PICK®

Operating System

PROGRAMMER'S REFERENCE GUIDE

$4.95

FILE AND ATTRIBUTE DICTIONARY DEFINITIONS

Attr. No.	Name	File Def. Item	Syn. Def. Item	Attr. Def Item
1	D/CODE	D, DX, DY, DC, DCX, DCY	Q	A
2	F/BASE Or A/AMC	Base FID Of file	Account-Name	Amc
3	F/MOD or V/TAG	Modulo of file	Synonym File-name	Tag or Heading
4	F/SEP or V/STRUC	Separation of file	Not used	Controlling/ Dependent
5	L/RET	Retrieval lock code(s)		Reserved
6	L/UPD	Update lock code(s)		Reserved
7	V/CONV	Conversion specification(s)		
8	V/CORR	Reserved		Correlative
9	V/TYPE	Justification on type code		
10	V/MAX	Maximum field length		
11 And 12 ..... Reserved				

ATTRIBUTE DEFINITION ITEMS

No.	Code	Processing
1	A	Defines item as an Attribute
2	N	AMC/defines attribute number n
3	Text	Columnar heading text
	\	Reserved for null tag
4	C or D	Defines Controlling/Dependent Attribute
7, 8	A(exp)	Arithmetic Correlative NI item counter NV value counter NS sub-value counter NB current break level ND detail line counter D system date T system time R(x,y) remainder of x/y S(n) sums multivalues of n [exp] sub-string arithmetic: +, -, *, / relational: <, >, >=, <=, =, * (exp) precedence (conversion)
	C	Concatenation
	D	Date conversion
	F(exp)	Same as A but in Reverse Polish
	G	Group extract
	L	Length restrictions
	MC	Manipulate Character
	M	Mask & Formatting
	MT	Mask Time conversion
	MX	Mask Hex conversion
	P	Pattern match
	R	Range output restriction
	S	Substitute value
	T	Text extract
	T	Translate
9	L	Left justify output
	R	Right justify output
	T	As L, folds excess at blank

FILE DEFINITION ITEMS

D	Defines item as a Dictionary or Data file.
DX	Do not save file on filesaves.
DY	Do not save data on filesave. Returns empty on restore
DC	Binary data (Basic Program file or Pointer-File).

TERM COMMAND

To Change:				
TERM (a, b, c, d, e, f, g, h, t)	a	=	Page Width: (TERM)	79
	b	=	Page Depth: (TERM)	24
	c	=	Line Skip:	0
	d	=	LF Delay:	1
To Display:	e	=	FF Delay:	1*
> **TERM** <CR>	f	=	Backspace:	21
	g	=	Page Width (Printer)	32
*0 = No Form Feed at	h	=	Page Depth (Printer)	64
initialization.	t	=	Term Type:	R

EDITOR COMMANDS

A	Again, repeats L(ocate) command
AS	Assembler format
B	Position pointer to item bottom
C	Display column numbers
?	Display file, name and line number
DE{n}	Delete current to n lines
DE{n}"str"{p-q}	Delete current to n lines if string found in columns p thru q
EX	Exits item work-copy
EXK	Exit item work-copy and Editor
F	Updates work-copy
FI{O{K{L}}}{(DICT{ filename}}{ item-id}	Update item — in specified file — with specified item-id, exits work copy; defaults to originating file and item-id. O — overwrite existing item K — exits Editor L — as a list
FS{O{L}}{(DICT{ filename}}{ item-id}	Same as above, but does not exit work-copy
FD{K}	Deletes item from file. K — exits Editor
Gn or n	Go to line n(umber)
I {data}	Insert, Input
L{n}	List n(umber) of lines; a space n is equivalent
L{n}"str"	Locate str, search n lines
M	Macro expansion, 1st On/2nd Off
MEn/{item-id}/m	Merge n lines, start at line m
MEn({DICT} filename item-id)m	Merge from another file
N{n}	Next n line(s)
n	Go to line n
P command(s)	Prestore Editor command
P{n}	Where n = 0 thru 9 of Prestore command to be executed
P thru P9	Pn any number of Editor commands separated by the esc-key
PD	Displays pre-stored commands
R{n}	Replace current for n lines
R{n}/str/str2/{p{-q}}	Replace n lines, str with str2
RU{n}/str/str2/{p{-q}}	Replace all str's with str2's (in columns p thru q)
S	Suppress, 1st On/2nd Off
S?	Displays item size
TB n n ...	Tab set at line(s) n
T	Go to top of item
U{n}	Go up n lines
X	Cancel preceding I/DE/R/ME command
Z{p{-q}}	Zone limits on all output

GENERATING ENGLISH/ACCESS SENTENCES

```
>verb filename {item-list} {selection criteria}
  {sort keys} {output specs} {print limiters}
  {modifiers} { (options . . . options) }
```

Forms of Filename

file-name	
dictname, dataname	
DICT file-name	
DATA file-name	

Item-list

Specifically enumerated item-id(s) enclosed in quotes or backslashes (',",\).

Selection Criteria

EVERY or EACH	All inclusive modifier
WITH or IF	Attr. Selection criteria
WITHIN	Sub-list connective
AND or OR	Connectives
WITHOUT	Excluding criteria

Logical Operators:

= or EG	> or GT or AFTER
<= or LE	< or LT or BEFORE
>= or GE	# of NE or NOT or NO

Sort Keys

BY	Attribute sort sequence
BY-DSND	Sort descending sequence
BY-EXP	Exploding attribute sort
BY-EXP-DSND	Exploding descending sort

Output Specifications

Dictionary attribute definition *Name(s)* for output. (Default of sequential numeric used.)

Print Limiters

Refers to specific attributes. Uses Logical Operators & Connectives as with Selection criteria. Does not use WITH.

Modifiers

BREAK-ON	Defines control — options: 'B' Break 'D' No Break on single lines 'L' Suppress preceding blank line 'P' Page eject after Break 'R' Inhibits page ROLLOVER 'U' Underline TOTAL fields 'V' Inserts value of control-break
COL-HDR-SUPP	Heading, column suppress
DBL-SPC	Double space
DET-SUPP	No detail with TOTAL or BREAK-ON
FOOTING	Footing "text" (see HEADING options)
GRAND-TOTAL	Grand total text — options: 'L' line suppress 'U' underline 'P' page-eject

HEADING	Heading "text" — options: 'B' Insert break value 'C' Centers text line 'D' Inserts current date 'F' Inserts file-name 'Fn' Left justifies file-name in n blanks 'L' Causes line feed 'P' Insert current page no. 'T' Insert current time and date '' Print a single quote
HDR-SUPP or SUPP	Heading suppress
ID-SUPP	Item-id suppress
LPTR	Output to line-printer
ONLY	Lists Item-id's only
NOPAGE	No page to Terminal
TAPE	Data from mag tape
TOTAL	Totals specified attr.
Options (enclosed in parentheses)	
C	Columnar suppress
D	Detail suppress
F	New page per item
H	Header suppress
I	Item-id suppress
P	Output to printer

HEXADECIMAL and DECIMAL CONVERSION

From hex: locate each hex digit in its corresponding column position and note the decimal equivalents. Add these to obtain the decimal value.

From decimal: (1) locate the largest decimal value in the table that will fit into the decimal number to be converted, and (2) note its hex equivalent and hex column position. (3) Find the decimal remainder. Repeat the process on this and subsequent remainders.

HEXADECIMAL COLUMNS												
6		5		4		3		2		1		
HEX	DEC	HEX	DEC	HEX	DEC	HEX	DEC	HEX	DEC	HEX	DEC	
0	0	0	0	0	0	0	0	0	0	0	0	
1	1,048,576	1	65,536	1	4,096	1	256	1	16	1	1	
2	2,097,152	2	131,072	2	8,192	2	512	2	32	2	2	
3	3,145,728	3	196,608	3	12,288	3	768	3	48	3	3	
4	4,194,304	4	262,144	4	16,384	4	1,024	4	64	4	4	
5	5,242,880	5	327,680	5	20,480	5	1,280	5	80	5	5	
6	6,291,456	6	393,216	6	24,576	6	1,536	6	96	6	6	
7	7,340,032	7	458,752	7	28,672	7	1,792	7	112	7	7	
8	8,388,608	8	524,288	8	32,768	8	2,048	8	128	8	8	
9	9,437,184	9	589,824	9	36,864	9	2,304	9	144	9	9	
A	10,485,760	A	655,360	A	40,960	A	2,560	A	160	A	10	
B	11,534,336	B	720,896	B	45,056	B	2,816	B	176	B	11	
C	12,582,912	C	786,432	C	49,152	C	3,072	C	192	C	12	
D	13,631,488	D	851,968	D	53,248	D	3,328	D	208	D	13	
E	14,680,064	E	917,504	E	57,344	E	3,584	E	224	E	14	
F	15,728,640	F	983,040	F	61,440	F	3,840	F	240	F	15	
0123		4567		0123		4567		0123		4567		
BYTE				BYTE				BYTE				

Command	Result
LISTPEQS	Displays the print file control block
Options:	
A	Displays only the print files created on account that you are logged on to
L	Displays those print files which are deleted as well as those which are active
'account'	Displays all print files generated on that specific 'account'
C	Displays only the number of print files and the amount of storage
n	Displays print file control block entry #n
n-m	Displays print files n through m
F	Displays only print files enqueued for output
LISTPTR {n{,m}}	List the currently allocated printer control blocks, including number, status, and form allocated where n = specific line number thru m
LISTABS {n}	Displays the assignment of each line or specific line n
SP-ASSIGN	Defines the print file destination
Options:	
H	Send to Holdfile
S	Suppress printing
I	Link on at initiation
T	Send to tape
C	Choke the process
O	Keep the print file open until closed or logoff
Rn	Initiate a print file with this SP-ASSIGN to be generated by PRINT ON n, where n is between 0 and 125 (used only in a basic program)
?	Displays the current line assignment
n	Number of copies to be printed 1-125
Fn	Form number between 0 and 125
SP-CLOSE	Terminated the SP-OPEN condition, so that the print file is closed
SP-EDIT	
Options:	
"nothing"	The default is to edit all hold files generated on the account onto which you are currently logged
U	Edit all hold files (SYS2 privileges)
'account'	Edit print files generated on account 'accountname' (SYS2 privileges)
n	Edit print file number n
n-m	Edit print files whose entry numbers are n through m
M	MASTER CONTROL — execute options as specified in remainder of statement
P	Print — overides current SP-ASSIGN S (does not change SP-ASSIGNMENT)
R	Causes output at current SP-ASSIGNment rather than that of time file was generated
S	Spool each hold file selected
D	Delete each hold file selected
T{W}	Spool to tape, optionally wait for tape drive

Command	Result
SP-KILL {n{-m}}	Kills the current output from printer n thru m where n defaults to zero
SP-KILL Fn{-m}/B	Disenqueues a print file waiting to be output where n = the entry number thru m or where B = all elements
SP-KILL Dn{-m}/B	Deletes printer n thru m or where B = all printers
SP-OPEN	Causes a sequence of jobs to be taken as one job for purposes of output
SP-STATUS	Displays the current status of the spooler and each defined printer
SP-TAPEOUT	Prints a tape created by the spooler
STARTPTR	Used for original printer set-up; changes to form numbers and page skips; and to restart a stopped printer (after a STOPPER n)

Examples used for initial printer setup:

```
STARTPTR 0,0,1,P0
         | | | |
         a b c d
```

a	This is the printer referenced as 0
b	Specifies that this printer will print jobs spooled under SP-ASSIGN F0
c	Eject 1 pages after each job
d	Specifies hardware address parallel printer 0

```
STARTPTR 8,3,0,S20  (SX66
         | | | |    | |
         a b c d    e f
```

a	This is the printer referenced as 8
b	Specifies that this printer will print jobs spooled under SP-ASSIGN F3
c	Eject 0 pages after each job
d	Specifies hardware address, port 20 as a Serial printer
e	Indicates that the initial form-feed command at the start of a print file is to be ignored by a serial printer
f	Specifies line counting size 66

STOPPTR{n}	Flags the specified printer that it is to stop at the end of the current print file where n = printer number (defaults to 0)

PERIPHERAL DEVICES (TAPE DRIVE)

Command	Result
T-ATT {n}	Attaches the tape drive to a user's process with a block size of n
T-DET {U}	Detaches the tape from the user's line, allowing other processes access to it; where U = unconditional detach from any user's line
T-FWD n	Moves the tape forward n # of records
T-BCK n	Moves the tape backward n # of records
T-SPACE n	Causes multiple T-FWDs n # of files
T-EOD	Moves the tape forward to end-of-data
T-REW	Rewinds the tape back to BOT (beginning of tape)
T-WEOF	Writes an end-of-file mark on the tape
T-CHK {A}	Checks the tape for parity errors; where A = check to EOD
T-DUMP filename {item(s)}	Moves data from disc files to tape
T-LOAD filename	Moves data from tape to disc files
Options:	
I	Inhibits terminal output
O	Overlay existing items
T-READ	Allows inspection of the contents of a tape
Options:	
A	EBCDIC to ASCII
X	In hex
P	To printer
n-m	n = current through m
T-RDLBL	Allows inspection of the tape label
S-DUMP filename	Sorts data, then moves the data from a disc file to tape (same options as available in ENGLISH)

BACKING UP HOLD FILES PRIOR TO FULL FILE RESTORES

To back up all spooled hold files on tape or diskette prior to executing a full file restore:

```
> T-ATT XXXX   (XXXX = block size)
> SP-ASSIGN (T)
> SP-EDIT (MUST)
```

To reload all saved files from the above procedure following a Full File Restore:

```
> T-ATT XXXX
> SP-ASSIGN HS
> SP-TAPEOUT
```

COMPILE filename program-name {(options)}
CATALOG filename program-name
  (creates program-name verb in account's MD)
DECATALOG program-filename program-name
RUN filename program-name {(options)}

BASIC PROGRAM STATEMENTS

ABORT {errnum{,param,param,...}}
BREAK OFF inhibits use of the break key
BREAK ON restores use of the break key
CALL variable (argument list) subroutine call
CALL subroutine (argument list) subroutine call
CASE --- BEGIN CASE
CASE exp
stmts
CASE exp
stmts
•
•
END CASE
CAT or : — concatenation
CHAIN exp "any TCL command"
CLEAR initializes all variable to zero
CLEARFILE {file-var}
CRT same as print, but always to terminal
COM{MON} var{,var} — values passed to subroutines
DATA exp — data to be passed to TCL PROC stack
DELETE {file-var,} item-name
DEBUG — sends program into Basic Debugger
DIM vector(# of elements) {,vector(# of elements)}
DIM matrix(rows, columns) {,matrix(row,columns)}
ECHO OFF terminal input not displayed
ECHO ON terminal input display back to ON
END
ENTER catalogued-program — (passes any COM variables)
ENTER var.program (same as above)
EQU{ATE} secondary-var TO primary-var {, ...}
FOOTING exp — set up data to print at bottom of pages
FOR var = exp TO exp {STEP exp} {WHILE or UNTIL exp}
GO{TO} stmt-label
GOSUB stmt-label
HEADING exp
IF exp {THEN stmts/ELSE stmts}
INPUT var {,exp}{:}
INPUT @ (x,y): var mask
INPUT ERR exp — display special error messages
INPUT TRAP 'xx...' GOTO n,n... — acts on INPUT character
INPUT TRAP 'xx...' GOSUB n,n,... — acts on INPUT character
INPUT NULL x character used to null the default input
LOCATE(str,item{,attr#{,val#}};var{;seq}) {THEN stmt}
  and/or {ELSE stmt}
LOCK exp {THEN/ELSE stmts}
LOOP {stmts} UNTIL or WHILE exp DO {stmts} REPEAT
MAT array = exp or MAT array = MAT array
MATREAD{U} array FROM {file-var,}item-name
  THEN/ELSE stmts
MATWRITE{U} array ON {file-var,} item-name
NEXT var — used in conjunction with FOR stmt
NULL — specifies a non-operation
ON exp GOTO or GOSUB stmt-label {,stmt-label, ...}
OPEN{DICT,} file-name {TO var} THEN/ELSE stmts
PAGE exp — used to set page counter
PRECISION n — range of 0 to 4

PRINT {ON exp} expression
PRINTER exp/ON/OFF/CLOSE
PROCREAD var {THEN exp and/or ELSE exp} — reads entire primary input buffer
PROCWRITE exp — writes exp into primary input buffer
PROMPT character-exp
READ{U} var FROM {file-var,} exp THEN/ELSE stmts
READNEXT var {,var}{FROM select-var} THEN/ELSE stmts
READT var THEN/ELSE stmts
READV{U} var FROM {file-var,}item-name,attr# THEN/ELSE stmts
RELEASE {{var,} exp}
REM or * or ! Comments (must be first character of stmt)
RETURN {TO stmt-label} — used with GOSUB
REWIND THEN/ELSE stmts
RQM {time or seconds} — release time quantum
SELECT {file-name/file-var}{TO select var}
SLEEP {time or seconds}
STOP {errnum{param,param,...}}
SUBROUTINE name (argument list)
UNLOCK {exp}
WEOF THEN/ELSE {exp}
WRITE{U} exp ON {file-var,} item-name
WRITET exp THEN/ELSE stmts — tape write
WRITEV{U} exp ON {file-var,} item-name,attr#

INTRINSIC FUNCTIONS

@(column{,row}) — cursor control

Special cursor function values:

-1 = Clear and Home
-2 = Home /upper left
-3 = Clear current position to end of screen
-4 = Clear to end of line
-5 = Start blink
-6 = Stop blink
- 7 = Initiates 'protect'
- 8 = Stops 'protect'
- 9 = Back one character
-10 = Moves cursor up one line
-99 = Turn off page flag, zeroes line counter

ALPHA(exp) — true(1) if alpha
ASCII(exp) — converts EBCDIC to ASCII
EBCDIC(exp) — converts ASCII to EBCDIC
SEQ(exp) — converts ASCII to Decimal
CHAR(exp) — converts Decimal to ASCII

Common uses of CHAR:

7 = Bell
12 = Clear screen, or Form Feed
252 = Secondary Value Mark
253 = Value Mark
254 = Attribute Mark

INTRINSIC FUNCTIONS (continued)

COUNT (string,substring)
DATE() — internal format
DCOUNT (string,substring)
DELETE(array,attr.#{,val.#{,sub-val.#}})
EXTRACT(array,attr.#{,val.#{,sub-val.#}}) or
array<attr.#{,val.#{,sub-val.#}}>
FIELD(string,delimiter,occurrence #)
COL1() returns preceding column position of the substring located in last FIELD stmt
COL2() returns following column position of the substring located in last FIELD stmt
ICONV(exp,conversion type) — external to internal
INDEX(string,substring,occurrence #)
INSERT(array,attr.#{,val.#{,sub-val.#}};new value)
LEN(exp) — returns length of string expression
NOT(exp) — true(1) if evaluates to zero
NUM(exp) — true(1) if evaluates to numeric character(s)
OCONV(exp,conversion type) — internal to external
REPLACE(array,attr.#{,val.#{,sub-val.#}};new value) or
array<attr.#{,val.#{,sub-val.#>= new value
SPACE(number of blanks)
STR(exp,# of occurrences)
TIME() — internal format
TIMEDATE() — external format
TRIM(exp) — removes extraneous blanks
SYSTEM (exp)
SYSTEM(1) = returns a 1 if the printer is turned on
SYSTEM(2) = page size (based on TERM setting)
SYSTEM(3) = page depth (based on TERM setting)
SYSTEM(4) = lines remaining to be printed (based on TERM setting)
SYSTEM(5) = page number
SYSTEM(6) = line counter
SYSTEM(7) = terminal type (based on TERM setting)
SYSTEM(8) = tape record size
SYSTEM(9) = the current CPU units in miliseconds
SYSTEM(10) = returns a 1 if the source of input is the stack, and 0 if the next input statement will request input from the terminal

Please note that 4 thru 6 only have relevance in BASIC when a HEADING or FOOTING is in effect.

ARITHMETIC FUNCTIONS

ABS(exp) = absolute value
EXP(exp) = exponential
LN(exp) = natural logarithm
INT(exp) = returns integer
REM(x,y) = remainder of x/y
PWR(x,y) = power x to y
COS(exp) = cosine
RND(exp) = random number
SIN(exp) = sine
TAN(exp) = tangent
SQRT(exp) = square root

ARITHMETIC and LOGICAL OPERATORS

+ unary plus	< or LT	less than
- unary minus	> or GT	greater than
* multiplication	= or EQ	equal
/ division	# or >< or NE	not equal
+ addition	<= or LE	less or equal
- subtraction	>= or GE	greater or equal
↑ exponentiation		

Programmer Debugger entry:

```
>RUN program-file program-name (D) [CR]
*E1
*Z [CR] PROG NAME? File-name program-name [CR]
```

COMMANDS

Bvoc{voc}	Breakpoint set on variable condition where v is variable, c is condition, and o is logical operator <,>,=,#.
B$on	Breakpoint set on line no. condition where o is logical operator <,>,=,#; and n is line number. (Limit of 4 breakpoint settings.)
D	Display Trace & Breakpoint tables
DE or DEBUG	Escape to System Debugger
En	Single/multiple step execution
END	End program execution & return to TCL
G	Proceed from breakpoint
Gn	Proceed to specified line 'n'
K	Remove all breakpoints
Kn	Remove specified breakpoint 'n'
L	Display source code current line
Ln	Display source code 'n' no. of lines from current line
Lm-n	Display source code no. of lines from m-n
L*	Display source code all lines
LP	Switch output from terminal to printer or from printer to terminal
N	Pass one breakpoint before stopping
Nn	'n' breakpoints
OFF	LOGOFF
P	Inhibit output
PC	Printer-close output to spooler
R	Pop return stack
S	Display return stack
T	Switch turns trace table on/off
Tv	Trace specified variable 'v' (Limit of 6 trace settings.)
U	Remove all traces
Un	Remove trace n (1-6) from table
V	Verifies object code
Z	Request symbol table
$	Display current line number
/v	Print value of variable 'v'
/m(x)	Print value of element 'x' in array 'm'
/m(x,y)	Print value of element 'x,y' in matrix 'm'
/m	Print value of entire array 'm'
/*	Print entire symbol table
[x,y]	Zone limits on output
[	Remove zone limits
?	Display name, line number, and verifies

THE COPY PROCESSOR:

>COPY {DICT} filename item-list (or *) {(options)}

The COPY processor copies items from a file or dictionary to another file or dictionary; or to the terminal or printer.

OPTIONS:

D — Delete source item after copy is successful.

F — Form-feed. Initiates new page with each item.

I — Inhibits listing of the item-ids.

N — On file-to-file; New item inhibit. Copies only items which already exist in destination file. Does not create new items.

Terminal/Printer destination: Inhibits end of page wait.

O — Overlay. Copies ALL items, overlays those which already exist in the destination file.

P — Copies items to the line printer/spooler.

S — Inhibits error message display on terminal or printer output.

T — Terminal copy only.

X — Displays in hexadecimal format on terminal or printer print out.

The COPY processor may utilize a pre-selected list as follows:

```
>GET-LIST XYZ <CR>
  2 ITEMS SELECTED
>COPY FILE1 <CR>
TO: (FILE2 <CR>
     1 ABC
     2 DEF
2 ITEMS COPIED
```

In the above, Items 'ABC' AND 'DEF' are copied from FILE1 to FILE2.

POWERS OF 2

2^n	n
256	8
512	9
1 024	10
2 048	11
4 096	12
8 192	13
16 384	14
32 768	15
65 536	16
131 072	17
262 144	18
524 288	19
1 048 576	20
2 097 152	21
4 194 304	22
8 388 608	23
16 777 216	24

$2^0 = 16^0$
$2^4 = 16^1$
$2^8 = 16^2$
$2^{12} = 16^3$
$2^{16} = 16^4$
$2^{20} = 16^5$
$2^{24} = 16^6$
$2^{28} = 16^7$
$2^{32} = 16^8$
$2^{36} = 16^9$
$2^{40} = 16^{10}$
$2^{44} = 16^{11}$
$2^{48} = 16^{12}$
$2^{52} = 16^{13}$
$2^{56} = 16^{14}$
$2^{60} = 16^{15}$

POWERS OF 16

16^n	n
1	0
16	1
256	2
4 096	3
65 536	4
1 048 576	5
16 777 216	6
268 435 456	7
4 294 967 296	8
68 719 476 736	9
1 099 511 627 776	10
17 592 186 044 416	11
281 474 976 710 656	12
4 503 599 627 370 496	13
72 057 594 037 927 936	14
1 152 921 504 606 846 976	15

A	Move data from input to output buffers
B	Back up input pointer
BO	Back up output pointer
C	Specifies comment
D	Output from either input buffer to terminal
F	Move input pointer forward
G or GO	Unconditionally transfers control
H	Move text string to either output buffer
IF	Conditionally execute specified command
IH	Move text string to either input buffer
IP	Terminal Input to either input buffer
IS	Terminal Input to secondary input buffer
IT	Tape Label Input to primary input buffer
O	Output text string to terminal
P	Execute output buffer
PP	As in P, displays Output Buffers
PW	As in PP, waits for user response
PH	As in P, suppresses terminal output
PX	As in P, returns to TCL after processing
RI	Clear (resets) input buffer
RO	Clear (resets) output buffer
S	Positions input pointer
SP	Select primary input buffer
SS	Select secondary input buffer
STON	Select secondary output buffer (stack)
STOFF	Select primary output buffer
T f{,..}	Terminal Cursor Control output: f is B=bell, C=clear,(x,y), "text," also see **Cursor Functions** under BASIC INTRINSIC FUNCTIONS
U	Exit to user-defined subroutine
X	Exit back to TCL level, or calling PROC
+	Add number to parameter/input buffer
-	Subtract number from parameter/input buffer
()	Link to another PROC
[ ]{n}	Local or other PROC Subroutine, to label n

```
SET-BAUD 1200,0
   or    | |
SET-PORT a b
```

a	Indicates baud rate
b	Specifies port number

SELECTIVE RESTORES FROM TAPE

To selectively restore files, or selected items within a file, from a file save tape.

> SEL-RESTORE file-name (item list; or *) { (options) }

OPTIONS:

O — Overlay existing items on the file.

A — The tape is already positioned within the proper account. The "ACCOUNT NAME" prompt will not occur.

N — The file to be restored will be identified by its file number from a current stats report. The prompt "FILE # ?" will appear.

I — Inhibit item-id display during restore.

C — When the 'N' option is used, this causes every item before the next end of file to be a candidate for restore. This allows data to be restored in the event a D Pointer has been damaged on tape.

It may be necessary to position the tape beyond the inital end-of-file marker and label or header record with a T-READ OR T-FWD 1.

ITEM AND GROUP COMMANDS

> ITEM file-name item-id [(options)]

Locates and displays the Base FID of the group containing the specified item -id. Item-ids of all other items within the same group are also displayed.

> GROUP file-name [(options)]

OPTIONS for ITEM and GROUP:

P — Output to Printer
S — Suppress Item List

DUMP COMMAND

> DUMP X 1-{ — X 2 }-{ (Options) }

X = FID location to be dumped. FID addresses may be located by either the ITEM or GROUP command.

To specify hex location, precede location with a period (.). If no period precedes location, decimal is assumed.

OPTIONS:

C — Core dump
G — Entire Group is dumped
L — Links only are dumped (no data)
N — No end of page wait, non stop
P — Printer destination
U — Backward links utilized (default is forward)

EXAMPLE: DUMP .2348 (GX)

Will cause the Group at hex location 2348 to be displayed in both decimal and hex for examination.

1985 PICK OPERATING SYSTEM TRAINING CLASSES

	TECHNICAL CLASSES						NON TECHNICAL CLASSES			
	POS 104	P/B 204	ADV 304	GFE 101	COR 404	ASY 303	ORI 302	WP 301	OPR	PICK EXPO
APR	22-25	—	15-18	19	—	—	8-10	—	26	—
MAY	—	6-9	—	10	2-3	13-16	—	20-21	—	—
JUN	3-6	17-20	10-13	14	—	—	26-28	—	7	—
JUL	22-25	29-8/1	—	—	—	—	—	—	26	—
AUG	—	—	19-22	23	—	—	—	—	—	—
SEP	23-26	—	30-10/4	—	19-20	—	—	—	27	9-10
OCT	21-24	28-31	—	—	—	—	—	—	25	—
NOV	—	—	18-21	1	—	—	—	—	—	—
DEC	16-19	—	—	20	6-7	—	—	—	—	—

Class Codes: POS-PICK Op. Sys. Concepts; P/B-Basic Programming; ADV-Advanced Programming Concepts; GFE-GFE Handling & Sys. Maint.; COR-Advanced Correlative/conversions; ASY-Internal PICK Prog. Language; ORI-Systems Operations PICK EXPO — PICK USERS SHOW (San Diego, CA).

Location: Laguna Hills, California

Instructors: Jonathan E. Sisk (J.E.S. & Assoc.); Steve Lambert (Laguna Software & Consulting); John Timmons (Data Cache, Inc.) and staff.

Information: Call TMS, Inc. (714) 855-4442. Company discounts available for multiple students/class.