LIST verb. System default is POINTER-FILE. If list-name is not specified, the system default is the at sign '@'.

4.5.2 MERGING LISTS

When concatenating several lists generated by different SELECTIONs, duplicate item-ids may occur. To generate a "unique" list, the GET-LIST verb with the "U" option stores only unique item-ids in the resultant list. The "U" option may be used even if lists are not being concatenated to ensure unique item-ids in a single list.

```
GET-LIST {filename} {list-name} {list-name} {(U)}
```

4.5.3 DELETING ITEMS FROM A LIST USING A SECOND LIST

When it is desirable to delete items from a list using a list, the "D" option is used. The matching contents of the first list specified are removed from the second specified list creating a new list from the second list.

```
GET-LIST {filename} {list-name} {list-name} {(D)}
```

Example:

Lists in POINTER-FILE are:	LIST1	LIST2
	001 ITEM1	001 FIVE
	002 ITEM2	002 FOUR
	003 ITEM3	003 ITEM1
	004 NEW.ITEM	004 ITEM2
		005 ITEM3
		006 THREE

```
>GET-LIST LIST1 LIST2 (D)
```

New list generated from LIST2 is:	001 FIVE
	002 FOUR
	003 THREE

Note that attribute four of LIST1 is ignored.

4.5.4 SORTING LISTS

SORT-LIST will sort specified list(s) into alphabetic order within each list.

SORT-LIST {filename} {list-name} {list-name}

Example of SORT-LIST:

Lists in POINTER-FILE are:

	ABC	DEF
001	C	F
002	A	E
003	B	D

>SORT-LIST ABC DEF

Lists in POINTER-FILE are now:

	ABC	DEF
001	A	D
002	B	E
003	C	F

BASIC 5

BASIC will extend process workspace as required, rather than terminate with an "out of workspace" error.

The maximum string size supported is 16 megabytes. When a string exceeds a pre-determined size, it is stored in overflow space, rather than process workspace. Although strings may exceed 32K bytes, dynamic array subscripts may not exceed 32767. That is, `Z<35000>` is illegal.

Final BASIC object code may not exceed 32767 bytes (32K). Final BASIC object code includes compiled source from the `$INSERT`, `$INCLUDE`, `INCLUDE` and `$CHAIN` directives.

A file that contains BASIC object code does NOT have to be a DC type file.

The Basic Compiler will print all the line numbers that are incorrect in a program, instead of stopping on the first incorrect line. Also, when compiling a "list" of programs and an item in the list is not found in the file being compiled, error message 202 is output and the compile continues to the next item in the list. Therefore, when compiling multiple programs, it may be useful to use the E and P options of the `COMPILE` or `BASIC` verbs to list errors to the printer.

Table 5-1. BASIC Statements

ABORT	EQUATE	INPUTTRAP	PAGE	REMOVE
BREAK	ERROR	INPUTVAR@	PRECISION	RETURN
CALL	EXECUTE	LOCATE	PRINT	RETURN TO
CANCELTRAP	FOOTING	LOCK	PRINTCHAR	REWIND
CASE	FOR	LOOP	PRINTER	RQM
CHAIN	GO	MAT	PRINTERR	SELECT
CLEAR	GOSUB	MATBUILD	PROCREAD	SLEEP
CLEARFILE	GOTO	MATPARSE	PROCWRITE	STOP
COMMON	GO TO	MATREAD	PROMPT	SUBROUTINE
CRT	HEADING	MATREADU	READ	UNLOCK
DATA	IF	MATWRITE	READNEXT	WEOF
DELETE	IFR	MATWRITEU	READT	WRITE
DIM	INPUT	NEXT	READU	WRITET
ECHO	INPUT@	NULL	READV	WRITEU
EDITTRAP	INPUTCLEAR	ON GOSUB	READVU	WRITEV
END	INPUTERR	ON GOTO	RELEASE	WRITEVU
ENTER	INPUTNULL	OPEN	REM	

Table 5-2. BASIC Intrinsic Functions

@	DQUOTE	LANERROR	REM	STR
ABS	EBCDIC	LEN	REPLACE	SUM
ALPHA	EREPLACE	LN	RND	SUMMATION
ASCII	EXP	MAXIMUM	SCAN	SWAP
CHAR	EXTRACT	MINIMUM	SENTENCE	SYSTEM
COL1	FIELD	MOD	SEQ	TAN
COL2	ICONV	NOT	SIN	TIME
COS	INDEX	NUM	SOUNDEX	TIMEDATE
COUNT	INMAT	OCONV	SPACE	TRIM
DATE	INSERT	PROTOCOL	SQRT	TRIMB
DCOUNT	INT	PWR	SQUOTE	TRIME
DELETE				

Table 5-3. Reserved Words

The following words may NOT be used as variable names in PICK BASIC.

AND	ELSE	GO	MATCHES	STEP
AT	END	GT	NE	SETTING
BEFORE	EQ	IN	NEXT	THEN
BY	FROM	LE	ON	TO
CAPTURING	GE	LOCKED	OR	UNTIL
CASE	GOSUB	LT	REPEAT	USING
CAT	GOTO	MATCH	RETURNING	WHILE
DO				

The following words may NOT be used as dimensioned variable names in PICK BASIC.

ABS	EBCDIC	INDEX	NUM	SQRT
ALPHA	ELSE	INMAT	OCONV	STEP
AND	END	INSERT	ON	STR
ASCII	EQ	INT	OR	SUM
BEFORE	EREPLACE	LANERROR	PROTOCOL	SUMMATION
BY	ERROR	LE	PWR	SWAP
CASE	EXCHANGE	LOCATE	REM	SYSTEM
CAPTURING	EXP	LEN	REPEAT	TAN
CAT	EXTRACT	LN	REPLACE	THEN
CHAR	FIELD	LOCKED	RETURNING	TIME
COL1	FOLD	LT	RND	TIMEDATE
COL2	FROM	MATCH	SCAN	TO
COS	GE	MATCHES	SEQ	TRIM
COUNT	GO	MAXIMUM	SENTENCE	TRIMB
DATE	GOSUB	MINIMUM	SETTING	TRIMF
DCOUNT	GOTO	MOD	SIN	UNTIL
DELETE	GT	NE	SOUNDEX	USING
DO	ICONV	NEXT	SPACE	WHILE
DQUOTE	IN	NOT	SQUOTE	

5.1 BASIC VERBS

5.1.1 RUNI

The RUNI verb is equivalent to the RUN verb, but inhibits the use of strings longer than 32739 bytes, so that the variables may be preserved for use by another BASIC program using the "I" option to inhibit initialization of the data area. If the RUN verb is used instead, variables which contained long strings may be un-assigned when another BASIC program is run using the "I" option.

5.1.2 BLIST

BLIST is a TCL-II verb used to produce a formatted listing of a BASIC program. The output may be directed to either the CRT or the spooler. Structured program constructs (IF/THEN/ELSE, FOR/NEXT, LOOP/WHILE/DO/REPEAT, BEGIN CASE/CASE/END CASE) are indented and all statement labels are left justified in the left margin.

The format for use of the BLIST verb is:

>BLIST file-name item-list (options)

Where "file-name" is any BASIC source file; "item-list" is a list of program names, an asterisk (*) for all items, or null if there is a select list active. If the file is a "list" (DC) type file, a program stored as a "list" is acceptable input for the BLIST verb. Available options are:

- B - Expand lines containing only an exclamation point (!) into a line of asterisks.
- C - Do not indent comment lines.
- D - Double space the listing.
- L - Suppress the editor line number.
- N - No wait at end of page (on CRT listings).
- P - Output to the spooler.
- S - Draw vertical structure lines.
- T - Output to the terminal.

If a line is too long to fit into the page width (as setup with the TERM command), an attempt is made to fold the line at blanks, commas, or other delimiters. The folded portion of the line is indented to the same level as the original line. Any control characters encountered are displayed as periods.

If the S option is specified, then vertical lines will be drawn between the beginning of a structured construct and the end of the construct. This option is useful when constructs span pages or are nested many levels deep.

The T option is intended for use when the P option has been included in the default options (see next section). The T option overrides the P option and should never be included in the default options (else output to the spooler will be impossible).

5.1.2.1 BLIST CUSTOMIZATION

The BLIST verb is table driven. The table is stored in the ERRMSG file. The item-id of the control table is "BL" followed by a 4 character hexadecimal number. The table number is contained in line 4 of the verb definition. The default table number is 0, thus the default table item-id is "BL0000".

Line one of the control table defines the left margin, number of spaces to indent for each construct, and the default options. Each of these is separated by a comma. The default control table contains "7,3,(BS)" on line 1, designating a left margin of 7 (statement label area), 3 spaces to indent for each construct, and the options B and S are used by default.

Line two of the control table defines the key words used at the beginning of a statement which may begin or end a construct or which may begin a comment. Each value on this line contains the key word followed by a sub-value mark, followed by a type character ("*" for comment, ";" for statement), followed by two digits. The first digit specifies the number of outdents for the statement, the second digit specifies the number of indents for the statement. Note that some statements may contain a value in each digit (such as WHILE, which will outdent the statement containing the WHILE, and then indent the following statements).

Line three of the control table defines the key words used at the end of a statement which may begin or end a construct. The format is the same as line two.

5.1.3 BFORMAT

BFORMAT is a TCL-II verb used to format a BASIC source program and update the source file with the formatted item. The action of BFORMAT is identical to BLIST, except that the output is not printed, but is filed into the source file overwriting the original source program.

The format for use of the BFORMAT verb is:

>BFORMAT file-name item-list (options)

Where "file-name" is any BASIC source file, "item-list" is a list of program names, an asterisk (*) for all items, or null if there is a select list active. If the file is a "list" (DC) type file, a program stored as a "list" is acceptable input for the BFORMAT verb. Available options are:

O - Override existing margin and code indent

n1-n2 - n1 number of spaces to add to default margin indent
n2 number of spaces to add to default code indent

BFORMAT filename item (O 4-2)

R - Renumber statement labels and all references to the labels (GOTO, GOSUB, RETURN TO).

n1-n2 - n1 specifies new beginning statement label number
n2 specifies increment between statement labels
Both n1 and n2 default to 10.

BFORMAT filename item (R 5-5)

The R option is useful for renumbering a source program to make it easier to follow. If the R option is used, then a line containing only an exclamation point (!) and a statement label may be used to change the current new statement label number to the label specified on that line. This is true as long as the current statement label number is greater than what would be the next number (previous label number plus increment). This is useful when specific labels make the program easier to follow, such as at the beginning of subroutines, etc. Note: Certain forms of the GOTO, GOSUB, and RETURN TO statements will not be renumbered. This occurs when there is no blank between the key word and the destination label, e.g. GOTO20 will not be renumbered, but GOTO 20 will. Also, a mixture of numeric and alpha-numeric statement labels following an ON GOTO or ON GOSUB will fail to renumber any numeric labels past the first alpha-numeric label.

5.1.3.1 BFORMAT CUSTOMIZATION

The BFORMAT verb is table driven. The table is stored in the ERRMSG file. The item-id of the control table is "BF" followed by a 4 character hexadecimal number. The table number is contained in line 4 of the verb definition. The default table number is 0, thus the default table item-id is "BF0000".

The structure of the table is identical with the BLIST control table. The only default options are R for renumber, C for comment indent inhibit, and the numeric options for specifying the starting statement label number and increment. It is possible to specify the numeric options in the control table, and leave the R option as a run-time option. In this case, the numeric options need not be specified at run-time.

5.1.4 RENUMBER

The RENUMBER verb is equivalent to using the BFORMAT verb with the R option, with a control table specifying no indentation. The verb will renumber statement labels and all references to the labels: GOTO, GOSUB, RETURNTO, and trim redundant blanks from a source program with default parameters.

5.2 ASSIGNING VALUES TO VARIABLES

The format of the following addition, subtraction and concatenation assignment operators will create more efficient object code thereby improving the speed of a Basic program.

$X += 1$

is more efficient than

$X = X+1$

which produces the same results

$X<3> -= 1$

$X(2) := 1$

are more efficient than

$X<3> = X<3> - 1$

$X(2) = X(2) : 1$

which are equivalent.

5.3 BASIC or COMPILE

The basic compiler will allow null lines to be included in the basic source code.

Example: Program TEST

```
001 PRINT 'Following lines are null'
002
003
004 PRINT 'The above lines are null'
```

The basic compiler will allow a source line to contain a the label only, that is, no comment character need follow the label.

Example: Program TEST

```
001 10
002 INPUT X
003 IF X='' THEN GO 10
```

The compiler will distinguish between label names and variable names. Therefore, alpha-numeric labels may now have the same name as a variable name.

5.4 BASIC ITEM LOCKS

BASIC item locks allow applications to lock individual records, rather than groups of records. This is of major importance when writing an application where there are many users updating the same database. Item locks prevent updates from being "lost" by overwriting, since only one user is allowed to update a specific item at any one time. If a process "owns" the item by setting an item lock, then it is the only process that may access the item for writes, until it releases the lock. An item-locked item may be read with a READ, READV or MATREAD, but not written.

The noticeable difference between item locks and group locks, is that when a process attempts to lock an item that is already locked by another process, it will issue the bell character approximately every two seconds. A group lock issues the bell character every second.

The maximum number of Basic Item-Locks is 2036.

All group locks and item locks are released if a program is aborted from the system or Basic debugger.

The verb CLEAR-ITEM-LOCKS is available in the SYSPROG account, but should be used only in an emergency situation.

5.4.1 CHANGES TO BASIC STATEMENTS

The addition of item locks to BASIC will allow the same user applications to function. However, with the addition of item locks, it may be beneficial to restructure applications with item locks in mind.

The following is a brief explanation of how the various statements function in relation to group locks and item locks.

5.4.1.1 READ, READV, and MATREAD

READ, READV, and MATREAD set a group lock. If an item in the group does not exist, the group is released immediately. If the item does exist, the data from the item is transferred to the Basic variable, and the group is released. The THEN/ELSE clauses are then processed.

5.4.1.2 READU, READVU and MATREADU

READU, READVU and MATREADU attempt to set an item lock. If another user has locked the item via READU, READVU or MATREADU; then the process will "wait" until the lock is released by that user. The "wait" will cause the terminal to "beep" every two seconds unless the LOCKED clause is part of the statement. If the LOCKED clause is part of the statement the LOCKED is processed. If the item is not locked by another user, then the group is locked, the item is locked and read, and the group is unlocked.

5.4.1.3 WRITE, WRITEV, MATWRITE and DELETE

WRITE, WRITEV, MATWRITE and DELETE lock the group, update the file, and release the group lock. When the item update is complete, then the item lock is released.

5.4.1.4 WRITEU, WRITEVU and MATWRITEU

WRITEU, WRITEVU and MATWRITEU lock the group, update the file, and releases the group lock. Item locks are not released.

5.4.1.5 RELEASE

RELEASE does not operate on group locks. It will release one item lock or all item locks set by the current process.

5.4.1.6 STOP, ABORT and CHAIN

STOP, ABORT and CHAIN release all item locks set by the current process.

5.4.2 DISPLAYING GROUP AND ITEM LOCKS

The LIST-LOCKS and WHAT verbs display group locks followed by Basic item locks. The format of the display of each lock is as follows:

ffffff (xxxxx)-ppp Group lock

ffffff (xxxxx)-hhhh-ppp Item lock

where:

fff is the frame number of the base of the group containing the item

xxx is that same number expressed in hexadecimal

hhh is a hexadecimal hash of the item-id used to differentiate between different items in the same group

ppp is the port number of the process setting the lock

Example: LIST-LOCKS

15840 (03DE0)-030 <== Group lock

148652 (244AC)-600D-029 <== Item lock

```
### ### ### ### ### ### ### ### ### ### ### ### ### ### ###
### ### ### ### ### ### ### ### ### ### ### ### ### ### ###
### ### ### ### ### ### ### ### ### ### ### ### ### ### ###

### ### ### ### ### ### ### ### ### ### ### ### ### ### ###
### ### ### ### ### ### ### ### ### ### ###
```

5.5 BASIC CONVERSION CODES

The following conversion codes may be used in BASIC via the OCONV and ICONV functions. The general form of these functions is;

```
ICONV("string","input-conversion")
OCONV("string","output-conversion")
```

5.5.1 Center (XCn)

The CENTER conversion is used to produce a centered field in a BASIC listing or report. To use the CENTER conversion, place XCn, where n is the column width to center in, in the OCONV statement. 'n' must be greater than 4 and less than 32768.

```
VAR = OCONV("ABC","XC5")
```

Using the PRINT statement to print VAR would output "ABC" between two spaces. Thus, ABC is centered in column width of five.

5.6 BASIC SUBROUTINES (USER-EXITS)

The following user-exits have been provided for compatibility with other vendors.

5.6.1 GET INPUT LINE -- U31A1

A TCL input statement, without the verb, may be retrieved by a Basic program. If the program was cataloged, and the verb definition in the M/DICT has a "PG" in line 1, then this routine will return the entire input line including the options. This prevents the options from being used by the BASIC run-time system (e.g. P option is PRINTER ON, D option for DEBUG, etc.), and allows the program to access and act upon them instead. The calling sequence is:

```
TCL = OCONV('','U31A1')
```

where the variable TCL will contain the TCL input line (without the verb) after the call. The backslash character (\) will be changed to the double quote character ("). This routine is often used in conjunction with the next routine, PARSE, which parses the input line into discrete "words".

5.6.2 PARSE INPUT LINE -- U41A1

The input line is parsed into discrete "words". Each word (or phrase) is separated by an attribute mark. Words in the input line are separated by blanks, phrases are enclosed in single quotes, double quotes or back-slashes. All output "words" which were enclosed in delimiters are succeeded with a value mark followed by the delimiter character. This makes it easy for the program to determine if the word was quoted or not. The calling sequence is:

```
LIST = OCONV(TCL,'U41A1')
```

where TCL is the input string and LIST contains the output "words". Note that Get Input Line and Parse Input Line may be called together.

```
LIST = OCONV('','U31A1]U41A1')
```

where "]" is a value mark.

5.7 BASIC STATEMENTS

To provide compatibility with other vendors the following statements and functions have been expanded or added.

5.7.1 \$INSERT, \$INCLUDE, INCLUDE

The \$INSERT directive is used to insert source code from one item as part of another item. Typically \$INSERT would be used with a set of related BASIC programs with variables in COMMON. The COMMON declarations can be placed in a single item which is "inserted" into each program by the \$INSERT directive. Using the \$INSERT directive saves space, makes changes easier, and reduces the chance of declarations in one program mismatching those in another.

\$INCLUDE or INCLUDE may be substituted for \$INSERT if so desired. The form of the \$INSERT directive is:

```
$INSERT {filename} itemname or,
$INCLUDE {filename} itemname or,
INCLUDE {filename} itemname
```

If the filename is omitted, the file containing the program currently being compiled will be assumed. Users should note that the object code of any BASIC program or external subroutine, whether or not it contains \$INSERT directives, may not exceed 32768 bytes in size.

5.7.2 CLEARFILE

With the implementation of the "updates-only" file-save, which uses update flags set in either the file-definition item or the data-item to determine what should be saved, the BASIC CLEARFILE statement will not operate efficiently if the file data is expected to be saved at file-save time. That is, if the file is not a DY or DX type file. This is because the CLEARFILE statement cannot access the file-definition item and flag the file to be cleared, so it must flag every item in the file to be deleted. There are two ways to increase the efficiency of performing CLEARFILE operations. First, making the file a DX or DY file will prevent update flagging. Second, use of the TCL command CLEAR-FILE via the EXECUTE statement instead of the BASIC CLEARFILE statement. The CLEAR-FILE command accesses the file-definition item and flags the file to be cleared.

5.7.3 READING AND WRITING LISTS

Since lists are stored as normal items, BASIC programs may read and write lists using the READ and WRITE statements. Binary items may not be processed by a BASIC program, an attempted READ of a binary item will take the ELSE clause. This case may be identified by using the SYSTEM(0) function in the ELSE clause. SYSTEM(0) will return the value of 10 if the last READ attempted to read a binary item.

5.7.4 LOCATE

Both forms listed below locate an attribute, value, or secondary value within a dynamic array. If not found, the index value is returned which can be used in an INSERT statement to put the missing "string" value in the correct location. The same wildcard partials used in Access may be used within the locate string.

Format:

```
LOCATE(string,item{,a#{,v#{,exp}}};var{;seq}) THEN ... ELSE ...
```

```
LOCATE string IN item{<a#{,v#}>}{,exp}{BY seq} SETTING var  
THEN ... ELSE ...
```

If the locate string begins with the character / and ends with the character \ and the wildcard partial defined is [^], then the occurrence of the character [^] would be treated as wildcard characters instead of their actual value. If the [wildcard partial is used, it must immediately follow the / character. If the] wildcard partial is used, it must immediately precede the \ character. If the characters / and \ are used without any other characters, they are treated as normal characters, but cause more system overhead.

Example: Program TEST

```
001 EQU AM TO CHAR(254)
002 X='ABC':AM:'DEF':AM:'ABF':AM:'AGEFZ'
003 LOCATE '/[F\]' IN X SETTING VAL1 THEN NULL
003 LOCATE '/[B]\]' IN X SETTING VAL2 THEN NULL
003 LOCATE '/[EF\]' IN X SETTING VAL3 THEN NULL
003 LOCATE '/D^F\]' IN X SETTING VAL4 THEN NULL
003 LOCATE '/[G^F]\]' IN X SETTING VAL5 THEN NULL
```

The following values for VALn are:

```
VAL1 = 2
VAL2 = 1
VAL3 = 2
VAL4 = 2
VAL5 = 4
```

5.7.5 CALL and ENTER

CALL transfers control to subroutine and passes the subroutine all values specified in the argument list.

ENTER transfers control from one cataloged program to another cataloged program.

Both may now specify a filename within the CALL or ENTER statement.

For the @var format, var may contain the filename. When @var is used, var cannot be an array (i.e., ENTER @DIM(1) or CALL @DYN<1>).

Format: CALL {filename,}name {(argument list)}
CALL @var {(argument list)}

ENTER {filename,}name
ENTER @var

Example: Program TEST
001 CALL BP,TEST
002 VAR='BP,TEST'
003 CALL @VAR
004 ENTER BP,TEST
005 ENTER ACCT,BP,TEST
006 ENTER SYSTEM,ACCT,BP,TEST
007 ENTER @VAR

NOTE: The filename used in both the CALL and ENTER statements is the literal filename, not the file variable defined by the OPEN statement (i.e., OPEN 'BP' TO BPFILE THEN). Should be CALL BP,TEST, not CALL BPFILE,TEST.

5.7.6 PRINTCHAR

PRINTCHAR allows outputting of any single character. The main purpose of this function is to allow output of a segment mark character to the spooler. Output may be directed to the CRT or to the Spooler.

Format: PRINTCHAR char

Example: Program TEST
001 PRINTER ON
002 PRINTCHAR CHAR(255)

5.7.7 INSERT or REPLACE VALUE WITHIN VARIABLE STRING

The following syntax allows values within a variable string to be inserted or replaced, making it much easier to build print buffers for output.

Format: var[n{,m}] = string

n = starting position (must be > 0)

If var length is less than n, you get a blank fill

m = # of replaceable characters (must be >= 0)

If m is not present or negative, then overlay occurs

If replacement string length is greater than m, then insertion occurs

Example: Program TEST

001 X='123456789'

002 X[4]='ABC';* X = 123ABC789

Example: Program TEST

001 X='123456789'

002 X[4]='ABCD';* X = 123ABCD89

Example: Program TEST

001 X='123456789'

002 X[4,2]='ABCD';* X = 123ABCD6789

Example: Program TEST

001 X='123456789'

002 X[4,0]='ABCD';* X = 123ABCD456789

Example: Program TEST

001 X='12'

002 X[5]='ABC';* X = 12..ABC

Example: Program TEST

001 X=''

002 X[4]='ABC';* X = ...ABC

Note the blank fill in the last two examples as indicated with the periods.

5.7.8 INPUT @

Three new edit masks are available with the INPUT @ command. These are X, N, and S masks. These masks provide additional input requirements and display editing with INPUT @. If a mask is present then the length factor overrides any implied length value defined by the n option following the variable name.

Format: INPUT @(x,y):var{,n}{:} {mask} { } {THEN statement ELSE statement}

Xn edit mask: Allows for input of up to n ascii characters in the range x'20' through x'7E'. Any character outside the range will result in the error message 'Field must be ascii format!' followed by the display mask. The value n can have a value of 1 through the terminal width value of the TERM command.

Nn edit mask: Allows for input of up to n numerics in the range 0 through 9. Any non-numeric will result in the error message 'Field must be numeric!' followed by the display mask. The value n can have a value of 1 through the terminal width value of the TERM command.

Sn edit mask: Allows for input of up to n signed numerics in the range 0 through 9 including a sign character of - or +. If the + character is input, it will be output as a blank. Any non-numeric will result in the error message 'Field must be numeric!' followed by the display mask. The value n can have a value of 1 through the terminal width value of the TERM command.

Nn.m edit mask: Allows for input of up to n numerics in the range 0 through 9 including the decimal position. Any non-numeric will result in the error message 'Field must be numeric!' followed by the display mask. If the field is too long, the message 'Field is too long!' will be displayed followed by the display mask. The value n can have a value of 1 through the terminal width value. The value of m can have a value of 1 through one less than the value of n. (n-1)

Sn.m edit mask: Allows for input of up to n signed numerics in the range 0 through 9 including the decimal position and a sign character of - or +. If the + character is input, it will be output as a blank. Any non-numeric will result in the error message 'Field must be numeric!' followed by the display mask. If the field is too long, the message 'Field is too long!' will be displayed followed by the display mask. The value of n can have a value of 1 through the terminal width value. The value of m can have a value of 1 through one less than the value of n. (n-1)

Examples of edit masks:

Mask	Display	Input	Output	Stored
X4	AB	AB	AB
N4	NNNN	123	123	123
S4	SNNN	-12	-12	-12
N6.2	NNN.NN	145	1.45	145
S6.2	SNN.NN	+12	0.12	12
N6.2	NNN.NN	14.5	14.50	1450
S6.2	SNN.NN	12.	12.00	1200

Examples of INPUT @ Statement with edit masks:

```
INPUT @(5,10):VAR:X4
INPUT @(5,10):VAR N4
INPUT @(5,10):VAR:N6.2_ THEN.. ELSE..
INPUT @(5,10):VAR S6.2_ THEN.. ELSE..
```

5.7.9 INPUTVAR @

The INPUTVAR @ statement accommodates THEN and ELSE clauses for not-null and null variable values, respectively. The general form of the Masked Input statement is:

```
INPUTVAR @(x,y):variable{,n}{:}{mask}{_} THEN statement ELSE statement
```

This is a very complex input function. It is capable of replacing as many as 20 lines of BASIC code used in screen input. Its functions include cursor addressing, editing, output masking, error messages, input masking, and exception trapping.

This command itself is used for the actual entry of the data. Ancillary functions can be performed by the commands described below. In the general form above, "variable" represents the name of the variable being input, and "mask" represents a standard PICK format mask. If the variable being used already has a value, it will be displayed at the specified cursor address using "mask" as the output mask. Regardless, the cursor is positioned one character back of "x" in the "@(x,y)" specification, the prompt character is printed and input is requested. If the user presses the return key, then whatever default value was there before will be accepted. Otherwise, the input will be verified against the mask, and if acceptable, will be assigned to "variable". If the mask contains a decimal digit specification and/or a scaling factor, then numeric checking will be performed. If the mask contains a length specification (for example, R#10), then length checking will be performed. If the mask is 'D' (or any other valid date mask), then a date verification will be performed.

If n is used, input is only accepted if less than or equal to n characters. When the nth character is input, there will be an automatic carriage return and line feed. If the underscore (_) is used in combination with the length specification (n), and an attempt is made to enter more characters than specified by 'n', the bell signal will be echoed to the terminal and the automatic carriage return is suppressed.

The THEN clause is taken if the value input for the variable is not null, or the value for the variable is not null. The ELSE clause is taken if only a carriage return is input and the value for the variable is null, or if the input null character is used to null the variable. (As opposed to INPUT@, where the ELSE clause is taken when the value for variable is not null and carriage return is input).

The THEN clause, ELSE clause, or THEN and ELSE clause must accompany the INPUTVAR @ statement. They are not optional. If you do not need the THEN or ELSE clause, use the INPUT @ statement.

5.7.10 DATE INPUT

A date reference in the statements INPUT @ and INPUTVAR @ will accept numeric input with or without a field separation character.

```
INPUT @(x,y):var 'D'
INPUTVAR @(x,y):var 'D'
```

Acceptable numeric input formats are:

where: D=Day M=Month Y=Year

U.S.International

MDD	DDM
MMDD	DDMM
MDDYY	DDMY
MMDDYY	DDMMYY
MDDYYYY	DDMYYY
MMDDYYYY	DDMMYYYY

Examples:

In U.S. format:

```
123   = JAN 23
21388 = FEB 13, 1988
```

In International format:

```
123   = 12 MAR
21388 = 21 MAR 1988
```

5.7.11 EDITTRAP

EDITTRAP will allow trapping with the INPUT @ statement. This allows more flexibility than INPUTTRAP since EDITTRAP works on individual trap characters or strings of characters and may have different routines for each different character. It also allows for clearing a single character trap without disturbing the other EDITTRAP commands. This, like INPUTTRAP, is not passed or returned through subroutines. Therefore, each subroutine can have its own EDITTRAP commands without disturbing the EDITTRAP commands from the calling program. EDITTRAP characters take precedence over INPUTTRAP characters.

```
Format: EDITTRAP char {THEN statements}
        EDITTRAP string {THEN statements}
```

EDITTRAP, without the THEN statement, clears the trap for that particular character or string of characters. EDITTRAP, with the THEN statement, will test for the EDITTRAP character or character within a string of characters, and if a match occurs, will take the trap and execute the THEN statements. The characters X'1B', X'5B', and X'FB' are all represented by character X'5B'. If the character in the EDITTRAP command is a left bracket ([X'5B') for instance, then a trap is taken when the characters X'1B' (Escape), X'5B' (Left bracket), or X'FB' (Modified escape character) are detected. Should the escape character (X'1B'), or characters X'5B' or X'FB', be entered with more than 1 character, for example, via a function key, then the character following the escape character can be recalled with system function SYSTEM(21). EDITTRAP character must not be a attribute, value or sub-value mark (i.e., cannot be a system delimiter).

Example: Program TEST

```
001 EDITTRAP "*" THEN ... Defines the * character as a trap character
                             for INPUT @.
002 EDITTRAP "!" THEN ...  Defines the ! character as another trap
                             character for INPUT @.
003 EDITTRAP "#$%" THEN ... Defines the # and $ and % character as trap
                             characters for INPUT @. If the character #
                             was entered at an INPUT @ statement, then
                             this trap would be taken. The same for the
                             characters $ or %.
004 EDITTRAP "$"           Clears the trap for the $ character.
005 EDITTRAP "*!"          Clears the trap for the * and ! characters.
006 CANCELTRAP             Clears all previous EDITTRAP character traps.
```

Example: Program INPUT@.TEST

```
001 * SAMPLE PROGRAM CONTAINING INPUT @, EDITTRAP, INPUTERR STATEMENTS
002
003 * INITIALIZATION
004 EQU TRUE TO 1, FALSE TO 0, AM TO CHAR(254)
005 PROMPT ""
006 END.OF.THE.WORLD=FALSE; LEN.LIM=1; MASK="X1"
007 MESSAGE="SAMPLE PROGRAM FOR INPUT @ WITH MASK, EDITTRAP, AND INPUTERR
STATEMENTS"
008
009 LOOP
010
011 * TRAP THE '%' OR ']' CHARACTERS ON A FOLLOWING INPUT @ STATEMENT
012 | EDITTRAP "%]" THEN
013 | | INPUTERR "ENTER '.' TO END ELSE <RETURN> TO REENTER ---->";
014 | | EDITTRAP "." THEN STOP; * TRAP THE '.' CHARACTER
015 | | INPUT @(49,23):CR
016 | | EDITTRAP "."; * CANCEL TRAP OF THE '.' CHARACTER
017 | END
018
```

```

019 * ENTER THE LENGTH LIMIT VALUE 'N' FOR INPUT @(X,Y):VAR,N
020 * TRAP '%' OR ']'
021 | CRT @(-1)
022 | CRT @(0,2):OCONV(MESSAGE,"XC79")
023 | CRT @(0,5):"ENTER LENGTH LIMIT ":"LEN.LIM
024 | CRT "('%' OR ']' TO TRAP)"
025 | INPUT @(20,5):INP.LIM THEN LEN.LIM=INP.LIM
026
027 * CANCEL EDITTRAP OF '%' CHARACTER
028 | EDITTRAP "%"
029
030 * ENTER THE FORMAT MASK FOR INPUT @(X,Y):VAR,N MASK
031 * TRAP ONLY ']'
032 | CRT @(0,8):"ENTER THE FORMAT MASK ":"MASK
033 | CRT "(']' TO TRAP)"
034 | INPUT @(23,8):INP.MASK THEN MASK=INP.MASK
035
036 UNTIL END.OF.THE.WORLD DO
037
038 * ENTER DATA INTO THE INPUT VARIABLE 'VAR' WHICH IS MASKED
039 | CRT @(0,12):"ENTER DATA STRING TO MATCH FORMAT MASK --->"
040 | CRT "(']' TO TRAP)"
041 | INPUT @(44,12):VAR,LEN.LIM:MASK
042 | CRT
043
044 * DISPLAY THE HEX SEQUENCE OF EACH CHARACTER IN 'VAR'
045 | CRT @(0,14):
046 | FOR I=1 TO LEN(VAR)
047 | | CRT VAR [I,1]:" ":ICONV(SEQ(VAR[I,1]),"MCXD")
048 | NEXT I
049
050 | INPUTERR "<RETURN> TO CONTINUE --->":
051 | INPUT CR:
052
053 REPEAT

```

5.7.12 CANCELTRAP

Clears all previous EDITTRAP character traps.

5.7.13 IFR

Functions identically to the IF statement unless variable 1 or variable 2, or both variable 1 and variable 2 are multi-valued fields. On all operators excepting "#", a logical OR comparison is done on each value until a true is achieved, on "#" a logical AND comparison is done. Statement formats are:

```
IFR expression THEN statement(s)
IFR expression ELSE statement(s)
IFR expression THEN statement(s) ELSE statement(s)
```

Example 1; THEN clause is taken:

```
EQU VM TO CHAR(253)
V1 = 'C'
V2 = 'A':VM:'B':VM:'C'
IFR V1 = V2 THEN
  CRT 'Value "B" in V1 equals values "A", or "B", or "C" in V2'
END ELSE
  CRT 'Value "B" in V1 does not equal values "A", or "B", or "C" in V2'
END
```

This would compute to: IF C = A OR IF C = B OR IF C = C THEN... ELSE...

Example 2; ELSE clause is taken

```
EQU VM TO CHAR(253)
V1 = 'C'
V2 = 'A':VM:'B':VM:'C'
IFR V1 # V2 THEN
  CRT 'Value "C" in V1 does not equal values "A", and "B", and "C" in V2'
END ELSE
  CRT 'Value "C" in V1 is equal to values "A", or "B", or "C" in V2'
END
```

This would compute to: IF C # A AND IF C # B AND IF C # C THEN... ELSE...

Example 3; ELSE clause is taken

```
EQU VM TO CHAR(253)
V1 = 'Y':VM:'Z'
V2 = 'A':VM:'B':VM:'C'
IFR V1 = V2 THEN
  CRT 'Values "Y" or "Z" in V1 equal values "A", or "B", or "C" in V2'
END ELSE
  CRT 'Values "Y" or "Z" in V1 do not equal values "A", or "B", or "C" in V2'
END
```

This would compute to: IF Y = A OR IF Y = B OR IF Y = C OR
IF Z = A OR IF Z = B OR IF Z = C THEN... ELSE...

Example 4; THEN clause is taken

```

EQU VM TO CHAR (253)
V1 = 'Y':VM:'Z'
V2 = 'A':VM:'B':VM:'C'
IFR V1 # V2 THEN
  CRT 'Values "Y" or "Z" in V1 do not equal values "A", or "B", or "C" in
    V2'
END ELSE
  CRT 'Values "Y" or "Z" in V1 equal values "A", or "B", or "C" in V2'
END

```

This would compute to: IF Y # A AND IF Y # B AND IF Y # C AND
IF Z # A AND IF Z # B AND IF Z # C THEN... ELSE..

Example 5; ELSE clause is taken

```

EQU VM TO CHAR(253)
V1 = 'X'
V2 = 'Y'
V3 = 'Z'
V4 = 'A'
V5 = 'B'
IFR V1:VM:V2:VM:V3 = V4:VM:V5 THEN
  CRT 'Values "X" or "Y" or "Z" in variables V1, V2 and V3 equal values
    "A" or "B" in variables V4 or V5'
END ELSE
  CRT 'Values "X" or "Y" or "Z" in variables V1, V2 and V3 do not equal
    values "A" or "B" in variables V4 or V5'
END

```

This would compute to: IF X = A OR IF X = B OR IF Y = A OR IF Y = B OR
IF Z = A OR IF Z = B THEN... ELSE...

5.7.14 REMOVE

The REMOVE statement successively extracts elements from a dynamic array. For use in repetitive extractions.

REMOVE substring FROM dyn.array AT posn SETTING delim

'dyn.array' is the dynamic array from which the elements are extracted. It is not modified by executing the REMOVE statement.

'substring' is to contain the extracted element.

'posn' defines the start position of the next element to be extracted. It should normally be initialized to zero before the REMOVE is executed for the first time. It is updated by the REMOVE statement to point to the delimiter which terminated the REMOVE.

'delim' will contain an indication of the delimiter which terminates the extracted substring, and will be one of the following values after the REMOVE statement has been executed.

- 0 - end of string
- 2 - substring terminated by attribute mark
- 3 - substring terminated by value mark
- 4 - substring terminated by subvalue mark
- 5 - substring terminated by CHAR(251)
- 6 - substring terminated by CHAR(250)
- 7 - substring terminated by CHAR(249)

Example:

The dynamic array 'DYN' contains the following data with attribute marks (CHAR(254)) represented by '^'; value marks (CHAR(253)) represented by ']' and subvalue marks (CHAR(252)) represented by '\'.

DYN = ABC]DEFG^HIJKLMN

Assuming that POSN = 0, the statement:

REMOVE STRING FROM DYN AT POSN SETTING DELIM

returns the following values as it is repeatedly executed

1. STRING = ABC
POSN = 4
DELIM = 3
2. STRING = DEFG
POSN = 9
DELIM = 2
3. STRING = HIJKMNOP
POSN = 18
DELIM = 0
4. STRING = ''
POSN = 18
DELIM = 0

5.7.15 ERROR

The ERROR statement allows output of error message data like the STOP and ABORT functions, but allows program continuation.

Format: ERROR errnum{,param{,param{, ...}}}

Example: Program TEST

```
001 10 CRT 'Enter ID: ':
002 INPUT ID:
003 READ STRING FROM FILE,ID ELSE
004   ERROR 24,ID
005   GO 10
006 END
```

If the item ID is not on file, and ID equals TEST, then error message "[24] The word "TEST" cannot be identified." will be displayed.

5.8 BASIC FUNCTIONS

5.8.1 SWAP()

The SWAP() function is synonymous to the EREPLACE() function.

EXAMPLE 1:

```
var=SWAP(source,string1,string2{,#{,start}})
```

Allows string replacement similar to the Editor replace command.

source: source string where replacement will take place.
 string1: text to be replaced
 string2: replacement text
 #: number of occurrences to replace
 start: which occurrence to start with

EXAMPLE 2:

```
var=SWAP(string,xx,yy)
```

Swaps one character for another or deletes all occurrences of a specified character.

xx: hexadecimal representation of the character to be replaced
 yy: hexadecimal representation of the replacement character

If "yy" is FF, then all occurrences of "xx" will be deleted. If "xx" or "yy" are less than two characters, then no conversion takes place.

5.8.2 INMAT()

The INMAT() function determines the number of attributes processed by the last MATREAD or MATPARSE statement.

Format is:

```
var=INMAT()
```

Is only valid immediately after MATREAD or MATPARSE statement.

5.8.3 SENTENCE()

The SENTENCE() function will retrieve the TCL input line, including the verb, that invoked the BASIC program.

Format is:

var=SENTENCE()

NOTE: EXECUTE statement will update the TCL input line.

5.8.4 INDEX

INDEX searches a string for a specified occurrence of a specified substring, and returns all locations of a substring. If the substring is not found, 0 is returned. The same wildcard partials used in Access may be used within the index substring.

Format: var = INDEX(string,substring,occurrence)

If the index substring begins with the character / and ends with the character \ and the wildcard partials defined are [^], then the occurrence of the characters [^] would be treated as wildcard characters instead of their actual value. If the occurrence value equals -1, then all locations of the substring will be returned in a multi-valued field. Although the wildcard partials [] can be entered, they are not used and not necessary with the INDEX function. If the [wildcard partial is used, it must immediately follow the / character. If the] wildcard partial is used, it must immediately precede the \ character. If the characters / and \ are used without any other characters, they are treated as normal characters.

Example: Program TEST

```
001 X='ABCADEAFG'
002 Y=INDEX(X,'A',-1)
003 Z=INDEX(X,'/A^E\,1)
004 L=INDEX(X,'/^E]\',1)
```

The value for Y will be a multivalued field equal to 1]4]7. The value for Z and L will be equal to 4.

5.8.5 EREPLACE

EREPLACE allows for string replacement similar to the Editor command. The same wildcard partials used in Access may be used within the replace substring str1.

Format: var = EREPLACE(source,str1,str2{,{start}})

If the replace substring str1 begins with the character / and ends with the character \ and the wildcard partials defined are [^], then the occurrence of the characters [^] would be treated as wildcard characters instead of their actual value. If the [wildcard partial is used, it must immediately follow the / character. If the] wildcard partial is used, it must immediately precede the \ character. If the characters / and \ are used without any other characters, they are treated as normal characters.

Example: Program TEST

```
001 X='ABCADEAFG'
002 Z=EREPLACE(X,'/^A\','*')
003 L=EREPLACE(X,'/[AD\','*')
004 R=EREPLACE(X,'/AD]\','*')
```

The value for Z would be AB*D*FG. The value for L would be *EAFG. The value for R would be ABC*.

5.8.6 SCAN

SCAN will scan a string for the first occurrence of up to three user definable ASCII characters in addition to any of the system delimiters: subvalue, value, attribute, and segment marks. The user defined characters must be separated by a system delimiter as specified above, and can be any combination of system delimiters.

Format: var = SCAN(string,char1}:del:char2}:del:char3})

Example: Program TEST

```
001 EQU AM TO CHAR(254), VM TO CHAR(253)
002 X="LIST MD 'WHO' 'TEST'"
003 S="':AM:':AM:\'"
004 Y=SCAN(X,S)
```

The value of Y will be 9, which would represent the position where the single quote was located. This example of the SCAN function was searching for the first occurrence of a double quote, single quote, or back slash character. As well as stopping on the first occurrence of any of the three characters specified, the process will also stop on any system delimiter.

5.8.7 SUM

The SUM function sums the numeric data within a dynamic array as a single value of the next higher level of that dynamic array. That is, if a dynamic array contains multiple attributes, multi-values within any of those attributes and multi-subvalues within some of the values, then the first time the SUM function is called, all sets of multi-subvalues are summed to single numeric values. The second call sums all sets of multi-values to single attributes and the third call sums all attributes to a single number.

The SUM function supports up to three levels of dynamic array resolution. The delimiters used are CHAR(252) through CHAR(254).

CHAR(252) = SubValue represented by "\"
 CHAR(253) = Value represented by "]"
 CHAR(254) = Attribute represented by "^"

Example:

In the following example the dynamic array 'ABC' contains data with attribute marks '^', value marks ']', and subvalue marks '\'.

$$ABC = 10^1 \backslash 2 \backslash 3 \backslash 4 \backslash 5 \backslash 6^7 \backslash 8 \backslash 9 \backslash 10 \backslash 11^12 \backslash 13^14$$

$$SUM(ABC) = 10^3 \backslash 7 \backslash 11^15 \backslash 19 \backslash 11^12 \backslash 13^14$$

$$SUM(SUM(ABC)) = 10^{21} \backslash 45^25^14$$

$$SUM(SUM(SUM(ABC))) = 115$$

5.8.8 SYSTEM

The SYSTEM function returns the value or status of various system functions.

SYSTEM(expression)

where expression is a number from 1 to 41 which specifies one of the system function queries shown below. Explanation of SYSTEM functions 1-20 may be found in the BASIC Reference manual 88A00778A.

<u>Expression</u>	<u>System Function Queried</u>
SYSTEM(20)	Returns system privilege level (0, 1, 2 or 3).
SYSTEM(21)	Returns the character following the escape character as defined in the EDITTRAP statement.
SYSTEM(22)	Reserved.
SYSTEM(23)	Returns 0 if <BREAK> is enabled, returns 1 if <BREAK> disabled from Basic, returns a 2 if <BREAK> disabled from TCL, returns a 3 if <BREAK> disabled from Basic and TCL.
SYSTEM(24)	Allows testing prior to a RETURN statement to determine how the program entered the routine. A 2 is returned if the program is to return from a GOSUB statement. A 1 is returned if the program is to return from a CALL statement. A 0 is returned if there is nothing stacked; if a RETURN was done with nothing stacked, Basic error message "[B27] RETURN executed with no GOSUB or CALL" would be output.
SYSTEM(25)	Returns the current line number that the program is on.
SYSTEM(26)	Returns the name of calling program if in a subroutine. Returns a null if not in a subroutine.
SYSTEM(27)	Returns a 1 if running from Proc, otherwise a 0.
SYSTEM(28)	Returns the system privilege level (0, 1, 2 or 3). Same as SYSTEM(20). Added for Microdata compatibility.
SYSTEM(29)	Reserved.
SYSTEM(30)	Returns a 1 if pagination is active, otherwise a 0.
SYSTEM(31)	Returns data frame size 500 or 2000.
SYSTEM(32-39)	Reserved.
SYSTEM(40)	Returns the name of the program that is currently executing.
SYSTEM(41)	Returns the ABS software release version.

5.8.9 TRIMF

The TRIMF function trims leading blanks from a string. It works identically to the L type TRIM function.

```
Format: var = TRIMF(string)
        or
        var = TRIM(string," ","L")
```

Example programs:

```
TRIMF
001 X=" ABC"
002 A=TRIMF(X)
003 PRINT "Leading Blanks Should Be Trimmed ':'A:'"
```

```
TRIM
001 X=" ABC"
002 A=TRIM(X," ","L")
003 PRINT "Leading Blanks Should Be Trimmed ':'A:'"
```

Both programs will print:

Leading Blanks Should Be Trimmed 'ABC'

5.8.10 TRIMB

The TRIMB function trims trailing blanks from a string. It works identically to the T type TRIM function.

```
Format: var = TRIMB(string)
        or
        var = TRIM(string," ","T")
```

Example programs:

```
TRIMB
001 X="DEF "
002 A=TRIMB(X)
003 PRINT "Trailing Blanks Should Be Trimmed ':'A:'"
```

```
TRIM
001 X="DEF "
002 A=TRIM(X," ","T")
003 PRINT "Trailing Blanks Should Be Trimmed ':'A:'"
```

Both programs will print:

Trailing Blanks Should be Trimmed 'DEF'

multiple tape units 6

A system may have a single cartridge tape unit and up to four 1/2-inch tape units configured. A single user may attach all four 1/2-inch tape units to a process or four users may attach a separate 1/2-inch tape unit to a process or two users may each attach two 1/2-inch tape units, etc. Also, another user may attach the cartridge tape unit. No user may attach both cartridge tape and 1/2-inch tape.

By attaching multiple 1/2-inch tape units to a process, it is possible to execute tape to tape copies or start a multi-reel operation that will access the attached tape drives in the specified order.

If multiple tape units are attached to a process, then the first unit specified during the attachment becomes the PRIMARY tape unit. The primary unit is considered the default unit in most tape related verbs. It should be noted that T-LOAD, T-DUMP, SAVE, RESTORE, SEL-RESTORE, :FILELOAD and Basic all expect to access the primary unit to begin execution of the verb or program, and will cascade to the next specified unit for multi-reel operations only if the cascade table has been setup by the attach verb or the SET-CASCADE verb. No user intervention is required providing the tape units are loaded and online. As each tape is completed, it will rewind and unload from the tape unit. If the next tape unit is "not ready", the system will prompt the user:

Tape unit# n NOT READY

To continue cascading tapes, press (Y)

To continue on primary unit only, press (N)

Press (Y/N):

MOUNT REEL# nn ON UNIT# n AND PRESS 'C' WHEN READY

If 'N' is selected, the cascade table is initialized.

If the number of reels required is more than the number of tape units attached, then the first unit in the cascade table is the next tape unit selected.

The 'U' option of verbs that attach and detach tape requires the user to have SYS2 privileges. The use of this option should be used with caution, as the potential for causing other users problems multiplies with the addition of more tape units to a system; i.e., if USERA has tape units 0,1,2 attached to his process and USERB attaches tape unit 1 with the 'U' option, then USERA will end up with units 0 and 2 attached while expecting to have all three units.

The users primary tape unit will be changed by the verbs SET-MT, T-ATT ,n and SET-CASCADE, or during cascading, where the unit in operation is the primary unit. That is, if an operation completes on the second unit in the cascade table, that unit is the primary unit, or the unit that will be accessed for the next tape operation. To make the first unit in the cascade table the primary unit, use the RESET-CASCADE verb.

It should be noted that each time a user executes a verb that attaches tape, that user's cascade table is cleared. That is, if the user executed SET-MT 4096,0,1 (C, and then executed T-ATT 4096,2; unit 2 becomes the primary tape unit and the cascade table must be setup again with the SET-CASCADE verb. Note that attaching unit 2 will not detach units 0 and 1. T-ATT 16384 changes the block size and attaches the primary unit, but does not clear the cascade table.

NOTE: The verb T-EOF will keep the 1/2-inch tape controller busy, locking out other 1/2-inch tape users until the T-EOF is complete.

6.1 CLEAR-CASCADE

Clears the users cascade table. (1)

6.2 RESET-CASCADE

Sets the primary unit pointer back to the beginning of the cascade table. If cascade table is null, has no effect. Expects the selected device to be 1/2-inch tape. (1)

6.3 SET-CASCADE

Sets or resets the order in which multiple attached 1/2-inch tape units will be accessed for cascading during multi-tape operations. Use of this verb resets the primary unit. If multiple units were attached by SET-MT 4096,2,1 the primary unit is 2 but will not cascade since the C option was not used. Executing SET-CASCADE 1,2 sets the primary unit to 1 and the order of cascade to 1,2. If unit 2 is detached after the cascade table is set, the cascading operation will try to attach unit 2. If unit 2 is not available, the appropriate error message is displayed. If the first unit specified is not attached to the user, error message 86 is output and cascade table is not setup. (1)

The general form of the verb is:

SET-CASCADE n,n,n,n

where: n is physical unit number (0-3)

Example:

SET-CASCADE 0,1 sets cascade table to 0,1. Tape operations will begin on unit 0 and if more than one tape is required and unit 1 is online, unit 1 will be accessed automatically.

6.4 SET-MT

Signifies to the system the user's intention to use 1/2-inch tape. If the specified unit(s) is not already attached to another user, the requested unit(s) is attached to the user's process at a specified or default block size. If one of the requested units is already attached to another process, then the appropriate error message(s) display. If the user has SYS2 privileges, the U option may be used to unconditionally attach the unit(s).
(1)

The general form of the verb is:

SET-MT {blksz}{,n{n{n{n} {U} {C}

where:

blksz any record length greater than 2 and less than or equal to 16384.

n tape unit number. Must be 0, 1, 2, or 3. May be specified in any order. The first unit # specified becomes the "primary" tape unit. If no tape unit number is specified 0 is default.

U attach tape units unconditionally—must have SYS2 privileges.

C set cascade table in the order of attached units.

The default block size immediately after logon is 4096. However, if the user was using cartridge tape at a block size of 512 and then executes the SET-MT verb with default block size, the block size defaults to 512; i.e., the last block size used since logon is default. This verb executed without the C option clears the cascade table. SET-MT detaches any previously attached 1/2" tape unit(s) attached to the user, and attaches only the unit specified.

Examples:

SET-MT ,3 Attach tape unit 3 at the default block size
 (Any previously attached units are detached)

SET-MT 512,3,2 (C Attach tape units 3 and 2 at block size 512. Primary
 tape unit is 3 and the cascade table is set to 3,2.
 Multi-reel operation will automatically fill the tape
 mounted on unit 3, then continue to the tape mounted on
 unit 2 without user intervention, providing the tape
 unit is online.

SET-MT Attach tape unit 0 at a default block size.

SET-MT 16384 U Attach tape unit 0 even if it is attached to another
 process at a block size of 16384. Must have SYS2
 privileges.

6.5 T-ATT

Attempts to attach the default tape unit as defined by the user's last SET-MT or SET-CT command. If SET-CT was previously specified, then the format of the verb is:

T-ATT {n} {U}

where: n specifies block size. Must be 512 or 4096. If not specified
 default block size is used.

U attaches the tape even if another user has it attached. Requires
 SYS2 privileges.

If SET-MT was previously specified, and if the specified unit(s) is not already attached to another user, the requested unit(s) is attached to the user's process at a specified or default block size. If one of the requested units is already attached to another process, the appropriate error message(s) display. If the user has SYS2 privileges, the U option may be used to unconditionally attach the unit(s). This verb clears the cascade table when a unit is attached without the C option. T-ATT does not detach previously attached 1/2" tape units, but changes the primary unit.(1)

The general form of the verb is:

T-ATT {blksz}{,n{n{n{n} {U} {C}

where:

blksz any record length greater than 2 and less than or equal to 16384.

n tape unit number. Must be 0, 1, 2, or 3. May be specified in any order. The first unit # specified becomes the "primary" tape unit. If no tape unit number is specified the last primary or 0 is default.

U attach tape units unconditionally—must have SYS2 privileges.

C set cascade table in the order of attached units.

The default block size immediately after logon is 4096. However, if the user was previously using a block size of 512 and then executes the T-ATT verb with default block size, the block size defaults to 512; i.e., the last block size used since logon is default.

Examples assume SET-MT:

T-ATT ,3 Attach tape unit 3 at the default block size.

T-ATT 16384,2 U Attach tape unit 2 even if it is attached to another process at a block size of 16384. Must have SYS2 privileges.

T-ATT 512,3,2 (C Attach tape units 3 and 2 at block size 512. Primary tape unit is 3 and cascade table is set to 3,2. Multi-reel operation will automatically fill the tape mounted on unit 3, then continue on unit 2 without user intervention, providing unit 2 is online.

T-ATT Attach last primary tape unit at default block size.

6.6 T-COPY

For T-COPY to function, at least two 1/2-inch tape units must be installed in the system. This verb will copy all files or a specified number of files to a specified unit at the same block size or to a different block size. Only two tape units, source and destination, will be used for T-COPY regardless of the number of tape units attached. T-COPY will handle multi-reel operations but the user must mount the next tape, whether it be a source or destination tape, on the appropriate source or destination tape unit. (1)

The general format of the verb is:

T-COPY source.unit#,destination.unit# {,#files}{,blksz} {(options)}

where:

source.unit# = specifies the source unit (0,1,2,3). Cannot be the same as the destination unit.

destination.unit# = specifies the destination unit (0,1,2,3). Cannot be the same as the source unit.

#files = specifies the number of files to copy from the source unit to the destination unit. Default is 1 if 'E' option is not specified.

blksz = specifies the new block size data is copied to on the destination tape. May be larger or smaller blocks than the source tape. Default, if not specified, is the same as source tape.

If block size = 4096 on source tape and new block size for destination tape is 16384, then four source tape blocks are packed into one destination tape block.

If block size = 16384 on source tape and new block size for destination tape is 4096, then one source tape block is unpacked into four destination tape blocks.

Options:

- E specifies that every file is copied until the end of data is encountered. EOD = 2 consecutive EOF marks.
- L display tape labels to terminal as they are encountered.
- U T-UNLOAD the destination tape unit when copy is completed.

Examples:

T-COPY 0,1 (EU copy all files of source tape on unit 0 to destination tape on unit 1 at the same block size as source tape, and unload tape unit 1 when the copy is complete.

T-COPY 0,2,,4096 copy 1 tape file from tape unit 0 to tape unit 2 at a new block size of 4096.

T-COPY 0,1,,8192 (EL copy all files from source tape to unit 1 at a new block size of 8192, and list the labels encountered on the source tape.

6.7 T-DET

Detaches tape unit from the user's line. Optional 'U' detaches device from any other user's line except the Spooler. U option requires SYS2 privileges. If multiple 1/2-inch tapes are attached, then the user must specify which tape unit(s) to detach. If no unit is specified, all tape units are detached. (1)

T-DET {n}{,n{,n{,n} {U}

where:

n tape unit number. Must be 0, 1, 2 or 3.

U detach tape units unconditionally—must have SYS2 privileges.

Examples:

T-DET detach all tape unit(s) from the user's line.

T-DET 3 U detach tape unit 3 from another user's process. Must have SYS2 privileges.

T-DET 1,2 detach tape units 1 and 2 from the users process.

6.8 T-REW

Rewinds tape to load point. On 1/4-inch tape, causes last tape buffer to be written to tape before rewind. When used with 1/2-inch tape, the tape unit number must be supplied or the primary tape unit is default. (1)

T-REW {n{,n{n{n{,n}

where: n specifies 1/2-inch tape unit. Must be 0, 1, 2 or 3. If 1/4-inch tape is attached 'n' is ignored.

Example 1/4-inch tape:

T-REW rewinds tape to load point, and if data was being written to tape, flushes last tape buffer to tape before rewinding.

Example 1/2-inch tape:

T-REW rewinds primary tape unit to load point.

T-REW 2 rewinds tape unit number 2 to load point.

T-REW 1,2 rewinds tape units 1 and 2 to load point.

6.9 T-STATUS

Displays status information for available devices. Tells which devices are attached and gives the line numbers using the devices, or reports device availability. Also, displays the primary unit and cascade table for the user executing the verb.

Example:

[32] Mag tape unit 0 is attached to line 25 (Primary unit).

[32] Mag tape unit 1 is attached to line 25.

[32] Mag tape unit 2 is attached to line 126.

[32] Mag tape unit 3 is available.

[33] Cartridge tape unit 0 is available.

[34] Cartridge disk unit 0 is available.

[56] Floppy disk unit 0 is available.

[27] Cascading order for your line is: 0,1,..

6.10 T-UNLOAD

Rewinds and unloads 1/2-inch tape from the tape unit. After the tape has been loaded and processed, this verb allows the user to rewind and physically remove the tape without pressing the ON-LINE or RESET and UNLOAD buttons on the tape unit. If the unit specified is attached to another user, the verb is ignored. If cartridge tape or disk are attached, a rewind only occurs. (1)

T-UNLOAD {n{,n{,n{,n{

where:

n tape unit number to rewind and unload tape. Must be 0, 1, 2, or 3.

Default is primary tape unit if no tape unit number specified.

Example:

If tape units were attached T-ATT 4096,3,2,1; then the primary tape unit is 3.

T-UNLOAD primary tape unit (3) is rewound and unloaded.

T-UNLOAD 2,1 rewind and unload tape units 2 and 1.

SMA standard 1/4-inch tape

7

The addition of SMA standard format for 1/4-inch tape, to the GA PICK Operating System, allows single reel 1/4-inch tape to be exchanged between vendors. There is no SMA standard for multi-reel 1/4-inch tape. Therefore, the ability to exchange multi-reel SMA standard 1/4-inch tape between vendors is not to be relied upon. Multi-reel SMA format tapes between GA Zebra models will function correctly.

With the addition of SMA format, there are three supported 1/4-inch tape formats, "Old," "Normal" (formerly called "NEW"), and "SMA" standard. All GA models running the 3.8 version of the Pick Operating System can read and write "Normal" and "SMA" standard format. The 1350, 1750 and 3750 models can read "Old" format; but cannot write "Old" format. "Normal" format, is the system default for 1/4-inch tape. The format is selected by and option of the T-FORMAT verb.

The differences between "SMA" format and "Normal" format are noted below.

- Labels occupy only 512 bytes on tape, rather than using a full virtual block of tape.
- SMA format generated on Zebra machines are forced to a virtual block size of 8192, as recommended by SMA for maximum portability.
- Zebra machines can read SMA format tapes generated at a block size other than 8192, providing the following two conditions are met.
 1. The block size is evenly divisible by 512.
 2. The block size is no greater than 24576.

Note that even though the tape may be generated at a block size other than 8192, the tapes will still be attached at 8192, and all messages will reflect a block size of 8192.
- Issuing a T-REW command to "flush" the temporary buffer is not required for SMA format tapes. The flush is explicitly performed by virtual. T-REW is used only to rewind the tape to Beginning of Tape (BOT).
- Physical file marks, not pseudo file marks, are used in SMA format.
- Tape writes are terminated by one physical file mark, rather than two (and backspace one) as is standard with the "Normal" format and 1/2-inch tape. This is due to the limitations of the drive, which cannot backspace.

- The T-ASSIGN verb is not necessary when using "SMA" format. While in "SMA" format, End of Tape (EOT) is detected, the tape will rewind, and the user will be prompted for the next reel. This will allow any length of 1/4-inch tape to be used. Both "Old" and "Normal" format write an explicit number of blocks to the tape depending on the options of the T-ASSIGN verb. The only tape lengths allowed for "Old" and "Normal" format is 450 ft. and 600 ft. tapes.

The verbs that follow, list only the verbs and definitions as they have changed in relation to "SMA" format. If "SMA" did not change or add to the definition of the verb, the verb is not listed here.

7.1 T-FORMAT

T-FORMAT assigns the format in which to read or write 1/4-inch tape. If no option is used, the current option assignment is displayed. The system default at boot time is "Normal" format. Once the format is set by the user for a particular line, the format will remain in effect for that line until explicitly changed. Logging off will not change the format.(1)

The general form of the verb is:

T-FORMAT {(option)}

Options:

- S Set "SMA" format for read and write
- N Set "Normal" format for read and write
- O Set "Old" format for read only on 1350, 1750 and 3750

7.1.1 T-ATT

If T-FORMAT with the S-option precedes T-ATT, the block size is forced to 8192 even though another block size has been entered.

7.1.2 T-CHK

If T-FORMAT with the S-option precedes T-CHK with A-option, then tape is checked for parity errors until end of data (EOD) is encountered. EOD is two physical file marks.

7.1.3 T-EOD

If T-FORMAT with the S-option in effect, moves 1/4-inch tape to end of data position.

7.1.4 T-EOFD

If T-FORMAT with the S-option in effect, moves 1/4-inch tape to end of data position.

7.1.5 T-REW

If T-FORMAT with the S-option in effect, rewinds 1/4-inch tape to beginning of tape position.

7.1.6 T-WEOF

If T-FORMAT with the S-option in effect, causes a physical file mark to be written to 1/4-inch tape immediately, does not wait for T-REW.

floppy disk 8

The integration of floppy disk into the GA Pick Operating System allows the user to interchange media with personal computers, AT or XT type, that use the PICK operating system. The implementation is SMA magnetic media interchange standard, which treats the floppy disk as if it were tape. In other words, using disk in a tape emulation mode. Most of the verbs pertaining to the 1/2-inch mag tape utility apply to floppy disk.

It should be noted that Pick uses 500 byte blocks. An option is available that allows use of the full 512 bytes of the floppy sector. Using the full 512 bytes is not PICK compatible but may allow the user to write programs that read or write other operating system formats.

The floppy peripheral device has a density switch that allows the user to read/write either high density, double sided disks (PC-AT type); or double density, double sided disks (PC-XT type). The user must be sure the density switch, floppy disk type, and SET-FD verb, all match when trying to format, or read/write a disk.

PC-AT Type:

Double sided
HIGH density
96tpi
80 tracks
15 sectors/track
512 byte sectors

PC-XT Type:

Double sided
Double density
48tpi
40 tracks
9 sectors/track
512 byte sectors

PICK compatible floppies must format completely with NO errors. If format errors are encountered, data corruption will occur.

8.1 FORMAT-FD

The FORMAT-FD verb formats a floppy as specified by the SET-FD verb. Formatting initializes all the data on the floppy with an "EE" pattern one track at a time. After a track is formatted each sector is read back to insure it can at least be read. After each sector is read in a period "." is displayed. If a sector cannot be read back after four attempts a percent "%" is displayed and the format continues. When finished all errors will be displayed. PICK compatible floppies must format without errors or data corruption may occur when data is written to the disk. (1)

The general form of the verb is:

FORMAT-FD {(N,V,P,n1{-n2})}

Options:

- N No read. The read after track format is inhibited and any possible read errors will not be detected.
- V Verify. The track formatting is inhibited and each sector is verified that it can be read. See VERIFY-FD verb.
- P Print all detected errors to the Spooler.
- n1 Starting track number to format.
- n2 Ending track number to format.

8.2 SET-FADDR

The SET-FADDR verb will set the sector to be read in when a tape operation is initiated. (1)

The general form of the verb is:

SET-FADDR n

Option:

- n Sector number to read.

8.3 SET-FD

The SET-FD verb attaches the floppy disk drive as a tape unit. The unit can be setup in two PICK compatible modes. Default setup is PC-AT type with a 500 byte block size. The optional PICK compatible setup is PC-XT type, also with a 500 byte block size. A third option, not PICK compatible, sets a 512 byte block size. All options of this verb must be used in conjunction with the floppy density switch on the floppy unit.(1)

PC-AT Type:

Double sided
HIGH density
96tpi
80 tracks
15 sectors/track
512 byte sectors

PC-XT Type:

Double sided
Double density
48tpi
40 tracks
9 sectors/track
512 byte sectors

The general form of the verb is:

SET-FD {(W,X)}

Options:

- W Whole sector. Allows the use of the full 512 bytes on a floppy sector. THIS IS NOT PICK COMPATIBLE AND CANNOT BE USED WITH T-DUMPS, T-LOADS, SAVE AND RESTORE.
- X XT compatible. Initializes the system format, read or write double density floppies. It must be used in conjunction with the floppy density switch in the double density (low) mode.

8.4 VERIFY-FD

The VERIFY-FD verb will verify that all sectors on the floppy can be read. After each sector is read, a period "." is displayed. If a sector cannot be read back after four attempts a percent "%" is displayed and the verify continues. When finished all errors will be displayed.(1)

The general form of the verb is:

VERIFY-FD {(P,{n1{-n2}})}

Options:

- P Print all detected errors to the Spooler.
- n1 Starting track number to verify.
- n2 Ending track number to verify.

protocol 9

PROTOCOL is expected to function correctly on those ZEBRA models that use Octal Serial Driver Board 31-03476A or 31-03606A or Octal Serial Interface board 77A00224A. Those systems using OSI 77A00224A may require jumpers to be inserted or moved on the board before some PROTOCOL functions will work correctly. Models 1350, 1750, and 3750 do not support all Protocol functions and some functions are available on ports 4 or 5 only and are so noted in the following documentation.

It should be noted that if no cable or an open end cable is connected to a port that has enabled hardware handshaking or auto logoff, the controlling input line to the system (CTS, DCD or DTR) may drift into a state which looks "true" to the software. This will cause erroneous operations and may cause constant interrupts to the system. Constant interrupts may slow the system down to a point where it looks as though all ports on the system are in a "hang" condition. Therefore, careful management of the PROTOCOL verb and Basic functions is a must.

9.1 PROTOCOL VERB

The PROTOCOL verb allows the user to manipulate and display the input and output characteristics of the system's ports.

Format:

PROTOCOL {a,b,c,d,e,f,g,h,i,j,k,l,m,o,p} {(n,Z,P)}

a Baud rate specification; default = 9600

Available rates: 75, 110, 134, 150, 300, 600, 1200, 1800,
2000, 2400, 4800, 9600, and 19200.

Note: A change in baud rate will cause erroneous characters to be printed until the baud rate on the attached device is changed to a matching baud rate.

b Parity specification; default = M

Options: M - mark no parity
E - even parity
O - odd parity

Note: Parity is generated for output data and is not checked on incoming data.

c Data bits specification; default = 8

Options: 5, 6, 7, 8

d Stop bits specification; default = 1

Options: 1, 1.5, 2

e Type ahead buffer; default = (Y)es

Options: (Y)es will enable the type ahead buffer
(N)o will disable the type ahead buffer

When enabled, data is put into the input buffer until the buffer is filled. If the input buffer is filled and data is received then an OVER-RUN warning character (if enabled) will be sent (echoed) and the received character is discarded.

When disabled, received data is only allowed when the system requires data. If data is received before the system requires it an OVER-RUN warning character (if enabled) will be sent (echoed) and the received data is discarded.

Note: The type ahead buffer is automatically ENABLED and cannot be turned off when receiver flow is ENABLED.

- f Receiver flow control; default = (N)o
 Options: (Y)es will enable receiver flow control
 (N)o will disable receiver flow control

SOFTWARE FLOW CONTROL:

When enabled, all incoming data is passed into OUR input buffer and OUR system will send an XOFF character to the inputting source when OUR input buffer is almost full. An XON will be sent to the inputting source when OUR input buffer is almost empty. If characters are still being received after the first XOFF, then XOFF's will be sent and any received character will be put into OUR input buffer until the buffer is full. If OUR input buffer is full and characters are still being received, XOFF's will be sent and the characters will be thrown away.

When disabled, an OVER-RUN warning character will be sent (if enabled) when a character has been received and OUR input buffer is full. The character will be discarded.

- Note: The type ahead buffer is automatically ENABLED and cannot be turned off when receiver flow control is ENABLED.

HARDWARE FLOW CONTROL:

When enabled, all incoming data is passed into OUR input buffer and OUR system will de-assert the RTS line when OUR input buffer is almost full. The RTS line will be re-asserted when OUR input buffer is almost emptied. While the RTS line is de-asserted NO other data will be recognized until the RTS line is re-asserted.

When disabled, an OVER-RUN warning character will be sent (if enabled) when a character has been received and OUR input buffer is full. The character will be discarded.

- Note: For ZEBRA 1350, 1750, 3750, Receiver Hardware Flow Control functions on Port 5 only.

- Note: OSI Board 77A00224A requires jumper for the designated port at label CTS be moved from INT to EXT for both RTS and CTS hardware flow control.

- g Transmitter flow control; default = (Y)es
 Options: (Y)es will enable transmitter flow control
 (N)o will disable transmitter flow control

SOFTWARE FLOW CONTROL:

When enabled, OUR receiver will be filtering incoming data for XON and XOFF characters to throttle OUR transmission of characters from OUR output buffer. A received XOFF will cause the transmission of characters to stop while a received XON will cause transmission to continue (this is valid only after an XOFF has been received). In all cases, the XON and XOFF characters are NOT put into OUR input buffer.

When disabled, the XON and XOFF characters will be passed into OUR input buffer if it is NOT full.

HARDWARE FLOW CONTROL:

When enabled, OUR transmitter will send data from OUR output buffer IF the CTS line is asserted. If the CTS line is de-asserted, OUR transmitter will NOT send ANY data.

When disabled, OUR transmitter will send data from OUR output buffer regardless of the state of the CTS line.

- Note: If hardware flow control is enabled ALL XON and XOFF characters will be passed into OUR input buffer regardless of this option. See specification 'p'.
- Note: For models 1350, 1750 and 3750, transmitter Hardware Flow Control functions on port 5 only.
- Note: OSI Board 77A00224A requires jumper for the designated port at label CTS be moved from INT to EXT for both RTS and CTS hardware flow control.

- h Receiver over-run warning; default = (Y)es
 Options: (Y)es will enable over-run warning
 (N)o will disable over-run warning

When enabled, an OVER-RUN warning character will be sent when buffer FULL conditions arise AND receiver flow control is NOT enabled.

When disabled, OR receiver flow control is enabled, an OVER-RUN warning character will NOT be sent when input buffer FULL conditions arise.

- i Receiver over-run warning character; default = X'07'
Options: Any single hexadecimal representation
of an ASCII character

This character will be transmitted whenever an OVER-RUN warning character is required.

- j 8th data bit enable; default = (N)o
Options: (Y)es will enable the function
(N)o will disable the function.

When enabled, all characters received will retain the state of the eighth data bit.

When disabled, all characters received will have the eighth data bit cleared.

- k Auto logoff via DTR; default = (N)o
Options: (Y)es will enable auto logoff via DTR
(N)o will disable auto logoff via DTR

When enabled, the de-assertion of the DTR line will cause the process associated with the UART to be logged off. While auto logoff is in effect, no data will be transmitted or received. Data will be allowed to be received or transmitted only after the DTR line is re-asserted, auto logoff is disabled or auto logoff is changed to DCD. Specification '1' is disabled automatically when this control is enabled.

When disabled, the DTR line is ignored and LOGOFF must be done through normal means.

Note: Auto logoff is intended for lines that can be logged on only (i.e. terminals or modems). This function is NOT intended for printers.

Note: Auto logoff via DTR is not available on 1350, 1750 or 3750.

Note: OSI Board 77A00224A requires jumper at label DSR/DCD be inserted. This enables auto logoff via DTR or DCD for all ports on the board. DTR or DCD is selected via the PROTOCOL verb or function.

- 1 Auto logoff via DCD; default = (N)o
 Options: (Y)es will enable auto logoff via DCD
 (N)o will disable auto logoff via DCD

When enabled, the de-assertion of the DCD line will cause the process associated with that UART to be logged off. While auto logoff is in effect no data will be transmitted or received. Data will be allowed to be received or transmitted only after the DCD line is re-asserted, auto logoff is disabled or auto logoff is changed to DTR. Specification 'k' is disabled automatically when this control is enabled.

When disabled, the DCD line is ignored and LOGOFF must be done through normal means.

Note: Auto logoff is intended for lines that can be logged on only (i.e. terminals or modems). This function is NOT intended for printers.

Note: For models 1350, 1750 and 3750, Auto logoff via DCD functions on port 5 only.

Note: OSI Board 77A00224A requires jumper at label DSR/DCD be inserted. This enables auto logoff via DTR or DCD for all ports on the board. DTR or DCD is selected via the PROTOCOL verb or function.

- m Software flow control ON character--XON; default = X'11"
 Options: Any single hexadecimal representation
 of an ASCII character

This character will be used whenever an XON character is to be sent or when an XON character is to be detected during SOFTWARE flow control.

- o Software flow control OFF character--XOFF; default = X'13'
 Options: Any single hexadecimal representation
 of an ASCII character

This character will be used whenever an XOFF character is to be sent or when the XOFF character is to be detected during SOFTWARE flow control.

- p Flow control; default = (S)oftware flow control enable
 Options: (H)ardware flow control enable
 (S)oftware flow control enable
 (N)o flow control enable

This specification governs the functionality of the receiver and/or transmitter flow control, specifications 'f' and 'g' respectively.

If NO FLOW CONTROL is enabled, then specification 'f' and 'g' are both treated as disabled regardless of their current state.

Note: Since OSI Board 77A00224A is physically jumpered at label CTS, (N)o flow control is required for hardware flow control. Thus, if (H) for hardware flow control is selected it will show as (N) for no flow control.

Note: Hardware flow control for ZEBRA 1350, 1750 and 3750 is available on Port 5 only.

- n Port number. Indicates specifications to be changed for this port or port specifications to be displayed. Requires SYS2 privileges to view or change port other than the current port.

Specifications 'a' through 'p' may be entered in full by placing a comma after each specification or in part by using commas with no specifications as holders (similar to the TERM statement). If no options are specified, then the status of each specification for the current port will be displayed.

- Z Will display specifications for all ports to screen.
 PROTOCOL (Z

- P Will force the screen output to spooler.
 PROTOCOL (Z,P

9.1.1 EXAMPLE PROTOCOL DISPLAY

>PROTOCOL (Z

Line	Port	Baud	P	Data	Stop	T	RTX	LTX	OVRN	OVRN	8th	DTR	DCD	XON	XOF	Flow	ctl
	type	rate	R	bits	bits	A	ctl	ctl	enbl	char	bit	clt	ctl	chr	chr		type
0	DCE	9600	M	8	1	Y	N	Y	Y	07	N	N	N	11	13	Soft	
1	DCE	9600	M	8	1	Y	N	Y	Y	07	N	N	N	11	13	Soft	
2	DCE	9600	M	8	1	Y	N	Y	Y	07	N	N	N	11	13	Soft	
* 3	DCE	9600	M	8	1	Y	N	Y	Y	07	N	N	N	11	13	Soft	
4	DCE	9600	M	8	1	Y	N	Y	Y	07	N	N	N	11	13	Soft	
5	DCE	9600	M	8	1	Y	N	Y	Y	07	N	N	N	11	13	Soft	
6	DCE	9600	M	8	1	Y	N	Y	Y	07	N	N	N	11	13	Soft	
7	DCE	9600	M	8	1	Y	N	Y	Y	07	N	N	N	11	13	Soft	

9.2 BASIC PROTOCOL FUNCTIONS

Allows the user to sense or change the specified parameter (n) for the current system port. Not all parameters may be changed. Whether a parameter is sensed or changed, depends on the syntax used.

Syntax to sense a parameter is:

var = PROTOCOL(n)

Syntax to change a parameter is:

PROTOCOL(n,var)

Where n is a number from 1 to 23; defined below.

n# Function

- 1 Not used.
- 2 Will sense or change the baud rate for this line. The default is 9600. See PROTOCOL verb (specification a).
- 3 Will sense or change the parity for this line. The default is M. M=Mark, O=Odd, E=Even. See PROTOCOL verb (specification b).
- 4 Will sense or change the number of data bits for this line. The default is 8. See PROTOCOL verb (specification c).
- 5 Will sense or change the number of stop bits for this line. The default is 1. See PROTOCOL verb (specification d).
- 6 Will sense, enable, or disable the type ahead function for this line. Can be 0 or 1. The default is 1 for enabled. See PROTOCOL verb (specification e).
- 7 Will sense, enable or disable Receiver flow control for this line. Can be 0 or 1. The default is 0 for disabled. Hardware flow control for ZEBRA 1350, 1750 and 3750 is available on Port 5 only. See PROTOCOL verb (specification f).
- 8 Will sense, enable, or disable Transmitter flow control for this line. Can be 0 or 1. The default is 1 for enabled. Hardware flow control for ZEBRA 1350, 1750 and 3750 is available on Port 5 only. See PROTOCOL verb (specification g).
- 9 Will sense, enable, or disable the OVER-RUN warning for this line. Can be 0 or 1. The default is 1 to allow the OVER-RUN warning character to be transmitted when an input buffer full condition has occurred. See PROTOCOL verb (specification h).

- 10 Will sense or change the OVER-RUN warning character for this line. Can be any hex byte. The default is X'07'. See PROTOCOL verb (specification i). i.e., PROTOCOL (10,CHAR(07)).
- 11 Will sense, enable, or disable the stripping of 8th data bit on received data. Can be 0 or 1. The default is 0. See PROTOCOL verb (specification j).
- 12 Will sense, enable, or disable auto logoff via DTR. Can be 0 or 1. The default is 0 for disabled. See PROTOCOL verb (specification k). Not available for 750, 1350, 1750 or 3750.
- 13 Will sense, enable, or disable auto logoff via DCD. Can be 0 or 1. The default is 0 for disabled. See PROTOCOL verb (specification l). For model 750, 1350, 1750, and 3750, available on port 5 only.
- 14 Will sense or change the flow control ON character (XON) for this line. Can be any hex byte. The default is X'11'. See PROTOCOL verb (specification m). i.e. PROTOCOL(14,CHAR(17)).
- 15 Will sense or change the flow control OFF character (XOFF) for this line. Can be any hex byte. The default is X'13'. See PROTOCOL verb (specification o). i.e. PROTOCOL(14,CHAR(19)).
- 16 Will sense or change the flow control TYPE for this line. The default is 1. 1=Software enable, 2=Hardware enable, 0=No flow control. Hardware flow control for ZEBRA 1350, 1750 and 3750 is available on Port 5 only. See PROTOCOL verb (specification p).
- 17 Not used.
- 18 Will sense DCD for this line. Will return a 1 for asserted and a 0 for de-asserted.
- 19 Will allow a single character read or write from Basic without going through TERMIO. Characters such as CTRL-W, CTRL-X, CTRL-R, and Backspace, are passed through. Therefore, screen editing must be done by the Basic program. However, the characters CTRL-S and CTRL-Q (XOFF/XON) are not passed through.
- 20 Will sense, assert or de-assert RTS (System Pin 8) for this line. Will return 1 for asserted and 0 for de-asserted. ZEBRA 1350, 1750 and 3750 will sense only.

Note: If Hardware flow control (PROTOCOL 16) is enabled and Receiver flow control (PROTOCOL 7) is enabled, then this line can only be sensed.

- 21 Will sense CTS (System Pin 7) for this line. Will return 1 for asserted and 0 for de-asserted. For ZEBRA 1350, 1750 and 3750, available on ports 4 and 5 only.
- 22 Will sense DTR (System Pin 9) for this line. Will return 1 for asserted and 0 for de-asserted. For models 1350, 1750 or 3750, available on ports 4 and 5 only.
- 23 Will sense, assert or de-assert DSR (System Pin 2) for this line. Will return 1 for asserted and 0 for de-asserted. For ZEBRA 1350, 1750, or 3750, available on ports 4 and 5 (sense only).

9.3 PROTOCOL CABLING

LAYOUT FOR STANDARD 9-PIN TO 25-PIN CABLES

Signal	SYSTEM (9-PIN) Pin No.	Direction	Direction	RECEIVER (25-PIN) Pin No.	Signal
Shield	1				
DSR	2	Output -->	Input --->	6	DSR
RXD	3	Input <---	Output <--	2	TXD
TXD	4	Output -->	Input --->	3	RXD
GND	5	---	---	7	GND
DCD	6	Input <---	Output <--	8	DCD
CTS	7	Input <---	Output <--	4	RTS
RTS	8	Output -->	Input --->	5	CTS
DTR	9	Input <---	Output <--	20	DTR

SOFTWARE FLOW CONTROL USING XON/XOFF

Signal	SYSTEM (9-PIN) Pin No.	Direction	Direction	RECEIVER (25-PIN) Pin No.	Signal
Shield	1				
DSR	---	---	---	---	---
RXD	3	Input <---	Output <--	2	TXD
TXD	4	Output -->	Input --->	3	RXD
GND	5	-----	-----	7	GND
DCD	---	---	---	---	---
CTS	---	---	---	---	---
RTS	---	---	---	---	---
DTR	---	---	---	---	---

Example of PROTOCOL verb for above cable connection:

>PROTOCOL 9600,M,8,1,Y,N,Y,Y,07,N,N,N,11,13,S (3

HARDWARE FLOW CONTROL USING DTR

Signal	SYSTEM (9-PIN) Pin No.	Direction	Direction	RECEIVER (25-PIN) Pin No.	Signal
Shield	1				
DSR	---	---	---	---	---
RXD	3	Input <---	Output <--	2	TXD
TXD	4	Output -->	Input -->	3	RXD
GND	5	-----	-----	7	GND
DCD	---	---	---	---	---
CTS	7	Input <---	Output <--	20	DTR
RTS	---	---	---	---	---
DTR	---	---	---	---	---

Example of PROTOCOL verb for above cable connection:

>PROTOCOL 9600,M,8,1,Y,N,Y,Y,07,N,N,N,11,13,H (3

HARDWARE FLOW CONTROL USING CTS

Signal	SYSTEM (9-PIN) Pin No.	Direction	Direction	RECEIVER (25-PIN) Pin No.	Signal
Shield	1				
DSR	---	---	---	---	---
RXD	3	Input <---	Output <--	2	TXD
TXD	4	Output -->	Input -->	3	RXD
GND	5	-----	-----	7	GND
DCD	---	---	---	---	---
CTS	7	Input <---	Output <--	4	RTS
RTS	---	---	---	---	---
DTR	---	---	---	---	---

Example of PROTOCOL verb for above cable connection:

>PROTOCOL 9600,M,8,1,Y,N,Y,Y,07,N,N,N,11,13,H (3

AUTO LOGOFF USING DTR

Signal	SYSTEM (9-PIN) Pin No.	Direction	Direction	RECEIVER (25-PIN) Pin No.	Signal
Shield	1				
DSR	---	---	---	---	---
RXD	3	Input <--	Output <--	2	TXD
TXD	4	Output -->	Input -->	3	RXD
GND	5	-----	-----	7	GND
DCD	---	---	---	---	---
CTS	---	---	---	---	---
RTS	---	---	---	---	---
DTR*	9	Input <--	Output <--	20	DTR

* When signal on pin 9 is LOW,
auto logoff occurs.

Example of PROTOCOL verb for above cable connection:

>PROTOCOL 9600,M,8,1,Y,N,Y,Y,07,N,Y,N,11,13,S (3

AUTO LOGOFF USING DCD

Signal	SYSTEM (9-PIN) Pin No.	Direction	Direction	RECEIVER (25-PIN) Pin No.	Signal
Shield	1				
DSR	---	---	---	---	---
RXD	3	Input <--	Output <--	2	TXD
TXD	4	Output -->	Input -->	3	RXD
GND	5	-----	-----	7	GND
DCD*	6	Input <--	Output <--	8	DCD
CTS	---	---	---	---	---
RTS	---	---	---	---	---
DTR	---	---	---	---	---

*When signal on pin 6 is HIGH,
auto logoff occurs.

Example of PROTOCOL verb for above cable connection:

>PROTOCOL 9600,M,8,1,Y,N,Y,Y,07,N,N,Y,11,13,S (3

750/1350/1750/3750 CABLE LAYOUT FOR PORT 5
25-PIN TO 25-PIN CABLES

Signal	SYSTEM (25-PIN) Pin No.	Direction	Direction	RECEIVER (25-PIN) Pin No.	Signal
Shield	1	-----			
TXD	2	Output ->	Input -->	3	RXD
RXD	3	Input <---	Output <--	2	TXD
RTS	4	Output ->	Input -->	5	CTS
CTS	5	Input <---	Output <--	4	RTS
DSR	6	Output ->	Input -->	6	DSR
GND	7	-----	-----	7	GND
DCD	8	Input <---	Output <--	8	DCD
DTR	20	Input <---	Output <--	20	DTR

750/1350/1750/3750--PORT 5
SOFTWARE FLOW CONTROL USING XON/XOFF

Signal	SYSTEM (25-PIN) Pin No.	Direction	Direction	RECEIVER (25-PIN) Pin No.	Signal
Shield	1	-----			
TXD	2	Output ->	Input -->	3	RXD
RXD	3	Input <---	Output <--	2	TXD
RTS	---	---	---	---	---
CTS	---	---	---	---	---
DSR	---	---	---	---	---
GND	7	-----	-----	7	GND
DCD	---	---	---	---	---
DTR	---	---	---	---	---

Example of PROTOCOL verb for above cable connection:

>PROTOCOL 9600,M,8,1,Y,N,Y,Y,07,N,N,N,11,13,S (5

750/1350/1750/3750---PORT 5

HARDWARE FLOW CONTROL USING DTR

Signal	SYSTEM (25-PIN) Pin No.	Direction	Direction	RECEIVER (25-PIN) Pin No.	Signal
Shield	1	---			
TXD	2	Output -->	Input -->	3	RXD
RXD	3	Input <---	Output <--	2	TXD
RTS	---	---	---	---	---
CTS	5	Input <---	Output <--	20	DTR
DSR	---	---	---	---	---
GND	7	-----	-----	7	GND
DCD	---	---	---	---	---
DTR	---	---	---	---	---

Example of PROTOCOL verb for above cable connection:

>PROTOCOL 9600,M,8,1,Y,N,Y,Y,07,N,N,N,11,13,H (5

750/1350/1750/3750---PORT 5

HARDWARE FLOW CONTROL USING CTS

Signal	SYSTEM (25-PIN) Pin No.	Direction	Direction	RECEIVER (25-PIN) Pin No.	Signal
Shield	1	---			
TXD	2	Output -->	Input -->	3	RXD
RXD	3	Input <---	Output <--	2	TXD
RTS	---	---	---	---	---
CTS	5	Input <---	Output <--	4	RTS
DSR	---	---	---	---	---
GND	7	-----	-----	7	GND
DCD	---	---	---	---	---
DTR	---	---	---	---	---

Example of PROTOCOL verb for above cable connection:

>PROTOCOL 9600,M,8,1,Y,N,Y,Y,07,N,N,N,11,13,H (5

750/1350/1750/3750---PORT 5
 AUTO LOGOFF USING DCD

SYSTEM (25-PIN)			RECEIVER (25-PIN)		
Signal	Pin No.	Direction	Direction	Pin No.	Signal
Shield	1	-----			
TXD	2	Output -->	Input --->	3	RXD
RXD	3	Input <---	Output <--	2	TXD
RTS	---	---	---	---	---
CTS	---	---	---	---	---
DSR	---	---	---	---	---
GND	7	-----	-----	7	GND
DCD*	8	Input <---	Output <--	8	DCD
DTR	---	---	---	---	---

*When signal on pin 6 is HIGH,
 auto loggoff occurs.

Example of PROTOCOL verb for above cable connection:

>PROTOCOL 9600,M,8,1,Y,N,Y,Y,07,N,N,Y,11,13,S (5

system verbs and procs 10

The number or letter in parentheses following verb or proc indicates the minimum system privilege level required for its use. Levels are 0, 1, 2, 3 and S. The lowest privilege is 0, and the highest, 3. The S indicates that the utility is found only in the SYSPROG account.

10.1 :BTI AND :ITB VERBS

The :BTI and :ITB verbs have been added to enable the conversion of linked binary items to or from normal items. (2)

10.1.1 :BTI VERB

:BTI converts linked binary items into normal items. If the binary item contains a segment mark, the normal item is terminated at the segment mark. Anything after the segment mark is lost, as it is possible that binary items would have multiple segment marks. If not, the last character of the binary item is changed to a segment mark to terminate the item. This results in a character being lost. This verb may be useful after loading a tape produced on a foreign machine using RESTORE with the "Y" option. The "Y" option loads CL items as linked binary, rather than converting them to normal items. (2)

10.1.2 :ITB VERB

:ITB converts normal items into linked binary items. The verb is used whenever a normal item needs to be converted to linked binary format. (2)

10.2 :SET-XOVR

:SET-XOVR sets the crossover point used by the file update routine. Normal items may be stored entirely in the file's "group", or indirectly, with a pointer in the "group" pointing to the item stored in its own space. The storage format is completely transparent to the user, and only effects file sizing, disk space usage, and system performance. System default is 32767 minus data frame size. The 32K default is useful when the user needs to have file compatibility between 3.8 and earlier versions of the GA Pick operating system. A smaller crossover point, about 70-80 per cent of the data frame size, is more desirable as it will allow various system processes to run slightly faster. Some of the processes that take advantage of the smaller crossover size are: SAVE, RESTORE, COPY and BASIC statements, SELECT and READNEXT.(S)

Format of the verb is:

```
:SET-XOVR {nnn}    (nnn = number of bytes between 250 and 32767
                    minus data frame size) (If linked frame size is
                    512 bytes, data frame size is 500, thus maxi-
                    mum number of bytes is 32267. If linked frame
                    size is 2048, data frame size is 2000;
                    therefore, maximum number of bytes would be
                    30767.)
```

```
:SET-XOVR 400
[78] Crossover size set to 400 bytes.
```

```
:SET-XOVR
[77] Current crossover size is 400 bytes.
```

10.3 :SWC

The :SWC verb changes the file type to DC. If the file type was DX, DY or DZ, then it is changed to DCX, DCY or DCZ respectively. Once a file type has been changed to DC, the "C" cannot be removed. The only way to change a file back to D is to delete and re-create the file. (1)

```
:SWC DICT filename item-list {(I}
```

where: I = Inhibit error message [782] 'A' updated.

10.4 :SWD

The :SWD verb changes a DX, DY or DZ file to a D file. If the file type was not DX, DY, or DZ (or DCX, DCY, or DCZ), then no operation takes place. If the file type was DCX, DCY or DCZ, then the file type is changed to DC. (1)

:SWD DICT filename item-list {(I}

where: I = Inhibit error message [782] 'A' updated.

10.5 :SWX

The :SWX verb changes the file type to DX. If the file type contained Y or Z, it is removed first, then the new code added. (1)

:SWX DICT filename item-list {(I}

where: I = Inhibit error message [782] 'A' updated.

10.6 :SWY

The :SWY verb changes the file type to DY. If the file type contained X or Z, it is removed first, then the new code added. (1)

:SWY DICT filename item-list {(I}

where: I = Inhibit error message [782] 'A' updated.

10.7 :SWZ

The :SWZ verb changes the file type to DZ. If the file type contained X or Y, it is removed first, then the new code added. (1)

:SWZ DICT filename item-list {(I}

where: I = Inhibit error message [782] 'A' updated.

10.8 COMPARE

Compares two items from the same or different files and lists them, indicating differences. New K-option to suppress all output to screen. Should be used with the L option. (2)

```
COMPARE filename item-name {...item-name} {(options)}
WITH:    (filename item-name {...item-name})
```

Options:

- A Compare entire lines. (See G option)
- B Compare assembly source lines, including comment.
- C Begin page on each mismatched item.
- D Suppress object code address line (when used with O option).
- F Begin page on each mismatched item.
- G When used with A, do not compare lines beginning with *.
- I Suppress NOT-ON-FILE message.
- K Suppress output to terminal. (See L option)
- L Generate saved list of mismatched items.
- N Nopage on terminal.
- O Compare object code only.
- P List on printer.
- S Suppress display of identical items.
- T Compare file on disk with file on tape.
- Y Output bell character at completion of each item.
- Z Output error lines only.

10.9 RENAME-FILE

RENAME-FILE is used to move or rename file definition pointers. Only file definition items may be operated on by this verb (not synonym definitions). The source file may be renamed or moved (and renamed) to a new dictionary. The RENAME-FILE verb always deletes the source file definition after updating the destination file definition. RENAME-FILE may only move file definition items to dictionaries that are at the same level as the source dictionary. That is, a file definition item in the master dictionary may not be copied to the SYSTEM dictionary or to a file dictionary, but could be copied to another account's master dictionary. It may be necessary to use the RENAME-FILE verb twice to completely rename a file, if the user desires to keep the data section name the same as the dictionary name. (1)

```
RENAME-FILE {DICT} file-name file-definition-item {(options)}
TO: {} {DICT} file-name {file-definition-item}
```

Options:

- F Form-feed. Display each item on terminal or printer onto new page.
- I Item-ids suppressed during file rename.
- N No page wait on terminal.
- P Printer is destination.
- T Terminal is destination.
- X Hexadecimal format used on terminal or printer copy

10.10 CREATE-FILE

The CREATE-FILE verb creates new accounts, dictionaries, and data files in the system. It creates a file by reserving disk space for the file and inserting a file definition item (D-pointer) into the parent file (SYSTEM for an account, MD for a dictionary, dictionary for a data file). (1)

The general forms of the CREATE-FILE verb are:

CREATE-FILE file-name m1{,s1} m2{,s2} (options)

CREATE-FILE dict-name,data-name m1{,s1} m2{,s2} (options)

CREATE-FILE DICT dict-name m1{,s1} (options)

CREATE-FILE DATA file-name m2{,s2} (options)

CREATE-FILE DATA dict-name,data-name m2{,s2} (options)

where:

file-name	is the name of the file
dict-name	is the name of the dictionary
data-name	is the name of the data section
m1	is the modulo of the dictionary
s1	is the separation of the dictionary
m2	is the modulo of the data section
s2	is the separation of the data section

Forms which do not specify DICT or DATA create both the dictionary and data sections of the file. If DICT is specified, only the dictionary section is created. If DATA is specified, the dictionary must already exist, and only the data section is created.

Options:

C	create DC file (only DICT file pointer will be DC)
Y	create DY file (only DATA file pointer will be DY)
X	create DX file (both DICT and DATA file pointers will be DX)
R	create file in another account (SYS3 privileges)

Either X or Y may be specified, but not both.

The R option is used to specify another account as the destination for the file being created. SYS3 privileges are required when the R option is specified. If the R option is specified, you will be prompted:

Account ?

A valid account definition (in SYSTEM) must be entered. The account definition may be either a D-pointer or a Q-pointer. Note that specifying SYSTEM as the account will create the file in the SYSTEM dictionary. In this case, a dictionary level file only must be created.

10.11 DELETE-FILE

The DELETE-FILE verb deletes an account (including all dictionaries and data level files), dictionary (including all data level files) or data file. (1)

The general forms of the DELETE-FILE verb are:

```
DELETE-FILE file-name (option)
DELETE-FILE DICT dict-name (option)
DELETE-FILE DATA data-name (option)
DELETE-FILE DATA dict-name,data-name (option)
```

where:

```
file-name   is the name of the file
dict-name   is the name of the dictionary
data-name   is the name of the data section
```

The form which does not specify either DICT or DATA deletes both the dictionary and data level files.

Option:

R delete file from another account. (SYS3 privileges)

If the R option is specified, you will be prompted:

Account ?

A valid account definition (in SYSTEM) must be entered. The account definition may be either a D-pointer or a Q-pointer.

To delete an account, specify the desired account as the file-name in the DELETE-FILE command (with the R option) and specify SYSTEM at the "Account ?" prompt. This will delete the desired account including all dictionaries and data level files. Note that the DICT or DATA modifiers may not be specified when using this verb to delete an account.

10.12 MESSAGE or MSG

Sends message text to a specified account or line-number. (2)

The general forms of these verbs are:

MESSAGE acct text {(options)}

MESSAGE ln text {(options)}

or

MSG acct text {(options)}

MSG ln text {(options)}

ln - line number to send the message

!* - send message to all lines

acct - account to send the message if logged on

* - send message to all accounts logged on

text - the message to be sent

Options:

S suppress the bell at destination

F replace the character "\" contained in "text" with a blank line. (format)

Example:

Message sent from SYSPROG on line 0:

>MSG !1 LOGOFF NOW

Message received on line one:

Message from line 0, SYSPROG (14:32:30 22 FEB 1988)

LOGOFF NOW

Note that the line number originating the message is now included as part of the message.

10.13 PEEK n

n: Line number to peek at

The PEEK verb allows a user with SYS2 privileges to view input and output of one other specified process. The originating process continues to function as before the verb was invoked.

10.14 PEEK-STOP {n}

n: Line number to STOP the peeking process

The PEEK-STOP verb is used to discontinue specified process from peeking. If no line number is specified, then the current line number is used.

PEEK cannot be discontinued on a line that invoked TANDEM.

10.15 PEEK-ON {n}

n: Line number to enable PEEK

Allows another process to PEEK at the specified process. If no line is specified, then the current line number is used. Requires SYS2 privileges to enable current line and SYS3 privileges to enable another line.

10.16 PEEK-OFF {n}

n: Line number to disable PEEK.

Inhibits another process from peeking at the specified process. If no line is specified, then the current line number is used. Requires SYS2 privileges to disable current line and SYS3 privileges to disable another line.

PEEK cannot be disabled on either line that is currently involved in TANDEM.

10.17 PEEK-STAT {n}

n: Line number to get PEEK status.

Displays the current status of the specified process. If no line is specified, then the currently line number is used. Requires SYS2 privileges.

10.18 POKE

Allows the user to input data on another line, if that line is allowed to be "poked". See POKE-ON verb. One frame of overflow is attached when the verb is executed and detached when the receiving line has received the input data. Port number must be specified. (1)

```
POKE n,{text string}
      or
POKE n [{filename {item-id}}]
      or
POKE n esc
      or
POKE n
```

where:

n line number to send data

text string input data that is to be used by the target process when data is requested.

NOTE: The ESC key maybe used to delimit a string of data for multi-line input. The ESC key is displayed as "[" when entered.

i.e. >POKE 1,WHO[WHAT[WHERE<

If line one is at TCL the functions WHO, WHAT and WHERE will be executed.

filename file where POKE item resides.

NOTE: If a filename is not present then the user is prompted with:

FILENAME:

If a valid file name is entered then the user is prompted with:

ENTER ITEM-ID:

item-id If a valid item-id is entered, then each line in the item will be used as input data.

If the item COMMANDS in the file POKE.FILE appears as:

```
COMMANDS
TOP
.P
001 WHO
002 WHAT
003 WHERE
EOI 003
```

and if line 1 is at TCL, then:

```
>POKE 1 [POKE.FILE COMMANDS
```

the functions WHO, WHAT and WHERE will be executed.

esc if ESC is entered at the "FILE NAME:" prompt then

```
0001:
```

will appear. At this prompt, single commands may be entered. After each command is entered, another numbered prompt will appear below it. This will continue until a null command is entered.

```
>POKE 1[<
FILE NAME:[

0001: WHO <CR>
0002: WHAT <CR>
0003: WHERE <CR>
0004: <CR>
```

If line one is at TCL the functions WHO, WHAT and WHERE will be executed.

Further examples:

>POKE n	Send <CR><LF> to line n's input buffer
>POKE n,	Send <CR><LF> to line n's input buffer
>POKE n,text-string	Send text-string to line n's input buffer
>POKE n[Prompt for filename, then itemname
>POKE n[filename	Prompt for itemname
>POKE n[filename itemname	Send command via filename itemname

10.19 POKE-OFF

Prevents another user from "poking" input data on the current line. POKE-OFF is system default at boot up time. A user with SYS3 privileges may prevent anyone from "poking" input data to a designated line 'n', a series of lines 'n-m', or all lines '*'. (1)

POKE-OFF {n}{-m}

where: n = starting line number or * for all lines
m = ending line number

10.20 POKE-ON

Allows the current line to accept "poked" input data from another line. A user with SYS3 privileges may set the poke status to on for another line by specifying the line number, a series of lines by specifying the starting line number and ending line number, or all lines by using an asterisk.(1).

POKE-ON {n}{-m}

where: n = starting line number or * for all lines
m = ending line number

10.21 POKE-RESET

This verb requires SYS3 privileges and should be used only if a condition occurs that will not allow the POKE-ON and POKE-OFF verbs to function. Use of this verb may cause a frame of overflow to be thrown away. (3)

POKE-RESET {n}

10.22 T-DUMP

An option has been added to T-DUMP to dump a compatible tape for moving data to older systems. The "C" option will dump a compatible tape, converting items in a DC type file to list (CL) items. Linked binary items and non-DC type items longer than 32267 bytes may not be dumped to a compatible tape (a warning will be printed if this is attempted).(1)

10.23 T-LOAD

The T-LOAD verb will accept a "save" format tape as input. If a tape is determined to be a "save" format tape, one or more of the following prompts will be issued: (1)

Account name on tape: For full-system save tapes or account level save tapes, for example:

SAVE SYSTEM SYSTEM or SAVE SYSTEM account-name
FILE-SAVE or ACCOUNT-SAVE account-name

Dictionary name on tape: For dictionary level save tapes, for example:

SAVE MD file-name or SAVE SYSTEM,account-name file-name

File name on tape: For data level save tapes, for example:

SAVE DICT file-name data-name
SAVE DICT SYSTEM,account-name,file-name data-name

Normal ACCESS selection criteria may be used to specify items which are to be loaded from the tape.

T-LOAD may be used with either full or updates-only save tapes as well as account, dictionary, and data file saves. If an updates-only save tape is used, only new or updated items will be restored. Deleted items will not be deleted from the file.

When using T-LOAD with a "save" format tape, the tape must be positioned at the beginning of a tape file before executing the verb, unless the A options is specified. If data is being restored from an account which begins on a reel other than the first, then a T-FWD should be issued to position the tape at the beginning of the next tape file prior to executing the T-LOAD. If this is not done, then the tape will probably not be identified as a "save" format tape, and erroneous data may be loaded.

Note that the T-LOAD verb used with a "save" format tape honors the retrieval and update restrictions of the file on the tape. That is, you may not load from account, dictionary or data file which you are not authorized to access.

Options:

- S The S option will suppress the list of file names displayed during the search for the file to be restored.
- N The N option will search for a specified file number rather than a file name. If the N option is specified, the following prompt will be issued:

File number on tape :

- A The A option will cause T-LOAD to assume a save format tape, at a position other than BOT (beginning of tape) or EOF (end of file). Since the logical location cannot easily be determined, the following prompt will be issued:

Level: 0-System 1-Account 2-Dictionary 3-Data File :

The levels are interpreted as follows: 0 - tape is positioned somewhere within the SYSTEM file. The first file prompt will be for Account name. 1 - Tape is positioned within the desired account. The first file prompt will be for Dictionary name. 2 - tape is positioned within the desired dictionary. The first file prompt will be for File name. 3 - Tape is positioned within the desired data file. No file name prompt will be issued. The A option is useful when T-LOADing from volumes other than the first volume in a multi-volume set, or when performing multiple T-LOADs from a single account.

10.24 VERIFY-SAVE

Verifies the currently attached tape as if it were being restored to disk, but discards the data after reading it. This operation will detect defective D-pointers, invalid item sizes, unrecognizable tape segments, and other tape format errors. Verb format is:

VERIFY-SAVE {account-name (A)}

Option: A Will prompt Account name on tape:
User may enter a return or account-name again.

10.25 VERIFY-SAVE-DATA

Verifies that the data on the tape is the same as the data on the system disk(s). The files on the disk are opened as they are encountered on the tape, and as each item is read from the tape, it is compared to the item on the system. Note any items that change on disk since the save will be reported as an error. Therefore, for the verification to be useful, the system should not be in use while the verification is in progress. Verb format is:

VERIFY-SAVE-DATA {account-name (A)}

Option: A Will prompt Account name on tape:

User may enter a return if account-name on disk is
the same as account-name on tape.

10.26 LIST-LOCKS

Both group and item locks are displayed with the decimal and hex representation of the group FID. Both group and item locks are terminated with the line number which executed the group or item lock. The item lock also displays the hex representation of the item that it is locking. (0)

Example: LIST-LOCKS

15840 (03DE0)-030 <== Group lock
148652 (244AC)-600D-029 <== Item lock

```
### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ###
### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ###
### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ###

### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ###
### ### ### ### ### ### ### ### ### ### ### ### ###
```

10.27 WHAT

The WHAT verb displays system statistics and process status.(2)

WHAT {options}

Options: L=Suppress lock status display
 P=Print system status to printer
 S=Suppress SP-STATUS display
 W=Suppress system process information.

In the example, the numbers in brackets identify Notes listed below; they are not a part of display.

15:59:29 18 DEC 1988

Memory	Lines	Pcb0	Wsstart	Wssize	Sysbase/Mod/Sep	Maxfid	Overflow	Data.size
7177K	131	1024	5216	127	55127 11 1	512767	418035	500
[1]	[2]	[3]	[4]	[5][6].....	[7]	[8]	[9]

68063 (109DF)-001 56110 (ODB2E)-126 [10]

56110 (ODB2E)-69D2-125 [11]

[12]
 ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ###
 ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ### ###

001 ### ### ### ### ### [13]
 ### ### ### ### ### ### ### ### ### ### ###

*000 0400 7F00 121.000 121.1DG 166.64A
 001 0420 7F00 TC 852.644 854.6E6
 006 04C0 3F00 170.064 170.0B6
 125 13A0 7F00 221.0B0 221.54C 221.0EE 232.2CA
 126 13C0 7F00 388.716
 127 13E0 7F00 119.2EC 232.434
 128 1400 3F00 326.0A2
 129 1420 3F00 323.0BA
 130 1440 3F00 170.064 170.154
 [14] [15] [16-17] [18] [19]

The spooler is inactive.

Printer # 0 is serial, inactive, and on line.
 The printer is running on line 6.
 Assigned output queues: 0.
 The number of inter-job pages to eject is 0.

NOTES for "WHAT":

- [1] Available memory for processing
- [2] Number of communication lines (terminals), LAN processes, and Spooler = number of processes on system.
- [3] PCB-FID for port zero; each following port PCB-FID is displaced by 32 frames from PCB0.
- [4] Extended workspace starting FID: $Wsstart = Pcb0 + 32 * \text{No. of processes (including spooler, ports, LAN)}$.
- [5] Extended workspace size; number of frames per workspace is 127 frames. There are three workspaces per line.
- [6] System base-FID/modulo/separation;
 $Sysbase = Wsstart + Wssize * 3 * \text{No. of processes}$.
- [7] Maxfid -- last frame on disk.
- [8] Overflow -- No. of linked frames + contiguous frames.
- [9] Data.size = 500 or 2000. If frame size = 512 bytes, first 12 bytes are links and the next 500 bytes are for data. If frame size = 204k bytes, first 48 bytes are links and the next 2000 bytes are for data.
- [10] Group-locks (if any). The format is ddddd (xxxxx)-cc

Where: ddddd = group FID (decimal)
 xxxxx = group FID (hex)
 cc = port number

Port number is internally coded from PCB-FID of port (upper nybble of lower byte concatenated with lower nybble of upper byte). It displays as the actual port number.

- [11] Item-locks (if any). The format is ddddd (xxxxx)-yyyy-cc

Where: ddddd = group FID (decimal)
 xxxxx = group FID (hex)
 yyyy = hex representation of item that is locked
 cc = port number or LAN

- [12] BASIC locks (48). Bits start at 127.20.

- [13] System lock bytes; ###=available; else has port number as above.

LOCK #	LOC	USAGE
0	127.0	Lock-table lock
1	127.1	Overflow table lock
2	127.2	Group-lock table lock
3	127.3	Linking Workspace lock
4	127.4	Binary save lock.
5	127.5	Message lock
6	127.6	Item-lock lock
7	127.7	Reserved
8	127.8	File restore lock (:FILELOAD)
9	127.9	Full file-save lock
10-25		Reserved

- [14] Port number; preceded by a "*" if your port.
 [15] PCB FID (hex) of port.
 [16] PIB status bits:
- | | | |
|---|------------|--------------|
| 7 - Unused, normally 0 | | |
| 6 - Sleep bit | 0 = Active | 1 = Normal |
| 5 - Disk bit | 0 = Active | 1 = Normal |
| 4 - Auto Echo | 1 = Active | 0 = Inactive |
| 3 - UART | 0 = Active | 1 = Normal |
| 0 maybe output buffer full during write operation or could be X/OFF | | |
| 2 - Write Roadblock | 0 = Active | 1 = Normal |
| 1 - Read Roadblock | 0 = Active | 1 = Normal |
| 0 - Coma bit | 0 = Active | 1 = Normal |
- COM.SYS usually ^C
- PIB-status of port messages:
- 7F/FF = Active, or ready to go
 - 7B/FB = Terminal output
 - 7D = Terminal input
 - 5F = Waiting for disk
 - 3F = Release Quantum/Sleeping
- Typically, LAN, spooler and printers is "3F"
- [17] PIB status-2: 00=Normal, 80=in debugger, 0A=debugger
 [18] T=Tape attached; C=Cache-on
 [19] Location counter (first address) and subroutine return = stack addresses.
 Entry format = fff.11 where fff=decimal FID; 111=hex location.

Typical locations;

5=TCL1	6/9=Terminal I/O	13-20=Editor
53-64=Access Compiler	71-77=List	161-185=Spooler
188-199=Basic Compiler	225-249=Basic Runtime	290-298=Runoff
850-856=Save	861-870=Restore	

10.28 CLEAR-GROUP-LOCKS

Clears Group Locks only. Group Locks are set by the system whenever a group(is actually being accessed and last for the duration of the access, either for reading or updating. Should be used with extreme caution. May cause GFE's if users are not logged off. (S)

10.29 CLEAR-ITEM-LOCKS

Clears Item Locks only. Item Locks are set by the Basic process for exclusive use of an item. They are invoked by the user via the READU, READVU, MATREADU or MATREADVU statements. They last until released by the invoking process. Should be used with extreme caution. May cause GFE's if users are not logged off. (S)

10.30 CLEAR-LOCKS

Clears all locks on the system including Item-Locks, Group-Locks, Basic Execution Locks (set by the Basic LOCK(N) statement) and System Level Locks. System Level Locks are set by the system to keep competing processes from destroying shared resources (such as the overflow table). Use very extreme caution; may cause GFE's, system aborts or catastrophic system problems if users are not logged off. (S)

10.31 LIST-GFES

This proc lists the GFES logged in the SYSTEM-ERRORS file. See GFE.RECOVER verb to recover certain types of logged items. (O)

10.32 GFE.RECOVER

Recovers GFES stored in the SYSTEM-ERRORS file due to an invalid hash address, duplicate item-id, item-id longer than 50 characters and item-ids that contain a system delimiter character. The Basic program creates a "D" pointer to a modulo 1 file using as a base address the FID reported in the error item. The system editor is entered to examine the item, and the item may then be copied to any file changing the ID if desired. (S)

10.33 ADD.UPDATE.MD

Compares selected account-name(s) Master Dictionary verbs against NEWAC verbs with same name and changes Master Dictionary verbs to agree with NEWAC. Updates old verb definitions and adds any additional NEWAC verbs to Master Dictionary. Deletes obsolete verbs from Master Dictionary if the user answers "Yes" to the query: "Do you wish obsolete verbs to be deleted (<Y>/N)?". (S)

ADD.UPDATE.MD {account-name}

If an account name is not specified, the program will prompt for account to update. Entering an asterisk '*' character as the optional account-name or the prompted account-name will cause the program to generate and display a list of the accounts currently on the system. It will place an asterisk next to the accounts that it will update. The system accounts that should not be updated are automatically excluded from the update list. The user has the ability to override the default list for any account by adding or deleting it from the list. If the account display requires more than one page, prompts are included for forward and backward paging. Accounts may only be added or deleted from the current page. Once the list of accounts is satisfactory, the update process may be started by entering an 'S' for start. Entry of a 'Q' will quit the program back to TCL, but it must occur during list selection. Once the 'S' to start the update is input, the 'Q' option is no longer available.

An example of the screen display during list selection is as follows:

* 1. MISC	2. ACC	* 3. ALPHA.TEST
* 4. ATP	* 5. BENCH	6. BLOCK-CONVERT
* 7. COM.SYS	* 8. CPL	* 9. CPL1
* 10. CS.PLUS	11. DAILY.MSG	12. ERRMSG
13. EXECUTE-CONTROL	* 14. GA	* 15. HR
* 16. LBK	* 17. PLASTICOLORS	18. PROCLIB
19. PSYM	* 20. SYS-TEST	* 21. SYSPROG
22. SYSTEM	23. SYSTEM-ERRORS	* 24. TEST
* 25. TESTACC.1	* 26. TESTACC.10	* 27. TESTACC.100
* 28. TESTACC.11	* 29. TESTACC.12	* 30. TESTACC.13
* 31. TESTACC.14	* 32. TESTACC.15	* 33. TESTACC.16
* 34. TESTACC.17	* 35. TESTACC.18	* 36. TESTACC.19
* 37. TESTACC.2	* 38. TESTACC.20	* 39. TESTACC.21
* 40. TESTACC.22	* 41. TESTACC.23	* 42. TESTACC.24
* 43. TESTACC.25	* 44. TESTACC.26	* 45. TESTACC.27
* 46. TESTACC.28	* 47. TESTACC.29	* 48. TESTACC.3
* 49. TESTACC.30	* 50. TESTACC.31	* 51. TESTACC.32
* 52. TESTACC.33	* 53. TESTACC.34	* 54. TESTACC.35

ENTER 1-54, (Q)uit, (S)tart update, (N)ext page:

An '*' indicates account will be updated.

10.34 DELETE.MD

Compares MD verbs against verbs in the DEL.NEWAC file and deletes them from the account's Master Dictionary. The DEL.NEWAC file, as a rule, does not exist and must be created. Verb definitions to be deleted must be added to the file.(S)

DELETE.MD {account-name}

If an account name is not specified, the program will prompt for account to update. Entering an asterisk '*' character as the optional account-name or the prompted account-name will cause the program to generate and display a list of the accounts currently on the system. It will place an asterisk next to the accounts that it will update. The system accounts that should not be updated are automatically excluded from the update list. The user has the ability to override the default list for any account by adding or deleting it from the list. If the account display requires more than one page, prompts are included for forward and backward paging. Accounts may only be added or deleted from the current page. Once the list of accounts is satisfactory, the update process may be started by entering an 'S' for start. Entry of a 'Q' will quit the program back to TCL, but it must occur during list selection. Once the 'S' to start the update is input, the 'Q' option is no longer available. See ADD.UPDATE.MD for an example of screen display of the list selection.

10.35 UPDATE.MD

Compares Master Dictionary verbs against NEWAC verb with same name and changes Master Dictionary verbs to agree with NEWAC. Does not add any additional NEWAC verbs to Master Dictionary. Deletes obsolete verbs from Master Dictionary if the user answers "Yes" to the query: "Do you wish obsolete verbs to be deleted (<Y>/N)?".(S)

UPDATE.MD {account-name}

If an account name is not specified, the program will prompt for account to update. Entering an asterisk '*' character as the optional account-name or the prompted account-name will cause the program to generate and display a list of the accounts currently on the system. It will place an asterisk next to the accounts that it will update. The system accounts that should not be updated are automatically excluded from the update list. The user has the ability to override the default list for any account by adding or deleting it from the list. If the account display requires more than one page, prompts are included for forward and backward paging. Accounts may only be added or deleted from the current page. Once the list of accounts is satisfactory, the update process may be started by entering an 'S' for start. Entry of a 'Q' will quit the program back to TCL, but it must occur during list selection. Once the 'S' to start the update is input, the 'Q' option is no longer available. See ADD.UPDATE.MD for an example of screen display of the list selection.

10.36 CONVERT.MD

Converts the master dictionary of 3.8 accounts to 3.4. If the account has SYS3 privileges, the privilege level is changed to SYS2. All 3.8 SYS3 and SYS2 verbs are deleted and 3.4 verbs are copied to the master dictionary of the account. When the account is saved to tape for transfer to a 3.4 system, the user must remember to use the C-option of the SAVE verb to make the tape backward compatible.(S)

CONVERT.MD {account-name}

If an account name is not specified, the program will prompt for account to update. Entering an asterisk '*' character as the optional account-name or the prompted account-name will cause the program to generate and display a list of the accounts currently on the system. It will place an asterisk next to the accounts that it will update. The system accounts that should not be updated are automatically excluded from the update list. The user has the ability to override the default list for any account by adding or deleting it from the list. If the account display requires more than one page, prompts are included for forward and backward paging. Accounts may only be added or deleted from the current page. Once the list of accounts is satisfactory, the update process may be started by entering an 'S' for start. Entry of a 'Q' will quit the program back to TCL, but it must occur during list selection. Once the 'S' to start the update is input, the 'Q' option is no longer available. See ADD.UPDATE.MD for an example of screen display of the list selection.

Only two messages appear:

Converting M/DICTS to 3.4

** account-name ** converted to 3.4

10.37 CHECK-SUM, STAT, SUM

These Access verbs do not use the pre-processor, therefore, explicit item-ids should be enclosed in quotes.

10.38 DEFAULT-MODSEP

This SYSPROG only verb allows the user to display or set the default modulo and separation for STAT-FILE, SAVELOG and RESTORE-ERRORS.

DEFAULT-MODSEP {filename mod{,sep}} {(P)}

where: filename = STAT-FILE, SAVELOG or RESTORE-ERRORS
 mod = modulo
 sep = separation
 (P = output to printer

```
>DEFAULT-MODSEP
STAT-FILE 53,1
SAVELOG 1,1
RESTORE-ERRORS 1,1
```

10.39 DATE-FLAG

Sets internal date format to either U.S. (mm/dd/yy) or international (dd/mm/yy). Date flag will not change until ABS is reloaded or verb is executed again. U.S. is the default.(3).

DATE-FLAG {option}
 I = International format
 U = U.S. format

Option may be enclosed in parentheses if desired.

10.40 TANDEM n

n: Line number to tandem to.

The TANDEM verb allows a user with SYS2 privileges to view input and output (and, optionally, control input) of one other specified process. Three levels are provided to the originator.

The 'VIEW' level allows the originator to only view the input and output generated by the target process. This level is provided when the verb is invoked. To (re)enter this level an <ESC> followed by a 'V' is entered. The word 'view' is displayed on the originators terminal to indicate the level is entered.

The 'MESSAGE' level allows the originator to view the input and output generated by the target and have the target process display the input generated by the originator. To (re)enter this level and <ESC> followed by an 'M' is entered. The word 'message' is displayed on the originators terminal to indicate the level is entered.

The 'FEED' level allows the originator to view the input and output generated by the target and have the target process use the input generated by the originator. To (re)enter this level and <Esc> followed by an 'F' is entered. The word 'feed' is displayed on the originators terminal to indicate the level is entered.

While in the feed level an <Esc> can be sent to the target by entering an <Esc> followed by an <Esc>.

While in the feed level a BREAK can be sent to the target by entering an <Esc> followed by a 'D'.

To determine the current TANDEM level, enter <Esc> followed by an 'S'. The current level will be displayed on the originating process.

Exiting TANDEM is done by entering an <Esc> followed by an 'X'. The originating process will be sent back to TCL.

10.41 TANDEM-ON {n}

n: Line number to enable TANDEM.

Allows another process to TANDEM to the specified process. If no line is specified, then the current line number is used. REquires SYS2 privileges to enable current line and SYS3 privileges to enable another line.

10.42 TANDEM-OFF {n}

n: Line number to disable TANDEM.

Inhibits another process from tandeming to the specified process. If no line is specified, then the current line number is used. Requires SYS2 privileges to disable current line and SYS3 privileges to disable another line.

TANDEM cannot be disabled on either line that is currently involved in tandeming.

10.43 TANDEM-STAT {n}

n: Line number to get TANDEM status.

Displays the current status of the specified process. If no line is specified, then the current line number is used. Requires SYS2 privileges.

10.44 DUMP

The DUMP verb may be used to display data in a frame or to display absolute core locations. The data display may be specified in either character or hexadecimal format. The general form of the DUMP verb is as follows:

DUMP n1{-n2},{options}

"n1" and "n2" are numbers that may be specified in decimal or in hexadecimal by preceding the hex number with a period (.). The n1 and n2 parameters contain the beginning (and ending) FID(s) of the frame(s) being dumped. After the first entry, if n1 is not entered, the next frame will be displayed. If n1 is the ABS area, the ABS frame will be displayed in 2048 byte format without links. Otherwise, the data frame will be displayed in 500 or 2000 byte format with links. The left-hand column gives the absolute decimal or hexadecimal displacement, unless the "B" option is used. Options are specified like normal statement options, as single characters, optionally separated by commas. Valid options are:

<u>Option</u>	<u>Description</u>
A	Converts EBCDIC to ASCII.
B	Display data frame (500 or 2000 bytes) starting with byte "1" of the data frame.
C	Core dump; specifies that absolute memory locations are to be dumped. A 512 or 2048-byte block starting at n1 is dumped. The whole frame is displayed as data. Forward or backward links are not displayed separately at the top.
G	Group; specifies that the data starting at frame n1 is to be dumped and that the dump continue following either the forward or backward links (depending on whether the U option is not or is specified). The dump will terminate when the last frame in the logical chain has been found.
L	Links; specifies that the dump be confined to the "links" of the frame(s) concerned; no data is displayed.
N	No page; if the data is printed on the terminal, specifies that the end-of-page stop be inhibited.
P	Printer; the display is routed to the line-printer.
U	The data or links are traced logically upwards; that is, the backward links are used to continue the display.
X	Outputs in hexadecimal and character format.

list of verbs by system privilege A

SYSO PRIVILEGE

NEWAC..... *A1..... *A2..... *A3..... *A4..... *A5.....

ADDD	PA	40A0			
ADDX	PA	A0			
BASIC	P	2	BE		UP
BFORMAT	P	2	119D		UP
BLIST	P	2	19D		
BLOCK-PRINT	P	500A	41		
CATALOG	P	2	E0		N
CHARGE-TO	P	6032			
CHARGES	P	5032			
CHECK-SUM	PX	35	4F		
COMPILE	P	2	BE		UP
CONTROL-CHARS	P	909E			
COPY	PZ	2	8C		UZ
COPY-LIST	PZ	5002	8C		SUZ
COUNT	PB	198	1069		
DAILY-MESSAGE	P	118C			
DATE	P	618C			
DELETE-LIST	P	5002	3064		SU
DIVD	PD	40A0			
DIVX	PD	A0			
DTR	P	10A0			
DTX	P	10A0			
ECHO	P	5080			
ED	PE	2	6014		CUPN
EDIT	PE	2	6014		CUPN
EDIT-LIST	PE	5002	6014		CPUNS
GET	PZ	2	142		FU
GET-LIST	P	5002	1064		ZS
GROUP	P	2	50A0		F
HASH-TEST	PA	198	106A		
ISTAT	PA	198	6A		
ITEM	P	2	30A0		N
JED	P	2	2154		CUN
JET-EDIT	P	2	2154		CUN
JET-IN	P	2	154		CUN
JET-OUT	P	2	1154		CS

(SYS0 Privilege Continued)

JIN	P	2	154		CUN
JOUT	P	2	1154		CS
LIST	PA	198	4D		
LIST-ITEM	PA	198	4D	508E	
LIST-LABEL	PA	198	4D	7D	
LIST-LOCKS	P	500A	30CD		
LISTABS	P	20AE			
LISTPEQS	P	AB			
LISTPTR	P	AE			
LOAD	PZ	2	142		FUN
LOGTO	PG	7032			
MULD	PM	40A0			
MULX	PM	A0			
NODE	P	147			
OFF	PZ	32			
P	P	2080			
PIE	PA	198	4D	11CC	
PLOT	PA	198	4D	01CC	
POVF	P	40BB			
PRIME	P	E6			PROCLIB
PRINT-ERR	P	2	43		
PROTOCOL	P	114			
PVERIFY	PZ	2	E0		N
QSELECT	P	2	303A		Z
RECOVER-FD	P	1012			
REFORMAT	PA	198	4D	12C	
RENUMBER	P	2	119D	1000	UP
RESET-TERM	PZ	818C			
RTD	P	20AD			
RUN	PR	2	E6		F
RUNI	PR	2	30E6		F
RUNOFF	P	2	126		S
SAVE-LIST	P	5002	2064		F
SEARCH	P	2	6B		CUZ
SELECT	PB	198	3076		
SEND	PB	198	140		
SET-BAUD	P	10E			
SET-SYM	P	2	89		F
SLEEP	P	107A			
SORT	PA	198	4E		
SORT-ITEM	PA	198	4E	508E	
SORT-LABEL	PA	198	4E	7D	
SORT-LIST	P	5002	72		SCU
SP-ASSIGN	P	10A7			
SP-CLOSE	P	A7			
SP-EDIT	P	A9			

(SYS0 Privilege Continued)

SP-KILL	P	10AD		
SP-OPEN	P	20A7		
SP-STATUS	P	50A6		
SPIE	PA	198	4E	11CC
SPLOT	PA	198	4E	01CC
SREFORMAT	PA	198	4E	12C
SSELECT	PB	198	4E	3076
STACK-OFF	P	2188		
STACK-ON	P	1188		
STAT	PF	35	4F	
SUBD	PS	40A0		
SUBX	PS	A0		
SUM	PW	35	4F	
TA-OFF	P	210E		
TA-ON	P	110E		
TABS	P	80		
TERM	PZ	607A		
TIME	PZ	3033		
TYPEAHEAD-OFF	P	210E		
TYPEAHEAD-ON	P	110E		
WHICH	P	409E		
WHO	P	10BB		
XTD	P	20A0		

SYS1 PRIVILEGE

NEWAC..... *A1..... *A2..... *A3..... *A4..... *A5.....

NEWAC.....	*A1.....	*A2.....	*A3.....	*A4.....	*A5.....
:LANOFF	P	150			
:SWC	PA	198	1090	43	UP
:SWD	PA	198	1090	44	UP
:SWE	PA	198	1090	45	UP
:SWX	PA	198	1090	58	UP
:SWY	PA	198	1090	59	UP
:SWZ	PA	198	1090	5A	UP
ACCOUNT-RESTORE	P	635D			
ACCOUNT-SAVE	P	1353			
B/ADD	PZ	2	53	57	CS
B/DEL	PD	2	53	57	CS
CACHE-OFF	P	1139			
CACHE-ON	P	139			
CLEAR-CASCADE	P	606D			
CLEAR-FILE	PO	8F			
CREATE-FILE	PO	8D			
CREATE-PFILE	PO	408D			
DECATALOG	PY	2	E0		NU
DELETE	PD	2	509E		CUZ
DELETE-FILE	PO	108F			
LANOFF	P	150			
LANSTOP	P	150			
PASSWORD	P	186			
POKE	PG	18B			
POKE-OFF	PG	318B			
POKE-ON	PG	218B			
PRINTRONIX	P	40B4			
RENAME-FILE	PF	2	308C		UZ
RESTORE	P	2	35D		FU
S-DUMP	PF	198	4E	25	
SAVE	P	2	353		
SEL-RESTORE	P	90D7			
SEND-MSG	PG	118B			
SET-1/2	P	20A8			
SET-1/4	P	DOA8			
SET-CASCADE	P	506D			
SET-CT	P	DOA8			
SET-MT	P	20A8			
SP-TAPEOUT	P	20B4			

(SYS1 Privilege Continued)

T-ASSIGN	P	6033	9A	
T-ATT	P	40A8		
T-BCK	PI	6033	A023	
T-CHK	P	B09A		
T-COPY	P	00C9		
T-DET	P	60A8		
T-DUMP	PB	198	25	
T-EOD	P	6033	709A	
T-EOF	PN	6033	3023	
T-EOFD	PO	6033	C09A	
T-ERASE	PN	6033	A09B	
T-FORMAT	P	6033	0151	
T-FWD	PI	6033	9023	
T-LOAD	PT	198	73	
T-RDLBL	PO	6033	2024	A09A
T-READ	P	99		
T-RET	PN	6033	B09B	
T-REW	P	306D		
T-STATUS	P	E0A8		
T-UNLOAD	P	R06D		
T-WEOF	PN	6033	1024	
T-WTLBL	PO	6033	2099	A09A
VERIFY-SAVE	P	935D		
VERIFY-SAVE-DATA	P	A35D		
XOFF-DISABLE	P	918C		
XOFF-ENABLE	P	A18C		

SYS2 PRIVILEGE

NEWAC..... *A1..... *A2..... *A3..... *A4..... *A5.....

:BTI	P	2	5352		U
:ITB	P	2	4352		U
:LAN-RESET	P	500A	5143		
:STARTSPOOLER	P	500A	60B6	10AF	
BREAK-KEY-DISABLE	P	418C			
BREAK-KEY-ENABLE	P	518C			
BREAK-KEY-OFF	P	418C			
BREAK-KEY-ON	P	518C			
COMPARE	P	2	17D		
DUMP	PZ	42			
EL	PE	5002	6014		CPUNS
LAN-RESTART	P	1150			
LANON	P	1150			
LANSTART	P	1150			
LINK-WS	P	40AC			
LOCK-FRAME	P	3041			
LOGOFF	P	1116			
LOGON	P	116			
MESSAGE	PG	1034			
MLIST	PY	2	20		C
MLOAD	P	2	1F		
MSG	PG	1034			
MVERIFY	P	2	101F		
PEEK	PG	01A4			
PEEK-OFF	PG	41A4			
PEEK-ON	PG	31A4			
PEEK-STAT	PG	21A4			
PEEK-STOP	PG	11A4			
POKE-STAT	PG	618B			
SET-DATE	PZ	2033			
SET-FADDR	P	10CC			
SET-FD	P	70A8			
SET-PARTIALS	P	33			
SET-TERM	PL	607A			
SET-TIME	PZ	1033			
SL	P	5002	2064		FS
STARTPTR	P	B2			
STOPPTR	P	AD			
TANDEM	PG	01A7			
TANDEM-OFF	PG	21A8			
TANDEM-ON	PG	11A8			
TANDEM-STAT	PG	01A8			
UNLOCK-FRAME	P	4041			
VERIFY-FD	P	20CC			
VERIFY-SYSTEM	P	11F			
WHAT	P	30BB			
WHERE	P	1079			

SYS3 PRIVILEGE

NEWAC..... *A1..... *A2..... *A3..... *A4..... *A5.....

:FILELOAD	P	135D	
:FILES	P	135D	
:SET-XOVR	P	800B	
CHK.DX	P	E6	PROCLIB
COM-CLEAR	P	11C0	
DATE-FLAG	PG	305B	
DEFAULT-MODSEP	P	359	
INIT-CURSOR	P	2	118F
POKE-RESET	PG	418B	
RESET-CURSOR	P	500A	318F
RESET-PORT	P	B18C	
SET-SAVE-DEFAULTS	P	0359	

list of procs by system privilege **B**

SYS0 PRIVILEGES

NEWAC..... *A1..... *A2..... *A3.....

B/LIST	PQ	(PROCLIB
CP	PQ	(PROCLIB
CS-TERM	PQ	(CS.CONTROL
CT	PQ	(PROCLIB
EXCHANGE	PQ	(PROCLIB
FORMAT	PQ	(PROCLIB
LIST-DEVICES	PQ	(PROCLIB
LIST-GFES	PQ	(SYSPROG-PL
LISTACC	PQ	(PROCLIB
LISTC	PQ	(PROCLIB
LISTCONN	PQ	(PROCLIB
LISTDICT	PQ	(PROCLIB
LISTFILES	PQ	(PROCLIB
LISTPROCS	PQ	(PROCLIB
LISTQFILES	PQ	(PROCLIB
LISTSYNS	PQ	(PROCLIB
LISTVERBS	PQ	(PROCLIB
LOOP-ON	PQ	(PROCLIB
SET-LPTR	PQ	(PROCLIB
SORTC	PQ	(PROCLIB

SYS1 PRIVILEGES

DELETE-IT	PQ	(PROCLIB
LISTU	PQ	(PROCLIB LISTUSERS
LISTUSERS	PQ	(PROCLIB
SET-FILE	PQ	(PROCLIB
T-SPACE	PQ	(PROCLIB

SYS2 PRIVILEGE

LO	PQ	(PROCLIB
----	----	----------

SYS3 PRIVILEGES

LIST.ALL.FILES	PQ	HRUN PROCLIB LIST.ALL.FILES P
----------------	----	-------------------------------

list of sysprog only C procs and verbs

Name..... Code Proc/link.....

:SHUTDOWN	PQ	(SYSPROG-PL
ADD.UPDATE.MD	PQ	(SYSPROG-PL ADD-UPDATE-MD
COLDSTART	PQ	(SYSPROG-PL COLD-START
CONVERT.MD	PQ	(SYSPROG-PL CONVERT-MD
CREATE-ACCOUNT	PQ	(SYSPROG-PL
DELETE.MD	PQ	(SYSPROG-PL DELETE-MD
FILE-SAVE	PQ	(SYSPROG-PL
FILE-SAVE-STATS	PQ	(SYSPROG-PL
FIND	PQ	** Long proc in M/DICT **
LIST-ERRS	PQ	(SYSPROG-PL
LIST-FILE-STATS	PQ	(SYSPROG-PL
LISTACCT	PQ	(SYSPROG-PL
LISTERRS	PQ	(SYSPROG-PL LIST-ERRS
STAT-RPT	PQ	(PROGLIB
SYS.LOAD	PQ	** Long proc in M/DICT **
TEST-CURSOR	PQ	** Long proc in M/DICT **
UPDATE-SAVE	PQ	(SYSPROG-PL
UPDATE.MD	PQ	(SYSPROG-PL UPDATE-MD

SYSPROG PRIVILEGES

NEWAC, SYSPROG.... *A1..... *A2..... *A3..... *A4..... *A5.....

:TASKINIT	P	13C	
ADD-ACCOUNTS	P	E6	SYSPROG-PL
APPLY-PTFS	P	E6	SYSPROG-PL
BUILD.SUBLIST	P	E6	SYSPROG-PL
CLEAR-GROUP-LOCKS	P	500A	3030
CLEAR-ITEM-LOCKS	P	500A	DD
CLEAR-LOCKS	P	500A	40DC
CREATE-PTF	P	E6	SYSPROG-PL
CREATE1	P	E6	SYSPROG-PL
CREATE2	P	E6	SYSPROG-PL
DEFAULT-MODSEP	P	359	
DEFINE-CURSOR	P	E6	SYSPROG-PL
DEL-ACCL	P	E6	SYSPROG-PL
DELETE-ACCOUNT	P	DE	
GFE.RECOVER	P	E6	SYSPROG-PL
MODSEP	P	E6	SYSPROG-PL
UPDATE	P	E6	SYSPROG-PL

list of 3.8 error message file

D

Error messages without brackets do not display the error message number when output.

- [2] Uneven number of delimiters (' " \).
- [3] Verb?
- [4] 'A' is not an account name.
- [5] The word "A" is illegal.
- [6] File name 'A' 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 D-pointer missing.
- [10] File name missing!
- [11] Frame locked at location X'A'
- [12] A power failures have occurred since last COLDSTART.
- [13] Data level descriptor (file-name in dictionary) is missing.
- [14] Tape input illegal.
- [15] The file-name is preceded by an illegal connective.
- [16] Current software release is:
 - ABS: V(1)
 - Monitor: V(2)
- [17] "WITHIN" is valid only in COUNT/LIST/SUM or STAT statements.
- [18] The last word may not be a connective.
- [19] A value without an attribute name is illegal.
- [20] Error in the 'USING' syntax.
- [21] Meaningless item-id in statement.
- [22] "TO" before item-id valid only in a CHANGE statement.
- [23] Account 'A' is save protected.
- [24] The word "A" cannot be identified.
- [25] The connective "WITH" may not immediately precede a value.
- [26] Attribute values may not both precede and follow an attribute name.
- [27] Cascading order for your line is: A,B,C,D
- [28] No configuration chip.
- [29] At least one item-id must be specified for a "WITHIN"-type statement.
- [30] Format error in MD entry defining verb.
- [31] is available.
- [32] Mag tape unit A
- [33] Cartridge tape unit A
- [34] Cartridge disk unit A

- [35] This verb requires that 3 parameters be entered consecutively.
- 36 Left bracket character : A
- Up arrow character : B
- Right bracket character: C
- 37 'A' is an illegal parameter character.
- 38 is attached to line A.
- [39] You must be on SYSPROG to change this password.
- [40] Passwords can only be deleted on SYSPROG.
- [41] Strange account pointer. Password cannot be changed.
- [42] - Illegal multiple connectives exist -
- [43] Error - receive or transmit function not specified.
- [44] Error - format specification incorrect.
- [45] You must be on the SYSPROG account for this function !!!
- [46] Error - incorrect device type.
- [47] You must be on line 0 for this function !!!
- [48] Id 'A' entered in table.
- [49] Receive entry 'A' completed.
- [50] Abort on transmit entry 'A' - function terminated.
- [51] Transmit entry 'A' completed.
- 52 is attached to line A (Primary unit).
- [53] Set in MD7/8/18: Item-id list required for WITHIN.
- [54] Wait time for receive entry 'A' matured - function terminated.
- [55] Local line-printer set operable.
- [56] Floppy disk unit A
- [57] ID 'A' deleted from table.
- [58] ID 'A' is not in table.
- [59] Account 'A' cannot be deleted because duplicate file definition items (D-Pointers) were encountered. Suggest that you change the file type to DX using the :SWX verb and when a File-Save and Restore are performed, the account will be deleted.
- [60] Account 'A' cannot be deleted.
- [61] 'A' is not an account.
- [62] A 'Q' type account cannot be deleted.
- [63] Accounts can only be deleted on SYSPROG.
- [64] There is at least one user logged on account 'A'.
- [67] International date format set
- [68] (I) or (U) options must be specified
- [69] U.S. date format set.
- [71] An illegal connective modifies the word 'A'
- [72] The value "A" is meaningless.
- [74] Abort on receive entry 'A' - function terminated.
- [75] REALITY-2780PLUS
data set not ready.
- [76] REALITY-2780PLUS
- [77] Current crossover size is A bytes.
- [78] Crossover size set to A bytes.
- [79] The number of SEPARATE and CLAUSE sets cannot exceed 9.
- [80] A system error has occurred in mode: A
This may be due to sort-key(s) preceding selection criteria.

- [81] Security code violation.
- [82] Your system privilege level is not sufficient for this statement.
- [83] Fatal tape read or write abort!
- [84] Verb is only valid for mag tape.
- [85] At beginning of cartridge tape buffer.
- [86] Unable to set primary tape unit for cascade.
- [87] Tape transfer not completed. Error code:A
- [88] Verb is not a valid cartridge tape verb.
- [89] Verb is only valid for cartridge (1/4") tape.
- [90] Tape successfully attached & block-size setup.
- [91] End tape check - A file(s)
- [92] End of recorded data - A file(s)
- [93] Attach the tape unit.
- [94] End of file.
- [95] Tape attached to line 'A'.
- [96] BOT
- [97] End of tape.
- [98] Parity error!
- [99] Block transfer not completed!
- [100] 'A ' is not.B
- [101] ASYNC device attached.
- [102] Media removed during read/write operation!
- [103] Selected device not available.
- [104] End of data.
- [105] Archive Cartridge Tape assigned for 450 ft. tapes.
- [106] Archive Cartridge Tape assigned for 600 ft. tapes.
- [107] DEI Cartridge Tape assigned for 600 ft. tapes.
- [108] Monitor and ABS dump completed.
- [109] Tape Error.
- [111] Item 'A' is not on file.
- [112] No entries in trace buffer.
- [113] Format error in trace buffer.
- [114] Illegal Cartridge Tape mode.
- [115] NORMAL Cartridge Tape mode set.
- [116] OLD Cartridge Tape mode set.
- [117] Illegal Format on Cartridge Tape.
- [118] The form 'WITH attribute AND attribute' is undefined
- [120] 'A' negative balance not permitted.
- [121] Line # must be specified!
- [122] Cannot POKE your own line!
- [123] Process is currently busy.
- [124] Spooler line cannot be used by this process.
- [125] More data required for POKE.
- [126] No over-flow available for POKE.
- [127] The specified line can not be POKE'd now.
- [128] SMA cartridge Tape mode set.
- [130] Specified line(s) can receive a POKE.
- [131] Specified line(s) are inhibited from receiving a POKE.
- [132] Specified line has been cleared.
- [136] 'A' does not match the G-CORRELATIVE specs.

- [137] Another process is currently PEEKing at specified process.
- [138] PEEKing at specified process.
- [139] PEEKing stopped for specified process.
- [140] PEEKing inhibited by others for specified process.
- [141] PEEKing allowed by others for specified process.
- [142] Specified process is PEEK inhibited.
- [143] Not allowed to PEEK at your own process.
- [144] The utility is currently being used. Try again.
- [145] Process A is peeking at process B.
- [146] Process A is not peeking.
- [147] and is being peeked at by A.
- [148] and is inhibited from being peeked at.
- [149] and is being PEEKed at by no one.
- [151] Line FID PS Return stack
- [152] TANDEM exited.
- [153] Can not TANDEM to your own line.
- [154] TANDEM cannot be used while another process
is PEEKing at specified process.
- [155] Another process is currently TANDEMing to this process.
- [156] Specified process is involved with TANDEM.
- [157] Specified process is TANDEM inhibited.
- [158] An illegal connective of the form "A" modifies "B".
- [163] An A-, F-, or T-correlative has failed compilation.
- [164] TOTAL or CONTROL-BREAK connective not succeeded by attribute
definition.
- [166] The A-correlative attribute name 'A' is illegal.
- [167] Missing terminal quote ("/") in A-correlative: 'A'.
- [168] Illegal A-correlative.: 'A'.
- [169] Missing left paren in A-correlative: 'A'.
- [170] Missing right paren in A-correlative: 'A'.
- [171] Missing right parenthesis 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.
- [181] Message 'A' entered in queue.
- [182] Data link control characters deleted from text.
- [183] End of queue.
- [190] Item-id size exceeds 50 bytes!
- [191] End tape check - A file(s) B record(s).
- [192] This is an erroneous object string.
- [193] Basic object code cannot be copied to an item.
- [194] Don't copy to the same item in the same file.
- [195] 'A' is not a list.
- [196] 'A' is too large to be an item.
- [197] The GET-LIST or SAVE-LIST specification is incorrect.
- [198] The list pointed to by pointer item 'A' is defective.
- [199] Insufficient work space for item 'A'.

[200] File name?
 [201] 'A' is not a file name.
 [202] 'A' not on file.
 [203] Item name?
 [204] File definition 'A' is missing.
 [205] No statements to be assembled.
 [206] 'A' assembled.
 [207] Undefined symbols.
 [208] Error in item-id list!
 [209] Columns must be in range 1-80.
 [210] File 'A' is access protected!
 [211] 'A' No assembled code can be found.
 [212] 'A' Has no frame statement.
 [213] 'A' Location counter error at line no. E.
 [214] 'A' Overflows frame B at line no E.
 [215] 'A' Hex error at line no. E.
 [216] 'A' loaded; frame = B size = C cksum = D .
 [217] 'A' verified; frame = B size = C cksum = D .
 [218] 'A' frame = B has F mismatches.
 [219] Illegal command: A
 [220] 'A' exited.
 221 'A' filed.
 222 'A' deleted.
 [223] 'A' exists on file.
 [224] 'A' is a pointer item. It may not be loaded into a non-pointer file.
 [225] TSYM must be DICT & DATA
 MOD/SEP 1,1 37,1
 Sorry for the inconvenience.
 [226] Tape format error!
 [227] Frame # A - verified.
 [228] Frame # A - B mismatches.
 [229] 'A' is not a valid account name.
 [230] 'A' Has invalid frame number.
 [234] Item size exceeds 32,000 bytes.
 [235] Attempt to write into update protected file!
 [236] File definition item 'A' not deleted.
 [237] Program 'A' compiled! B frames used.
 [240] A "SELECT" or "SSELECT" must be used preceding a 'SAVE-LIST'
 statement!
 [241] Program 'A' compiled! B frames used.
 [242] 'A' decataloged.
 [243] 'A' list saved. B frame(s) used.
 [244] 'A' cataloged!
 [245] List 'A' deleted.
 [246] 'A' is no longer a list or no longer on file.
 [247] 'A' is a binary (object code) item.
 [248] Basic object code or list item 'A' not overlayed.
 [249] An 'F' option is not valid with this verb.
 [250] Punt

[251] Done
 252 Frame=A, Size=B, Branches shortened=C, Hardness=D, Lines=E
 [260] Invalid batch lock command: A
 [265] Proc stack overflow!
 [267] Proc transfer to 'A' cannot be completed.
 [268] The destination of the proc "GO" statement:
 A, cannot be found.
 [269] Proc primary input buffer has exceeded 300 bytes!
 [270] Format error in the proc statement: A
 [272] A value exists for the attribute referenced by the element: A
 [273] Error in column-number/field-width or format specification at : A
 [274] Unrecognizable batch-string element : A
 [275] Y or F sub-element error at batch-string element : A
 [276] D-2 update without D-1 being specified, at batch-string element : A
 [277] J element missing at batch-string element : A
 [278] Error in processing secondary batch-string element : A
 [279] Incorrect scaling factor in F* batch-string element : A
 [280] File-definition batch element error at : A
 [281] D1 must have Y11 storage correlative ... Error at : A
 [282] Invalid parameter for select list.
 [287] Timeslice for line A equals B milliseconds.
 289 Terminal Printer
 Page width: A B
 Page depth: C D
 Line skip : E
 LF delay : F
 FF delay : G
 Backspace : H
 Term type : I
 [290] The range of the parameter "A" is not acceptable.
 [291] 'A' file-definition is missing.
 293 Total number of contiguous frames :A
 [295] Overflow table 11:18:37 17 OCT 1988 :
 [298] Format error in specifications.
 [299] Process A is tandeming to process B.
 [300] Process A is not using TANDEM.
 [301] and is being tandemed to by A.
 [302] and is inhibited from being tandemed to.
 [303] and is tandeming to no one.
 [304] TANDEM allowed by others for specified process.
 [305] TANDEM inhibited by others for specified process.
 [306] Must be floppy!
 [307] Illegal track number.
 [308] Illegal sector number
 [316] Numeric parameter missing.
 [318] Count too high.
 [330] Illegal line number.
 [331] The account file is missing.

335 Copyright 1982, Pick Systems, as an unpublished work. All rights reserved. This work is the property of, and embodies trade secrets, and confidential information proprietary to Pick Systems, and may not be reproduced, copied, disclosed, used, transferred, adapted or modified without the express written approval of Pick Systems.

```

          <<<      General Automation, Inc.      >>>
          <<<  Version: 3.8   C/L MON: a   ABS: 0   >>>
          <<<  11:18:37      17 OCT 1988   >>>
336 < Logged off at 11:18:37      on 17 OCT 1988      >
[337] User is not logged on.
[338] Account file statistics were not updated due to either:
      1. Insufficient work-space to contain the account file item, or
      2. SYSTEM dictionary changed while you were logged on.
[339] Improper or undefined format for date conversion.
440 < Connect time= A mins.; CPU= B units; LPTR pages= C      >
[341] Zebra Pick ABS version V(3) system verified!!!
[342] *** Zebra Pick ABS version V(3) system does not verify! ***
      There are A frame(s) with mismatches
[343] ABS version V(3).
      Frame(s) Averified!!
[344] ABS version V(3).
      Frame(s) A do not verify!!
      There are B frame(s) with mismatches.
[350] There is at least one user logged on account 'A'.
[390] 'A' loaded.
[391] 'A' verified.
[398] The maximum of 20 levels for a "WITHIN"-type statement has been
      exceeded.
[399] The file requires a V(ertical) correlative for this statement.
[400] File definition item 'A' not overwritten.
[401] No items present.
[402] Item 'A' is a 'D' pointer in the secondary file.
[403] End of list.
[404] A items selected.
[405] A items listed.
406 Item count= A, Byte count= B, Avg. Bytes/Item= C Avg.
      Items/Group= D Std. deviation= E, Avg. Bytes/group= F.
[407] A items counted.
[408] One item counted.
[409] Item 'A' is not in the secondary file.
[410] A synonym (Q-type) file cannot be specified in this statement.
[411] 'DICT' or 'DATA' must be specified in a clear-file statement.
[412] Insufficient disk space available for the file.
[413] The file name already exists in the master dictionary.
[414] Illegal or missing modifier used in defining the file area(s).
[415] Item 'A' exists on file.

```

[416] Range error in modulo or separation parameter.
 [417] File 'A' created; Base = B, Modulo = C, Separ = D.
 [418] File definition item 'A' not copied.
 [419] The specified file cannot be cleared or deleted!
 [420] Dictionary file deletion cannot be done without
 deletion of data-section(s) first.
 421 Statistics of A :
 Total = B Average = C Count = D
 422 Byte statistics for : A
 Total = B Average = C Items = D Cksum = E Bits = F
 423 Total of A = B
 424 Frame id : A
 Device address B
 Unit number .. C
 Cylinder D
 Head E
 Sector F
 [425] Invalid frame-id request!
 [426] Data file already exists!
 [427] There is no data section for this file.
 [428] 'A' object pointer not found.
 [429] 'A' program object verifies.
 Compiled on release B.
 [430] 'A' program object does not verify.
 Compiled on release B.
 [431] Unable to copy out of a pointer file!
 [432] 'A' Item name used in program-file dictionary, compile aborted.
 [433] Account name missing or invalid.
 [434] File definition items may only be copied to dictionaries
 at the same level as the source dictionary.
 [435] Source and destination files must be dictionaries for COPY-FILE.
 [436] At least one user is logged onto the account specified for deletion.
 [520] No print string for BLOCK-CONVERT.
 [521] Too many characters in word to block.
 [522] BLOCK-CONVERT file missing or improperly defined!
 [523] Block output would exceed page width.
 [524] The letter 'A' is not in the BLOCK-CONVERT file!
 [525] Input character 'A' is improperly formatted in BLOCK-CONVERT file!
 [530] Already logged on.
 [531] Process roadblocked.
 [532] Illegal user id!
 [533] Logon successful.
 [534] Logoff successful.
 [535] Illegal line number.
 [536] Already logged off.
 [537] Printer line cannot be logged off.
 [538] X-OFF function disabled.
 [539] X-OFF function enabled.

[540] Break-key disabled.
[541] Break-key enabled.
[542] Line has been cleared of roadblock.
[543] X-OFF function enabled, Computer port assigned.
[550] A required numeric parameter is missing or invalid.
[551] Specify number of lines to skip.
[552] Item 'A' has invalid format.
[560] End MT-CHECK processing.
[580] More data required to send message.
[581] No over-flow available to send message.
[582] ACC is not present. Unable to send message.
[583] Lines message buffer is full. Unable to send message
[584] User not logged on. Message not sent.
[585] Line is currently in the debugger. Message not sent.
[599] Object code in 'A' is in incorrect format.
600 'A' not converted.
601 'A' converted.
[604] 'A' is an undefined label reference.
[630] 'STOP' in effect. Use 'RESUME'!
[631] Spooler is waiting for printer cable to be attached!
[632] Spooler is waiting because the printer is off-line!
[633] Spooler is waiting for the tape to be detached!
[634] Spooler is waiting for the printer to be detached!
[635] Spooler is waiting because the tape is off-line!
[636] There was a parity error on writing a spooled file!
[637] The spooler needs the next reel of tape to continue!
[638] The spooler is writing a file on tape.
[639] The spooler is outputting a file to its console.
[640] The spooler is printing a file on the printer.
[641] Hold file #A added;
[644] An open file for line # A is being processed.
[645] A closed file for line #A is being processed.
[646] The current job has been aborted.
[651] Spooler is not active.
[652] Too many pages!
[653] No file being output or file originated on another line!
[654] Illegal output string length!
[655] Not attached!
[656] End of spool queue.
[657] Not a hold file.
[658] Illegal number! Use SP-STATUS.
[659] Line printer ready.
[660] Insufficient privileges; or device is attached to a spooler!
[661] Tapeout terminated because of ASSIGN T.

[700] Run-time F-correlative abort.
[701] Invalid function correlative definition : A
[704] Invalid P-Conversion definition : A
[705] Illegal conversion code : 'A'
[706] The translate conversion-code : A is illegal.
[707] DL/ID entry for T-conversion : A not found.
[708] 'A' cannot be converted.
[710] Invalid L- or R-conversion definition : A
[712] Value test impossible on attribute with F-conversion: A
[780] Item 'A' not on file.
[781] 'A' added.
[782] 'A' updated.
[783] 'A' deleted.
799 'A' is not a compatible item.
[800] < A > items dumped.
[802] A items dumped.
[803] A item(s) loaded
[804] A item(s) deleted.
[805] A item(s) copied.
[806] A items updated.
[807] A item(s) deleted.
[808] A item(s) moved and renamed.
[820] Invalid modem control parameter.
[821] Invalid XOFF character value.
[822] Invalid XON character value.
[823] Invalid auto off on CD parameter.
[824] Invalid auto off on DSR parameter.
[825] Invalid bit 8 enable parameter.
[826] Invalid buffer full warning character.
[827] Invalid buffer full warning enable parameter.
[828] Invalid receive XON/OFF enable parameter.
[829] Invalid transmit XON/XOFF enable parameter.
[830] Invalid type ahead enable parameter.
[831] Invalid number of stop bits.
[832] Invalid number of data bits.
[833] Invalid parity parameter.
[834] Invalid external status change interrupt parameter
[840] Disk cache activated on line A.
[841] Disk cache de-activated on line A.
[842] Cache not available.
[843] Illegal process attempting to execute cache command.
[844] Disk cache buffers are full.
[850] Node not available.
[851] Account name too long!
[852] Invalid node ID.
[853] Unexpected message received!
[854] LAN not running on this machine. Use LANSTART verb.
[857] Too many LAN requests are pending. Try again later.
[858] File name too long!
[859] Invalid LAN queue element on remote machine!

[860] Check code error.
 [861] No LAN controllers present !
 [862] Local area network status at 11:18:38 17 OCT 1988

Dev Num	Node Id	Chip Status	Ints Mask	Controller Address	Buffer Address	Receive Buffer	Virt Stat	Cont Bits	Xmit m/sec	LANP Line #
A	B	C	D	E	F	G	H	I	J	

Active node status :

[863] Access statement failed on remote machine.
 [864] LAN started.
 [865] LAN stopped.
 [866] LAN must be stopped first. Use LANSTOP verb.
 [867] Local Area Network status at 11:18:38 17 OCT 1988
 Node ID is A
 Active node status : B
 [870] LAN start in progress.
 [871] LAN stop in progress.
 [872] LANP is running on line # A.
 [873] SELECTP is running on line # A.
 [874] The LANP que table for jobs in progress is full.
 [875] SELECTP has run out of overflow space.
 [876] LANP has run out of overflow space.
 [900] Problem has occurred in the create account process.
 Please start over.
 [901] 'A' account created.
 [923] * * Illegal user ID * *
 [924] * Illegal password !!
 [949] Illegal frame-ID in primary space! program terminated.
 [950] Linkfield error - group at A - Frame B
 Links: C D E F
 [951] Illegal forward link. Group at A Frame B
 Links C D E F
 [952] Illegal backward link. Group at A Frame B
 Links C D E F
 [953] Should link back to: Group at A Frame B
 Links C D E F
 [954] A new error items created in dict TSYM.
 987 7F
 [990] Error in ABSDUMP frame limits specifications.
 [991] M/DICT entry missing, or required specification(s) missing.
 [992] A item(s) have been restored.
 [993] Account name must be specified.
 [995] Terminal type is : A

11:37:04 17 OCT 1988

Memory	Lines	PCBO	WSSTART	WSSIZE	SYSBASE/Mod/Sep	Maxfid	Avail.	overflow
AK	B	C	D	E	F	G	H	I

SET-CT

[1000] Which attributes ?
[1001] Coupled set specification error.
[1002] WINDOW and TAG connectives not allowed.
[1003] Dictionary attribute 13 improperly formatted.
[1004] 'A' not on file.
1006 Item 'A' exists on file.
1011 System error in the selection processor.
Try to generate a reproducible case.
Contact your system support analyst.
1050 That is an obsolete basic verb (RUN or CATALOG).
Please obtain the current definition.
1051 That is a program cataloged under pre-R80 protocols.
Please recatalog the program.
[1060] Tape is set to line A
[1061] Tape is on line A
[1062] No tape line set.
[1063] Bad line specification.
[1064] Tape line cleared.
[1070] Com is set to line A
[1071] Com is on line A
[1072] There is no com line.
[1073] Bad com line specification!
[1074] Com attached.
[1075] Com attached to line A
[1076] Comm line detached.
[1077] Bad line specification.
[1078] Com line cleared.
1099 Hold entry #
1100 Start code locked.
1101 Null printer number.
1102 Printer number too big.
1103 No form number.
1104 Illegal character.
1105 Printer must not be active.
1106 Form number too big -- exceeds 124.
1107 Too many pages in the page skip -- exceeds 9.
1108 Negative number.
1109 Too many output queues.
1110 Illegal printer type -- must be 'S'.
1111 Illegal line number.
1112 Illegal parallel printer number.
1113 Illegal serial printer number.
1114 The line which you specified is being used
as another printer on the system.
1115 Allocation attempted on uninitialized printer.

1116 There is no job enqueued for output on the forms you specified.
 Therefore, alignment is impossible.
 1117 Your align was just aborted by someone.
 You must start the align process over.
 1118 The printer control block has been initialized.
 Check for correct printer form and LPI.
 1119 You are attempting to start printer A on line B,
 which is not stopped.
 1121 You are improperly logged on.
 1122 Your output specification is no output.
 Reassign your line if you wish to output a holdfile.
 1123 Line # A is already logged on.
 [1126] Printer # A control block hammered.
 Cleared to null.
 [1127] Illegal printer number.
 [1129] A form queue elements unlinked.
 1130 Printer list elements 17 OCT 1988 11:18:39

	Stat	LK	LN	Curpos	Begfid	CP	FO	Frms	Date	Time	Acct
1131									17 OCT 1988	11:18:39	

	#	Stat	LK	LN	Statuses	CP	FO	Frms	Date	Time	Acct
1132											
1133											
1134									11:18:39		

	Printer	Output queues	Page	Dev or	Status
	Type	Number	Skip	Line #	
1135	Form queue	A			
[1140]	Your open files were closed.				
1141	Line	Status	Cop	form	
	#	ies	#		
1143	Align terminated; printer stopped.				
1144	Tape not available at this time.				
[1145]	Illegal specification number A.				
[1147]	Tape not attached!				
1148	Tapeout terminated because of assign T or null assignment.				
1149	A syntax error has occurred. Please contact your support analyst.				
1150	There is something wrong with the syntax of your verb's options.				
1151	Entry #	A			
1160	Your output specification is no output. Reassign your line if you wish to output a holdfile.				
1161	End of requested print files.				
1162	End of print file control block.				
[1169]	Illegal printer number. Must be between 0 and 3 inclusive.				

1170 Printer # A set to stop
 1171 and is inactive.
 1172 but is still active.
 1173 Printer # A control block hammered. Cleared to null.
 1174 is unallocated.
 1175 Parallel printer # A has been deleted.
 1176 Serial printer # A has been deleted, and its process sent to
 logon.
 1177 Printer # A is inactive.
 1178 The job being output on printer # A is not your print file.
 1179 Job aborted on printer # A
 1180 Print file # A was not unlinked because it is being output.
 1181 Print file # A was not unlinked because it is unused.
 1182 Print file # A was not unlinked because it is not spooled.
 1183 Print file # A was not created on the account onto which you
 are now logged.
 1184 Print file # A was unlinked and is available as a hold file.
 1200 The spooler is inactive.
 1201 The spooler is active.
 1202 needs to start printers
 1203 needs to log disk errors.
 1209 The control block for printer # A is in an ambiguous state.
 Delete the printer from the spooler system.
 1210 Printer # A is
 1211 unallocated.
 1212 serial
 1213 parallel
 1214 , inactive
 1215 , active
 1216 , stopped
 1217 , and on line.
 1218 , and off line.
 1219 *** The printer cable is off. ***
 1220 *** There is no controller for this printer. ***
 1221 The printer is defined as parallel printer # A.
 1222 The printer is running on line A.
 1229 Print file being output is element A. Aan open file for line # A
 1231 closed file for line # A
 1232 generated on account A
 1233 , which is A frames long.
 1234 and the output is choked.
 1235 .
 1239 no output queues assigned to printer.
 1240 Assigned output queues: A
 1241 , A
 1242 , A.
 1243 The number of inter-job pages to eject is A.
 [1300] A

2000

* * * TCL STACKER HELP MENU * * *

Command	Description
.L{n}{,m}{(p)}	= List n lines of stack starting at line m (Default: n=20,m=1) (p)-option lists line (p)'s stack
.R{n/old/new}	= Replace command (Same as Editor function 'R') .R{n}{U}{m}/old text/new text (or) .R{n}{,m}/old text/new text, n = starting line # (Def: n=1), U = change all, m = # lines
.Tn	= Put line n at Top-of-Stack
.Q	= Display stack status
.X{a,b,...c}	= Execute command line a, then b, etc. (Default: a=1)
.? {(P)}	= Display TCL STACKER HELP menu (P-option outputs to spooler)
.H{elp} {(P)}	

2001

* * * TCL STACKER HELP MENU * * *

Command	Description
.D{E}{n}{-m}	= Delete line n thru m (Default: n=1, m=1)
.E{n}	= Edit stack starting at line n (Default: n=1)
.Tn	= Put line n at Top-of-Stack
.O	= Toggles stacker on/off
.Q	= Display stack status
.L{n}{,m}{(p)}	= List n lines of stack starting at line m (Default: n=20,m=1) (p)-option lists line (p)'s stack
.R{n/old/new}	= Replace command (Same as Editor function 'R') .R{n}{U}{m}/old text/new text (or) .R{n}{,m}/old text/new text, n = starting line # (Def: n=1), U = change all, m = # lines
.U	= Toggles UPPER CASING of TCL statements on/off
.X{a,b,...c}	= Execute command line a, then b, etc. (Default: a=1)
.? {(P)}	= Display TCL STACKER HELP menu (P-option outputs to spooler)
.H{elp} {(P)}	

2002

* * * TCL STACKER HELP MENU * * *

Command	Description
.D{E} {n} {-m}	= Delete line n thru m (Default: n=1, m=1)
.E{n}	= Edit stack starting at line n (Default: n=1)
.K	= Delete entire stack
.Tn	= Put line n at Top-of-Stack
.O{(n)}	= Toggles stacker on/off
.Q{(A)}	= Display stack status. (A)-option displays all line's status.
.L{n}{,m}{(p)}	= List n lines of stack starting at line m (Default: n=20,m=1) (p)-option lists line (p)'s stack
.R{n/old/new}	= Replace command (Same as Editor function 'R')
	.R{n}{U}{m}/old text/new text (or) .R{n}{,m}/old text/new text, n = starting line # (Def: n=1), U = change all, m = # lines
.U{(n)}	= Toggles UPPER CASING of TCL statements on/off
.X{a,b,...c}	= Execute command line a, then b, etc. (Default: a=1)
.? {(P)}	= Display TCL STACKER HELP menu (P-option outputs to spooler)
.H{elp} {(P)}	

2003 On!

2004 Off!

2005 On!

[2010] No search strings entered.

[2011] A item(s) found.

2012 * Help for SEARCH verb *

Options: A - Assembler output format
 C - Skip comment lines ('*' only)
 D - Display entire item if search string found
 E - Used with 'L'-option to find items not containing
 search strings
 I - Item id suppression
 L - Create list of items found containing search strings
 P - Output to spooler
 N - Inhibit stop at end of page
 O - Display line with search string
 S - Line number suppression if A-option not in affect
 Object code suppression if A-option in affect
 U - Ignore case of search string (Forces UPPER case output)
 X - Hex output if A-option not in affect

- [2321] Item 'A' error.
- [2389] Remote phrase definition item 'A' is illegal!
- [2390] Remote phrase 'A' missing.
- [2391] Assumed 'A' is an item-id, or modifies the item-id.
- [2392] Assumed 'A' is a print-limiter which modifies B.
- [2393] Assumed 'A' is a value used as selection criteria on attribute B.
- [2394] Assumed 'A' is a value used as selection criteria on attribute B.
- [2395] Assumed 'A' is a value used to modify sort clause on attribute B.
- [2396] Assumed 'A' is a value used as selection criteria on attribute B.
- [2398] Circular phrase definition encountered in phrase 'A'
- [2399] Uneven number of delimiters (" ") in phrase 'A'
- [2401] Execute initialized. A workspaces pre-allocated.
- [2402] The number of workspaces to be allocated must be in the range 0-250.
- [2403] The number of nested levels must be in the range 0-15.
- [2404] The EXECUTE-CONTROL file (in SYSTEM) must have a D/Code of "DCY".
- [2405] The EXECUTE-CONTROL file (in SYSTEM) cannot be opened.
- [2406] The maximum number of nested EXECUTE statements has been exceeded.
- 4321 Time IS A
- [8000] This option is not supported in this release.
- [8001] Specified file name must be dictionary level file.
- [8002] Updates may not be saved on a partial file-save tape.
- [8003] Update-only save is not backward compatible.
- [8004] Complete file-save must be done prior to an update-only save.
- [8005] Dict or Data level save are not backward compatible.
- [8006] The file-save process may only be run by one user at a time.
- [8007] The L option may not be used when multiple files are specified or a select list is active.
- [8008] "all Items" is not valid. Only file definition items (D pointers) may be specified.
- [8009] Save complete (A Files, B Items.)
- [8020] Restore complete (A Files, B Items.)
- [8021] The parent file (as specified in the input statement) must be a dictionary level file (dict, account, or SYSTEM).
- [8022] All users must log off before restoring the system!
- [8023] The level of the save tape too high for the specified destination file. This may occur when attempting to restore an account from a tape which contains only dictionary and/or data, or when attempting to restore a dictionary from a tape which contains only data level files.
- [8024] Account name must be specified.
- [8025] Dictionary name must be specified.
- [8026] Tape verified.
- [8027] Tape has A errors.
- [8028] File not found.
- [8029] The specified data frame size must be a multiple of 500.
- 9009 PQ-SELECT register out of range 1 - 5.
- 9012 Update attempted on an input file at statement: A
- [9020] Compilation aborted.
- [9021] PROC format error in label.

[9022] Duplicate label found.
 [9025] Format error in 'GO' statement.
 [9876] Param 1 =A
 Param 2 =B
 Param 3 =C
 Param 4 =D
 [9877] Par 1 =A
 Par 2 =B
 Par 3 =C
 Par 4 =D
 [10000] Null password not allowed.
 [10001] Item 'A' not decoded due to password mismatch.
 [10002] Item 'A' not coded because to do so would cause the item to exceed
 the maximum item size of 32,267 bytes.
 [10003] Item 'A' not updated because to do so would cause the item to exceed
 the maximum item size of 32,267 bytes.
 [4-S.1] Communications line has already been started.
 [4-S.2] Illegal line number (Not Available). A
 Cannot be line # 0 and must be less than the spooler line!
 [4-S.3] Illegal baud rate.
 [4-S.4] Comm line name must be ALPHA only.
 [4-S.5] Format error (COM-START).

COM-START {line},{baud},{1-4 alpha character name}

COM-START L (list configuration table)

[4-S.6] Comm line started.
 [4-S.7] Communication Line A attached.
 [4-S.8] Comm line(s) detached.
 [4-S.9] Invalid comm line, not started...
 [4-S.10] Com. Name has already been used.
 [4-S.11] No Communication Lines Have Been Established!
 [4-S.12] The Communication Assignment Table Has Been Cleared.
 [4-S.13] That Com. line is attached to user 'A'.
 [4-S.14] No Communication Lines Are Attached!
 [4-S.15] A Time out error has occurred on the communications line.
 [4-S.16] Excessive error rate---ABORTED
 [4-S.17] Checksum failure!
 [4-S.18] Echo failure on data transmission!
 [4-S.19] Format error (COM-LIMITS).

COM-LIMITS {Line},{Retry count},{Time out in seconds}
 {Parity count},{% rate},{Minimum blocks}

[4-S.19] Format error (COM-LIMITS).
 [4-S.20] Statistics accepted.
 [4-S.21] Attempted overwrite of an existing item or
 an attempt to transfer a binary item to a non-binary
 file on the receiving computer.
 [4-S.22] Communications Verb function completed.
 [4-S.23] Communications Verb function ABORTED.

[4-S.24] Communication Line A killed.

[4-S.999] This system was expressly built for : Four-S, Inc.
Customers name :General Automation, Inc.

- #1 *** Invalid command ***
- #2 *** Not enough work space ***
- #3 *** Field is beyond screen ***
- #4 *** Invalid field ***
- #5 *** No prompt in field ***
- #6 *** Field has a dependent association ***
- #7 *** Line feed invalid ***
- #8 *** Required field ***
- #9 *** Value must be unique ***
- #10 *** 'AT' clause step number invalid ***
- #11 *** Error in screen definition item ***
- #12 *** Validation not passed by literal data ***
- #13 *** Invalid data in data array ***
- #14 *** File doesn't exist ***
- #15 *** Escape invalid ***
- #16 *** "row,column" where row and column are numerics ***
- #17 *** Must be numeric ***
- #18 *** (A)scending or (D)escending sequence ***
- #19 *** (Y)es or (N)o ***
- #20 *** Must be five unique numeric function codes (nnnnn) ***
- #21 *** Invalid data ***

B0 A CATALOGED!

- [B1] RUN-TIME ABORT AT LINE A
- [B9] WRITE, DELETE, or CLEARFILE operation attempted on read only file.
- [B10] Line A. Variable has not been assigned a value; ZERO used!
- [B11] Line A. Tape record truncated to tape record length!
- [B12] Line A. File has not been opened!
- [B13] Line A. Null conversion code is illegal; No conversion done!
- [B14] Line A. Bad stack descriptor
- [B15] Line A. Illegal opcode: C
- [B16] Line A. Non-numeric data when numeric required; ZERO used!
- [B17] LINE A. Array subscript out-of-range
- [B18] LINE A. Attribute number less than 0 is illegal.
- [B19] LINE A. Illegal pattern
- [B20] LINE A. COL1 or COL2 used prior to executing a FIELD stmt; ZERO used!
- [B21] LINE A. MATREAD: Number of attributes exceeds vector size.
- [B22] Called program 'B' is not a subroutine.
- [B23] File 'B' has not been opened!
- [B24] LINE A. Divide by zero illegal; ZERO used!
- [B25] Program 'B' has not been cataloged!
- [B26] LINE A 'UNLOCK C' ATTEMPTED BEFORE LOCK!
- [B27] Line A. RETURN executed with no GOSUB or CALL.
- [B28] LINE A NOT ENOUGH WORK SPACE
- [B29] CALLING PROGRAM MUST BE CATALOGED
- [B30] LINE A ARRAY SIZE MISMATCH
- [B31] LINE A STACK OVERFLOW

[B32] LINE A PAGE HEADING EXCEEDS MAXIMUM OF 1400 CHARACTERS
 [B33] LINE A PRECISION DECLARED IN SUBPROGRAM 'C' IS DIFFERENT FROM THAT
 DECLARED IN THE MAINLINE PROGRAM
 [B34] LINE A FILE VARIABLE USED WHERE STRING EXPRESSION EXPECTED
 [B35] 'MD' invalid object of 'CLEARFILE'; ignored!
 [B36] SYSTEM DICT illegal object of 'CLEARFILE'; abort!
 [B37] Argument mismatch between programs [B]
 [B38] MATWRITE increased the number of attributes.
 [B39] Division overflow; result in doubt!
 [B40] Error detected during releasing data buffer.
 Check program,variable: [B]
 [B41] LINE A LOCK NUMBER IS GREATER THAN 47.
 [B42] NOT ENOUGH DESCRIPTOR SPACE
 [B45] Line A. Item name exceeds max. (50 characters)
 [B48] Line A string exceeds 32739 bytes
 [B49] Line A string exceeds 16,777,214 bytes.
 [B51] Negative value raised to non-integer value. zero returned.
 [B52] String length greater than 32K. Item B truncated!
 [B55] MAT READ/WRITE operation must be to/from a vector.
 [B56] 'A' invalid command to 'U60EO' user exit. Ignored!
 [B57] A is out of range for obsize set. Ignored!
 [B60] PROCWRITE attempted outside of PROC mode.
 [B98] Line A. Data buffer corruption or invalid descriptor detected.
 [B100] LINE A COMPILATION ABORTED; NO OBJECT CODE PRODUCED.
 [B101] LINE A MISSING "END", "NEXT", "WHILE", "UNTIL", "REPEAT" OR "ELSE"; .
 COMPILATION ABORTED, NO OBJECT CODE PRODUCED.
 [B102] LINE A BAD STATEMENT
 [B103] LINE A LABEL 'C' IS MISSING
 [B104] LINE A VARIABLE 'C' IS DOUBLY DEFINED
 [B105] LINE A 'C' HAS NOT BEEN DIMENSIONED
 [B106] LINE A 'C' HAS BEEN DIMENSIONED AND USED WITHOUT SUBSCRIPTS
 [B107] LINE A "ELSE" CLAUSE MISSING
 [B108] LINE A "NEXT" STATEMENT MISSING
 [B109] LINE A VARIABLE MISSING IN "NEXT" STATEMENT
 [B110] LINE A 'END' STATEMENT MISSING
 [B111] LINE A "UNTIL" OR "WHILE" MISSING IN "LOOP" STATEMENT
 [B112] LINE A "REPEAT" MISSING IN "LOOP" STATEMENT
 [B113] LINE A TERMINATOR MISSING
 [B114] LINE A MAXIMUM NUMBER OF VARIABLES EXCEEDED
 [B115] LINE A VARIABLE 'C' IS USED BEFORE THE EQUATE STATEMENT.
 [B116] LINE A NON-COMMON VARIABLE IS USED BEFORE THE COMMON STATEMENT.
 [B117] LINE A LABEL OR VARIABLE 'C' IS OF IMPROPER TYPE
 [B118] LINE A VARIABLE 'C' IS THE OBJECT OF AN EQUATE STMT AND IS MISSING.
 [B119] LINE A WARNING - PRECISION VALUE OUT OF RANGE - IGNORED!
 [B120] LINE A WARNING - MULTIPLE PRECISION STATEMENTS - IGNORED!
 [B121] LINE A LABEL 'C' IS A CONSTANT AND CAN NOT BE WRITTEN INTO.
 [B122] LINE A LABEL 'C' IS IMPROPER TYPE.
 [B123] THE PROGRAM CONTAINS AN EQUATE WHICH CANNOT BE RATIONALIZED.
 [B124] LINE A LABEL 'C' HAS LITERAL SUBSCRIPTS OUT OF RANGE.

[B125] LINE A LABEL 'C' HAS A JUMP GREATER THAN 32K BYTES.
 [B126] OBJECT CODE EXCEEDS 65K.
 [B127] OBJECT CODE AND SYMBOL TABLE EXCEED 65K.
 [B128] LINE A LABEL 'C' EQUATED ARRAY SUBSCRIPT OUT OF RANGE.
 [B129] LINE A DIMENSIONED ARRAY VALUE MUST BE GREATER THAN 0!
 [B130] THERE IS AN UNRATIONALIZED EQUATE.
 [B131] PRECISION statement smaller than numeric values in program.
 [B132] Line A. ENTER cannot be used within a SUBROUTINE program!
 [B133] Line A. Use of ENTER in subroutine 'C' is illegal!
 [B154] FOR STATEMENT WITH NO NEXT STATEMENT
 [B160] Line A. Program 'C' being ENTER'ed
 must be recompiled. No common descriptor size.
 [B161] Line A. Program 'C' using ENTER
 must be recompiled. No common descriptor size.
 [B162] The number of COMMON variables in SUBROUTINE 'A' exceeds
 the number of COMMON variables in CALLing program 'B'.

 Common variables in CALLing program: C
 Common variables in SUBROUTINE program: D

[B163] Line A. Program 'C' being CALL'ed
 must be recompiled. No common descriptor size.
 [B164] Line A. Program 'C' using CALL
 must be recompiled. No common descriptor size.
 [B199] Format error in source file definition.
 [B209] LINE A FILE IS UPDATE PROTECTED.
 [B210] LINE A FILE IS ACCESS PROTECTED.
 [B220] LINE A 'CSYM' IS NOT A FILE NAME OR NEEDS A DATA LEVEL.
 [B261] \$INCLUDE or \$CHAIN file name not found.
 [B262] \$INCLUDE or \$CHAIN item-id not found.
 [B263] Null item body used with \$INCLUDE or \$CHAIN.
 [B264] Cannot \$INCLUDE or \$CHAIN program within itself.
 [B641] Hold file #A added;
 [B705] LINE A. CONVERSION CODE IS ILLEGAL.
 NO CONVERSION DONE!
 [B850] NODE NOT AVAILABLE
 [B853] UNEXPECTED MESSAGE RECEIVED
 [B854] NO LAN PROCESSOR ON THIS MACHINE
 [B855] Illegal BASIC operation attempted on remote machine.
 [B8605] Too many parameters!
 [G1] Maximum axis size out of range!
 [G2] Plot size would require more than 10000
 frames working storage. The job was terminated.
 [G3] Requested plot width A characters is greater than device maximum.
 [G4] Requested plot depth A is greater than device maximum depth.
 [G5] Plot size would require A frames working storage.
 There are not sufficient frames available.
 [G6] Out of overflow space!
 [G7] Minimum value must be lower than maximum value!

- [G11] Plot table definition item 'A' is not on file.
- [G12] Plot table definition item 'A' has ended prematurely.
- [G13] Plot table item 'A' has too few values.
- [G14] Plot table item 'A' has too many values.
- [G15] Plot table item 'A' contains non-numeric data.
- [G16] Plot device definition item 'A' has illegal type code in line 1.
- [G17] There are less than 50 lines in the message table 'A'.
- [G18] There are less than 30 lines in the device definition table 'A'.
- [G21] A PLOT statement requires at least 2 plottable attributes.
- [G22] Compiled string format error @A
- [G23] History string format error @A
- [G24] Illegal attribute sequence number.
The specified number does not have the "bar" option selected.
- [G25] Illegal attribute sequence number.
The specified number is already stacked with another attribute.
- [G26] Illegal attribute sequence number.
The number must be greater than 0.
- [G27] Illegal attribute sequence number.
The number must be lower than the current attribute.
- [G28] Illegal attribute sequence number.
The specified number is already grouped with another attribute.
- [G29] The label area (and optional width) will exceed
the available plot width.
- [G30] The label area (and optional depth) will exceed
the available plot depth.
- [G31] The required label area will cause the
circle diameter to be less than 2 inches.
- [G36] The number of pixels per byte in line 23 of the device
definition table is out of range (4-8).
- [G37] The number of color zones in line 24 of the device
definition table is out of range (1-8).
- [G38] The specification for the character cell fill factor
in line 14 of the device definition table is incorrect.
- [G39] The specification for the number of markers available in
line 15 of the device definition table is incorrect.
- [G40] PLEASE CONTACT YOUR SUPPORT ANALYST IMMEDIATELY.
CODE=0,A,B,C
- [G41] PLEASE CONTACT SYSTEM SUPPORT ANALYST IMMEDIATELY!
CODE = 1,A,B,C
- [G42] PLEASE CONTACT SYSTEM SUPPORT ANALYST IMMEDIATELY!
CODE = 2,A,B,C
- [G43] PLEASE CONTACT SYSTEM SUPPORT ANALYST IMMEDIATELY!
CODE = 3,A,B,C
- [G44] PLEASE CONTACT SYSTEM SUPPORT ANALYST IMMEDIATELY!
CODE = 4,A,B,C
- [G45] PLEASE CONTACT SYSTEM SUPPORT ANALYST IMMEDIATELY!
CODE = 5,A,B,C

- [G46] PLEASE CONTACT SYSTEM SUPPORT ANALYST IMMEDIATELY!
CODE = 6,A,B,C
- [G48] PLEASE NOTIFY GENERAL AUTOMATION USING RESPONSE CARD
SERIAL NUMBER A WILL EXPIRE IN B DAYS. CODE = C.
- [G90] THE ACCUPLOT ACCOUNT HAS BEEN INITIALIZED.
THE FRAME ASSIGNMENTS NEED TO BE VERIFIED.
THE MODES NEED TO BE LINKED.
THE DEVICES NEED TO BE LINKED.
THE MODES NEED TO BE LOADED.
- [G91] THE ACCUPLOT FRAMES HAVE BEEN ASSIGNED.
THE MODES NEED TO BE LINKED.
THE DEVICES NEED TO BE LINKED.
THE MODES NEED TO BE LOADED.
- [G92] THE ACCUPLOT MODES HAVE BEEN LINKED.
THE DEVICES NEED TO BE LINKED.
THE MODES NEED TO BE LOADED.
- [G94] THE ACCUPLOT SYSTEM IS OPERATIONAL.

