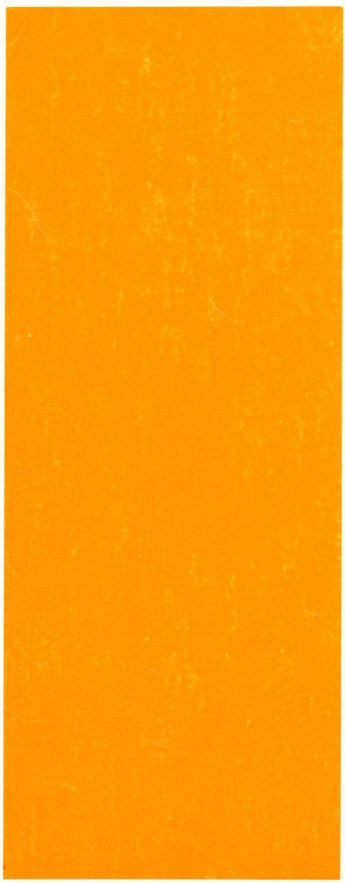
Honeywell

TIME-SHARING SYSTEM POCKET GUIDE

SERIES 600/6000 GCOS

SOFTWARE



Honeywell

TIME-SHARING SYSTEM POCKET GUIDE

SERIES 600/6000 GCOS

SUBJECT:

Brief Reference Information for Time-Sharing System Users.

SOFTWARE SUPPORTED:

Series 600 Software Release 8
Series 6000 Software Release F

DATE:

October 1974

ORDER NUMBER:

BS12, Rev. 2

User's Name_____

TSS Telephone Numbers

_____

_____

PREFACE

This guide is based on material extracted from the following references.

1. Time-Sharing BASIC, BR36
2. Time-Sharing System: System-Programmer's Reference, BR39
3. Time-Sharing System General Information Manual, BS01
4. Time-Sharing System Terminal/Batch Interface Facility, BR99
5. Time-Sharing Text EDITOR, BR40
6. FORTRAN, BJ67
7. ALGOL, BS11
8. JOVIAL, BS06

File No.: 1613, 1713

CONTENTS

NOTATION

[ ]	Enclosed items are optional.
{ }	Choice of one of items enclosed to be made.
Formats	TSS system typeouts/user responses in upper case.

Abbreviations

AFT	available file table
altname	alternate file name
ascfil	ASCII file
bcdfil	BCD file
␣	blank
catname	catalog name
filedescr	file descriptor
tempfile	temporary file
TSS	time-sharing system
UMC	user's master catalog
userid	user identification

underlined first letter implies permissible abbreviation; e.g., PRINT = P

TSS MAJOR SUBSYSTEMS

Subsystem	Function	Reference[1]
ALGOL	ALGOL language programming with I/O at TSS terminal.	7
BASIC	Algebraic language for problems involving small amounts of data.	1
CARDIN	Facility for building/submitting batch jobs at TSS terminal.	4
Text EDITOR and RUNOFF	Facility for building, maintaining, reformatting text files.	5
Series 600 FORTRAN	Algebraic language, permits subprogramming, chain overlays, peripheral I/O.	6
Series 6000 FORTRAN	Advanced version of Series 600 FORTRAN.	6
JOVIAL	JOVIAL language programming with I/O at TSS terminal.	8

TSS SERVICE AND UTILITY SUBSYSTEMS

Subsystem	Function	Reference[1]
ABC (Abacus)	Desk-calculator facility.	3
ACCESS	File system manipulation facility.	3
ASCASC	ASCII-to-ASCII conversion.	3
ASCBCD	ASCII-to-BCD conversion.	3
BCDASC	BCD-to-ASCII conversion.	3
BPRINT	Print ASCII file at computer site.	3

[1] See Preface

Subsystem	Function	Reference[1]
BPUNCH	Punch ASCII file at computer site.	3
Conversational I/O	Direct-access, I/O manipulation facility.	4
FDUMP	Remote-terminal, word-oriented, file inspection/maintenance facility.	4
HELP	Explanation of system error messages.	3
LIBED	Library editor for FORTRAN subroutines.	3
LODX	Subsystem checkout prior to integration into TSS.	2
Media Conversion Program	Generate TSS text file from card deck/vice versa.	3
RBUG	Conversational debug routine, for CARDIN.	4
RECOVERY	Recovery on the collector file.	3
SABT	Scan aborted file at TSS terminal, for TDS.	2
SCAN	Examine output of batch job from TSS terminal.	4
TDS	Terminal debug subroutine.	2

[1] See Preface

FILE DESIGNATIONS

Permanent Files

a. filedescr, having format of one of following:

1. filename
2. filename$password
3. userid/catname$password/filename$password

b. filedescr, permissions

c. filedescr "altname", permissions

Altname, 1-to-8 characters, enclosed in quotes

Asterisk may designate current file

Permissions

READ	EXECUTE
WRITE	APPEND

Null permissions imply READ and WRITE

FILE DESCRIPTIONS

File Names

filename	1-to-8 characters in length.
filename composition	Any combination of alphanumerics, periods, and minus signs, in any order.

File-Access Types

quick-access	Permanent file created by SAVE filename or PERM tempfile. Accessed by filename form of command. In case of data file, accessed by program reference. Has general READ permission.
quick-access with password	Permanent file created by SAVE filename$password. Accessed in same manner as quick-access file.

A quick-access file (with or without password) is only quick-access relative to file's creator.

nonquick-access	Permanent file not created by user or not emanating from UMC. Accessed by GET command or by ACCESS.

File Mode

ASCII	Linked file for variable-length records in ASCII.
binary	Linked file of variable-length records in binary.
random	Random file of fixed-length records in binary.

TSS COMMAND LANGUAGE

TSS Commands (See Reference 4 for complete description.)

Command	Operand	Function
ABC	expression	Call Abacus, evaluate given expression.
ACCESS		Call ACCESS, interface between TSS and file system.
APRINT	ascfil	Print ASCII file on printer equipped with an ASCII print train.
ASCASC	filedescr$_1$; filedescr$_2$	Convert time-sharing ASCII file to standard system format ASCII file.
ASCBCD	ascfil;bcdfil	Convert ASCII file to BCD file.
AUTOMATIC	m, n	Begin automatic generation of line numbers, start with m, incrementing by n.
AUTOX	m, n	Same as AUTO, standard terminating blank (i.e., mmmmb) not supplied.
BCDASC	bcdfil;ascfil	Convert BCD file to ASCII file.

Command	Operand	Function
BPUNCH	ascfil	Contents of file specified converted to BCD and punched.
BPRINT	ascfil	Contents of TSS file specified converted to BCD and printed.
BYE		Log off TSS system.
CATALOG	filename	List of user's catalog and file names, or only the attributes of named file.
CATALOG	/$catalog_1$/.../ $catalog_n$	List all catalog and file names which emanate from last specified catalog.
CATALOG	/$catalog_1$/.../ $catalog_n$*	List attributes of last specified catalog.
CATALOG	#CMD	List catalog and file names from the command library (CMDLIB).
CATALOG	#LIB	List all file names in library.
CATALOG	,x(date,n,R), FIRST/name/	Limit list of catalog and file names to optional date, number, name, and in reverse (R.) order.
DELETE	a,b,c-d,e,f-g	Delete, from current file, indicated lines.
DELETE	;*	Delete, from current file, all lines.
DONE		Return system control to next-higher level.
EDIT		Call EDITOR subsystem, edit the current file.
ERASE	$filedescr_1$...; $filedescr_n$	Erase file space associated with specified file(s), do not release file(s).
FDUMP		Call FDUMP to dump/correct file.
GET	$filedescr_1$;...; $filedescr_n$	Specified file(s) accessed and its filedescr placed in AFT.
HELP		Call HELP for error message explanation.

Command	Operand	Function
HOLD		Suppress warning or information message until subsequent SEND command given.
JABT	snumb	Batch-processing job specified aborted, X1 abort-code assigned.
JDAC	slave program name	Establish direct access with slave program.
JOUT	snumb	Permit manipulation of batch job.
JSTS	snumb	Current batch-processing status of job specified printed.
LEADER	title	Punch a paper tape leader with a title in bold, block letters followed by a listing of the current file.
LENGTH	filedescr	Report content length of current file, permanent file specified by filedescr.
LIB	filename	Named library file copied to current file.
LINELENGTHn		Set the length of an input line from a terminal to the value of n (80 to 160).
LIST	i,j	List from current file, or only lines numbered i and j.
LIST	i-j	List all the lines of the current file with line numbers i thru j.
LIST	filedescr(i,j)	List specified permanent file, or lines i and j thereof.
LIST	file list	Concatenate and list specified permanent files, and/or segments.
LIST	99999999	List the highest-numbered line of the current file.

Command	Operand	Function
LISTL		List the last line of the current file.
LISTH	(any LIST operand)	List file(s), print a header giving time and date.
LISTEnnn	(any LIST operand)	List file(s), "break" each line at character position nnn, continue with any remainder after carriage-return and line feed.
LISTI	n	List from current file each nth line.
LUCID (or #LUCID in EDITOR)		Prepare subsystem for non-ASCII paper tape input to follow.
NEW		Reinitialize current file (to begin new file).
NEWP	filedescr	Create named file(s) as quick-access.
NEWP#	filedescr	Create named file(s) as quick-access; remain user's current file(s) until log-off.
NEWUSER	[account number]	Log current user off terminal, do not disconnect; reissue log-on sequence.
NOPARITY		Send 8-bit parity independent code.
OLD	filedescr (i, j)	Copy specified file, or lines i through j onto current file.
OLD	file list	Copy specified files, and/or file segments, onto current file. Concatenate if semicolons used; merge (on line numbers) if colons used. (Resequencing not automatic.)
OLD	filedescr	Copy specified file(s) onto current file(s); remain user's current file(s) until form of OLD or NEW command given.

Command	Operand	Function
OLDP	filedescr	Make specified file current file (i.e., not a copy).
OLDP#	filedescr	Make specified file current file, specified file remains current file for any subsequent OLD or NEW commands (i.e., original contents could be overwritten or erased).
PARITY		Negate NOPARITY command.
PERM	tempfile; filedescr	Permanent file specified by filedescr created and/or accessed, temporary file tempfile copied onto it.
PRINT	file list	Reformat and print, at terminal, current file(s) and/or segment(s) specified. (CARDIN only.)
PRINT	i, k-m, ...	Print lines i and k thru m.
PURGE	$filedescr_1$;...; $filedescr_n$	Delete specified permanent file(s) from file system. (Release and overwrite file space.)
RECOVER (or #RECOVER in EDITOR)	filename	Provide recovery capabilities on collector file.
RELEASE	$filedescr_1$;...; $filedescr_n$	Release specified permanent file(s) from file system. (Does not overwrite file space.)
REMOVE	$filename_1$;...; $filename_n$	Remove the specified file(s) from AFT.
REMOVE	CLEARFILES (PERMFILES or TEMPFILES)	Remove all files from AFT, including current file.
RESAVE	$filedescr_1$;...; $filedescr_n$	Copy contents of current file on previous existing permanent file(s) specified.

Command	Operand	Function
RESEQUENCE	m, n	Resequence numbering of current file in order, begin with m and increment by n.
RESEX	m, n	Insert numbers at beginning of each line in current file. If first character of existing line is numeric, blank inserted following generated line number.
RESE#	m, n	Insert numbers at beginning of each line in current file. If first character of existing line is numeric, pound sign following generated line number.
ROLLBACK (or # ROLLBACK in EDITOR)	filename	Call recovery file that has data.
RUN		Execute selected system, taking source input from current file.
RUN	file list	Execute selected system, taking source (or object) input from specified permanent file(s) and/or file segment(s). (Not for BASIC)
RUN	(options)	Compile and/or execute according to the operand specifications.
RUNH		Execute selected subsystem and print a header (date and time) at the top of program execution report.
SAVE	filedescr$_1$;...; filedescr$_n$	Copy contents of current file on permanent file(s) specified.
SCAN	filedescr	Scan batch-output file specified.

Command	Operand	Function
SEND		Cancel effect of previous HOLD command, and cause last message withheld to appear at terminal.
STATUS		Print user's current-usage status: processor-time, file and terminal I/O usage, and list of open files.
STATUS	FILES	List only names of user's open files.
SYSTEM	subsystem name	Exit from current subsystem and call named subsystem, or, if no name is given, return control to subsystem-selection level.
TAPE (or #TAPE in EDITOR)		Prepare subsystem for paper tape input to follow immediately.

TSS Commands Applicability by Subsystem

Command	Subsystem: BASIC	ALGOL/JOVIAL/FORTRAN	CARDIN	EDITOR
ABC	Yes	Yes	Yes	No
ACCESS	Yes	Yes	Yes	Yes[1]
ASCBCD	No	No	Yes	Yes[1]
ASCASC	No	Yes	Yes	Yes[1]
• AUTOMATIC	Yes	Yes	Yes	No
BCDASC	No	No	Yes	Yes[1]
BPRINT	No	No	Yes	Yes[1]
BPUNCH	No	No	Yes	Yes[1]
BYE	Yes	Yes	Yes	Yes[1]
CATALOG	Yes	Yes	Yes	Yes[1]
DELETE	Yes	Yes	Yes	No
• DONE	Yes	Yes	Yes	Yes[1]
EDIT	Yes	Yes	Yes	No
ERASE	Yes	Yes	Yes	Yes[1]
FDUMP	No	No	Yes	Yes[1]
GET	Yes	Yes	Yes	Yes[1]
HELP	Yes	Yes	Yes	Yes[1]
HOLD	Yes	Yes	Yes	Yes[1]
JABT	No	No	Yes	Yes[1]
JDAC	No	No	Yes	Yes[1]
JOUT	No	Yes	Yes	Yes[1]
JSTS	Yes	Yes	Yes	Yes[1]
LEADER	Yes	Yes	Yes	Yes[1]
LENGTH	Yes	Yes	Yes	Yes[1]
• LIB	Yes	Yes	Yes	No
LINELENGTH	Yes	Yes	Yes	Yes[2]
LIST	Yes	Yes	Yes	Yes[1]
• LUCID	Yes	Yes	Yes	No
• #LUCID	No	No	No	Yes
• NEW	Yes	Yes	Yes	Yes[1]
NEWP	Yes	Yes	Yes	Yes[1]
NEWP#	Yes	Yes	Yes	Yes[1]
NEWUSER	Yes	Yes	Yes	Yes[1]
NOPARITY	Yes	Yes	Yes	Yes[1]
• OLD	Yes	Yes	Yes	Yes
OLDP	Yes	Yes	Yes	Yes
OLDP#	Yes	Yes	Yes	Yes
PARITY	Yes	Yes	Yes	Yes
PERM	No	Yes	No	Yes
PRINT	No	No	Yes	No
PURGE	Yes	Yes	Yes	Yes[1]
RECOVER	Yes	Yes	Yes	Yes[1]
• #RECOVER	No	No	No	Yes
RELEASE	Yes	Yes	Yes	Yes[1]
REMOVE	Yes	Yes	Yes	Yes[1]
RESAVE	Yes	Yes	Yes	Yes[1]

[1] "command" in direct-mode
[2] EDITOR recognizes only LINE n in the direct-mode.
• not applicable at subsystem-selection level

TSS Commands Applicability by Subsystem (cont)

Command	Subsystem: BASIC	ALGOL/JOVIAL/FORTRAN	CARDIN	EDITOR
•RESEQUENCE	Yes	Yes	Yes	No
•ROLLBACK	Yes	Yes	Yes	No
•#ROLLBACK	No	No	No	Yes
•RUN	Yes	Yes	Yes	No
•SAVE	Yes	Yes	Yes	Yes[1]
SCAN	No	No	Yes	Yes[1]
SEND	Yes	Yes	Yes	Yes[1]
STATUS	Yes	Yes	Yes	Yes[1]
•SYSTEM	Yes	Yes	Yes	No
•TAPE	Yes	Yes	Yes	No
•#TAPE	No	No	No	Yes

[1] "command" in direct-mode
•not applicable at subsystem-selection level

TSS TERMINAL OPERATION

Log-On

Activate terminal, dial TSS. Upon connection, TSS issues log-on message:

HIS SERIES 6000 ON [date] AT
[time] CHANNEL [no.]

Series of requests are made to which the user responds:

USER ID - userid [;account number]

PASSWORD

xxxxxxxxxxx (user's password typed over mask)

SYSTEM? { BASIC
FORTRAN
CARDIN
EDITOR
TFORTRAN
ALGOL
JOVIAL }

OLD OR NEW - { NEW, NEWP, NEWP#, OLD, OLDP, OLDP#, LIB, SAME }

READY

ENTER (for EDITOR only)

* (build-mode; user's input begins)

Input Procedures

blank spaces — Allowed, except within line numbers.

line numbers — Required in build-mode of BASIC, 600 FORTRAN, 6000 FORTRAN, ALGOL, JOVIAL, and CARDIN; optionally preceded by one or more blanks, always terminated by non-numeric character (may be a blank).

error corrections

teleprinter terminals

character(s)	function
@ (commercial "at" sign)	delete character
CTRL plus X	delete line

other terminals

character(s)	function
°(degree)	delete character
+ plus ATTN (for 2741)	delete line
+ plus INT (for DATEL)	delete line
+ plus carriage return	delete line

line-numbered file corrections or modifications

replacement: a numbered line replaces any identically numbered line previously typed or already contained on current file; i.e., last-entered line numbered nnn will be only line numbered nnn in the file.

deletion: a "line" consisting of a line number only, (i.e., nnn), causes deletion of any identically numbered line previously typed or already contained on the current file.

insertion: a line with a line-number value that falls between line-number values of two pre-existing lines inserted in file between those lines.

Interruption of Output Process — Teletypewriter terminals - BREAK

Typewriter-like terminals - ATTN or INT

Terminate or Log-Off

terminate session without terminal disconnect	Give NEWUSER command; new log-on sequence initiated.
exit from selected subsystem	Give DONE command; select new subsystem.
log-off (terminate session with TSS)	Give BYE command; report of user's TSS charges given, terminal disconnected.
	For subsystem without build-mode give DONE command to achieve subsystem selection level, then give BYE command.

PAPER TAPE INPUT

(See Reference 4 for complete details)

tape preparation — Maximum of 160 characters per line. Each line of data followed by:

carriage return
line feed
RUBOUT (one or more)

limits — Maximum of 2 links permitted, 6 links with LUCID.

input — In build-mode user specifies command (or LUCID for non-ASCII input)

input termination — Input terminated when:

XOFF encountered
one-second pause encountered with LUCID

The following symbol conventions apply:

n = any number	t = any single-letter variable
m = any number	l_n = any line-number
v = any variable	e_r = any relational expression
e = any expression	f = any filename
	fd = any file designator

BASIC-Language Statements

Statement Form	Function
CALL f_1	Call file f_i for use as subroutine.
CHANGE v_1 TO $v_1$$	Convert string characters to numerical code/vice versa.
CHAIN f_1	Chain file f_i to current file.
DATA $n_1, n_2, \ldots$	Specify numeric or alphanumeric values for variables in corresponding READ statement.
DEF FNX (v) = e	Define function repeatedly used in program.
DEF FNX FNEND	Terminate definition of multiple-line function.
DIM t(n), . . .	Specify maximum length (dimension) of list t
DIM t(m, n), . . .	Specify maximum dimensions (row, column) of table or matrix t.
END	Define end of program. (Optional final program statement)
FOR v=e_1 TO e_2 [STEP e_3]	Set up an execution loop between the FOR and subsequent NEXT statement. Loop is broken when value of v (which starts at e_1) exceeds e_2. Variable v is incremented each time through either by the step-value e_3 (optional), or by 1 (by default).

Statement Form	Function
GOSUB l_n	Transfer program control to routine beginning at line-number l_n. Control returns to the statement following the GOSUB. (Routine must terminate with a RETURN statement.)
GOTO l_n	Transfer program control unconditionally to statement numbered l_n.
IF e_r {THEN / GOTO} l_n	Conditionally transfer program control to statement numbered l_n if relational expression e_r is true for current values of variables specified therein.

Relational expressions (e_r) have the form [e_1 r e_2] when r is one of the following relational operators:

Operator	Meaning	Example
=	is equal to	A = B
<	is less than	A < B
<= or = <	is less than or equal to	A< =B or A =<B
>	is greater than	A > B
>= or = >	is greater than or equal to	A > =B or A =>B
< > or > <	is not equal to	A< >B or A> <B

INPUT v_1, v_2, . . .	Cause program execution to pause and a request for numerical input (?) issued. The n values entered at terminal are assigned in sequence to specified variables $v_i, \ldots, v_n$.
INPUT $v_1$$, $v_2$$, . . .	Same as above, except that $v_i$$, . . . , $v_n$$ are string variables.
LET v = e	Evaluate expressions e (if necessary), assign that value to the variable v. (Exp. e may contain v; e.g., LET A=A+1.)
LET v$ = "string"	Assign "string" to string variable v$.

Statement Form	Function
LET v$ = $v_1$$	Assign contents of string $v_1$$ to V$.
LET $v_1 = v_2 = v_3 = \ldots$	Make multiple variable replacement.
MAT	(MAT is special prefix for matrix-type statements; see description below.)
NEXT v	Terminate program loop headed by a FOR v=... statement. When loop is broken sequential execution of statements below NEXT ensues. The v's appearing in a FOR... TO/NEXT pair must be identical.
ON e {THEN / GOTO} l_n	Conditionally transfer processing sequence to designated statements.
(1) PRINT e_1, e_2,...	Evaluate and print values of e_1 (1) in standard zones, or (2) in compacted zones. (3) same as forms (1) and (2) except that $e_1$$ represents a string, (4) Print text verbatim. Forms (5) and (6) permit precise format control: (5) Print [e_1], tab to position [e_2] + 1, print [e_3], etc.; (6) Print [e_1], space right [e_2] positions, print [e_3], etc. Terminal comma or semicolon suppresses auto. carriage return. All forms may be combined.
(2) PRINT e_1; e_2;...	
(3) PRINT $e_1$$	
(4) PRINT "text",...	
(5) PRINT e_1;TAB(e_2);e_3,...	
(6) PRINT e_1;SPC(e_2);e_3,...	
PRINT USING l_n, list	Print formatted line; list is comma-separated arguments.
READ v_1, v_2,...	Read values from associated DATA statement, assign them in sequence to the specified variables $v_i, \ldots, v_n$.
READ $v_1$$, $v_2$$,...	Same as above, except that $v_i$$,..., $v_n$$ are string variables.

Statement Form	Function
REM any text	Supply remarks, or comments, for program listing.
RESTORE	Restore, or reinitialize, previously used blocks of numeric and string data from DATA statements.
RESTORE*	Restore, or reinitialize, previously used blocks of numeric data from DATA statements.
RESTORE$	Restore, or reinitialize, previously used blocks of string data from DATA statements.
RETURN	Terminate subroutine (called with a GOSUB); return control to the statement below the GOSUB.
STOP	Stop execution of program, wherever encountered in the program.
TRACE {ON / OFF}	Print line numbers of statements between TRACE ON, TRACE OFF statements.

Matrix Statements

MAT READ t_1, t_2, t_3...	Read matrices t_i (previously dimensioned). Data read in row-sequence from DATA statements.
MAT PRINT t_1, t_2, t_3...	Print matrices t_i, (Semicolon option may be used; see PRINT.)
MAT $t_1 = t_2 + t_3$	Add matrices t_2 and t_3, store result in t_1.
MAT $t_1 = t_2 - t_3$	Subtract matrix t_3 from t_2, store result in t_1.
MAT $t_1 = t_2 * t_3$	Multiply matrix t_2 by t_3, store result in t_1. (Dimensions must comply with rules of matrix-multiply.)

Statement Form	Function
MAT t_1 = t_2 * (e) MAT t_1 = (e) * t_2	Multiply matrix t_2 by value of expression e, store result in t_1.
MAT t_1 = INV (t_2)	Invert matrix t_2, store result in t_1.
MAT t_1 = TRN (t_2)	Transpose (interchange rows and columns) matrix t_2, store result in t_1.
MAT t = CON [(m, n)]	Set all elements of matrix t to 1 (optionally, redimension to m rows and n columns).
MAT t = ZER [(m, n)]	Set all elements of matrix t to 0 (optionally, redimension to m rows and n columns).
MAT t = IDN [(n, n)]	Set diagonal elements of matrix t to 1, yielding an identity matrix (optionally, redimension to square matrix of dimensions n, n).

Data File Statements

FILES f_1; f_2;...	Reference file(s) f_i prior to use in program.
APPEND #fd	Add data to data file designated.
BACKSPACE #fd	Position pointer for data file designated backward one delimiter.
DELIMIT #fd, {(char.) / (abbrev.)}	Use delimiter, other than comma, in data file designated.
FILE #fd, "f_1"	Replace data file fd with data file f_i.
IF END #fd {THEN / GOTO} l_n	Test for end of data in data file designated.
IF MORE #fd {THEN / GOTO} l_n	Test for remaining data or available space on data file designated.

Statement Form	Function
INPUT #fd, input list	Read data from data file designated into input list, treating line numbers as data items.
MARGIN #fd, e	Specify right-most character position of data file designated.
MAT READ #fd, matrix input list	Read data from data file designated into matrix input list.
MAT WRITE #fd, matrix output list	Write matrices specified in matrix output list to designated data file.
PRINT #fd, output list	Create data file from elements of designated data file, new data file to contain no new line numbers or delimiters.
PRINT #fd, USING statement number, output list	Format data written to a data file.
READ #fd, input list	Read data from data file designated into input list.
RESTORE #fd	Position pointer for data file designated to beginning of file, permit file to be read.
SCRATCH #fd	Place data file designated in write mode.
WRITE #fd, output list	Create data file from elements of designated data file.

Binary File Statements

Statement Form	Function
FILES f_1;f_2	Reference file(s) f_i prior to use in program.
APPEND:fd	Add data to binary file specified.

Statement Form	Function
BACKSPACE:fd	Position pointer for binary file specified backward one delimiter.
FILE:fd, "f_1"	Replace binary file fd with binary file f_1.
IF END:fd { THEN / GOTO } 1_n	Test for end of data in binary file designated.
IF MORE:fd { THEN / GOTO } 1_n	Test for remaining data or available space on binary file designated.
MAT READ:fd, matrix input list	Read data from binary file designated into matrix input list.
MAT WRITE:fd, matrix output list	Write matrices specified in matrix output list into designated binary file.
READ:fd, input list	Read data from binary file designated into input list.
RESTORE:fd	Position pointer for binary file designated to beginning of file, permit file to be read.
SCRATCH:fd	Place binary file designated in write mode.
SET:fd TO expression	Position pointer for binary file designated to desired point.
WRITE:fd, output list	Create binary file from elements of designated binary file.

Binary File Functions

LOC(fd)	Word pointer location
LOF(fd)	File length

Arithmetic Operators

+ addition	/ division
- subtraction/minus	** or ↑ exponentiation
* multiplication	

Mathematical Functions

SIN(X)	sine of X
COS(X)	cosine of X
TAN(X)	tangent of X
COT(X)	cotangent of X
ATN(X)	arctangent of X
EXP(X)	e to the power X
LOG(X)	natural logarithm of X
CLG(X)	common logarithm of X
ABS(X)	absolute value of X
SQR(X)	square root of X
INT(X)	truncation of X
RND(X)	a random number
SGN(X)	sign determination of X
DET(X)	determinant of last matrix inverted

Miscellaneous Functions

TIM(X)	Elapsed processor time
CLK$	Time of day
DAT$	Calendar date
NUM(X)	Count of matrix data elements
SST(X$, Y, Z)	Extract selected characters of string
TAB(X)	Character print position
SPC(X)	Space print position
LEN(X$)	Number of characters in string
LIN(X)	Last line number encountered
ASC(X)	Numeric value of character or abbreviation
STR$(N)	Expression to string conversion
VAL(S$)	String to expression conversion
TST(S$)	Nonzero output if string can be interpreted as a number
HPS(X)	Horizontal print position of next field, in current line, of file being written

SERIES 600 FORTRAN AND SERIES 6000 FORTRAN

Format of Input-Statement

n...nc	statement or continuation [; statement; ...; statement]
or	
n...nc comment	

where:

n...n	1-to-8 character numeric line number, value < 2^{18}
c	Single control character
	(b, &, *, C, #) following line number.
	b - next nonblank character begins new statement
	& - next nonblank character is continuation of prior statement
	* or C - information following is comment
	# - character following placed in column 1

Forms of TSS Commands Pertinent to Series 600 FORTRAN (TFORTRAN)

RUN

General form of the command:

RUN[H-n...n]filedescr(s)1=savefile (options) filedescr(s)2

where:

H generates time and date header

-n...n is processor time in seconds

filedescr(s)1 is permanent file(s) for compilation and/or execution; filedescrs are semicolon-separated

savefile compiles, executes, and saves current file

(options) are as follows:

NOGO - - file(s) is compiled, not executed

CHAIN - compilation saved in format required to be used as chain overlay file

ULIB - user's library searched in indicated order for specified routines

NOLINE - line numbers in core suppressed

filedescr(s)2 is file(s) in user's library; filedescrs are semicolon-separated

BYE For open temporary files, message is issued:

n TEMPORARY FILES CREATED

tempfile? carriage return	Ignore this file; pass to next file.
tempfile? NONE	Ignore all files; exit from system.
tempfile? SAVE filedescr	Save tempfile on filedescr.

Forms of TSS Commands Pertinent to Series 6000 FORTRAN (YFORTRAN or FORTRAN)

RUN

General form of the command:

RUN[H-n...n]fs=fh;fc (options)flib#fe

where:

H generates time and date header

-n...n is processor time in seconds

fs is filedescr(s) for source file

fh is filedescr for random file

fc is filedescr for sequential file

(options) refer to manual FORTRAN, BJ67

flib is filedescr(s) for random file(s) containing user libraries

fe is filedescr(s) for files required during execution

BYE For open temporary files, message is issued:

n TEMPORARY FILES CREATED

tempfile? carriage return	Ignore this file; pass to next file.
tempfile? NONE	Ignore all files; exit from system.
tempfile? SAVE filedescr	Save tempfile on filedescr.

Forms of RUNL Command

The RUNL command has the form:

RUNL C* file list = H* file (options) [ulib;] link list

C* file list is set of file descriptions for binary object image files

H* file is file descriptor of a random file

ulib is a file description of a random "RANLIB"

link list is a sequence of link phrases wherein each link phrase is used to specify the position at which segmentation is to take place. See also the manual FORTRAN, BJ67.

(options) are as follows:
ulib locates user library

CORE = nn set batch loader memory requirements

NAME = name provides name for the main link of the saved H* file

MAP = see the manual FORTRAN, BJ67

GO = allows user to enter execution link list; see the manual FORTRAN, BJ67.

Carriage Control in FORMAT statements

Spacing of printout is controlled by first character of format specification; first character may be supplied by 1Hc, where c is the character.

character	effect
0	carriage return, two line feeds
-	carriage return, three line feeds
+,1	carriage return, no line feed
&	no carriage return or line feed
any other character or blank	carriage return, line feed

ALGOL AND JOVIAL

Format of Input-Statement

n...nc	statement or continuation [; statement; ...; statement]
where	
n...n	1-to-8 character numeric line number, value $< 2^{18}$
c	Single control character
	(b or #) following line number
	b-next nonblank character begins content of new line
	# - character following placed in column 1

Forms of TSS Commands Pertinent to ALGOL and JOVIAL

(Same as Series 6000 FORTRAN)

EDITOR Commands

Following symbol conventions apply:

xxx	String of characters, any length, intended to match beginning of line(s) in file.
yyy and zzz	String of characters, any length, intended to match some portion of line(s) in file.
n	Numeric repetition-value, or asterisk(*). The *, as a value for n, specifies repetition to end of file.
/.../	Arbitrary choice of delimiter character (virgule, in this case). Any character not appearing in enclosed string may be used as delimiter.
+	AND function
-	OR function

General forms of EDITOR commands

(1) verb	Perform specified operation once, with reference to line currently pointed to.
(2) verb;n	Perform specified operation n times, with reference to line currently pointed to and next n-1 lines.
(3) verb:/xxx/	Perform specified operation once, with reference to next line beginning with string xxx.
(4) verb:/xxx/;n	Perform specified operation n times, with reference to next n occurrences of lines beginning with string xxx.
(5) verb:/xxx/,/yyy/	Perform specified operation once, with reference to block of lines starting with line beginning with string xxx through line beginning with string yyy.
(6) verb:/xxx/,/yyy/;n	Perform specified operation n times, with reference to block of lines starting with string xxx through line beginning with string yyy.
(7) verbS:/yyy/	Perform specified operation once, with reference to next occurrence of string yyy.
(8) verbS:/yyy/;n	Perform specified operation n times, with reference to next n occurrences of string yyy.
(9) verbS:/yyy/,/zzz/	Perform specified operation once, with reference to next occurrence of string yyy...zzz.
(10) verbS:/yyy/,/zzz/;n	Perform specified operation n times, with reference to next n occurrences of string yyy...zzz.
(11) verb:/yyy/+...+/zzz/	Perform specified operation on lines containing all strings listed.
(12) verb:/yyy/-...-/zzz/	Perform specified operation on lines containing any one of strings listed.

Editing Command Verbs

BACKUP -	Return search pointer to beginning of the file, or back up number of lines specified.

Notable form of the command:

BACKUP:n — "Back up search pointer n lines."

Applicable forms: 1, 2.

BUILD - Append text to file without return to SYSTEM ? level.

Applicable form: 1.

CASE - Search dual-case text.

Notable forms of the command:

CASE	"Search text, ignoring case."
CASE UPPER delimiter	"Search upper case text indicated by delimiters".
CASE LOWER delimiter	"Search lower case text indicated by delimiters".

Applicable forms: only as indicated here.

COPY - Copy line(s) implied by operand.

Notable form of the command:

COPY:n — "Copy next n lines, starting with current line."

Applicable forms:1-10.

CUT - Copy, remove line(s) implied by operand.

Notable form of the command:

CUT:n — "Copy, remove next n lines, starting with current lines."

Applicable forms: 1-10

DELETE - Delete line(s) or string(s) implied by operand.

Notable form of the command:

DELETE;n — "Delete next n lines, starting with current line."

Applicable forms: 1-10

FIND- Advance search pointer to (last) line implied by operand.

Notable form of the command:

FIND;n — "Advance search point n lines.

Applicable forms: 1 - 4, 7, 8, 11, 12

FS Forward space.

INSERT - Insert text following point(s) implied by operand. Insertion text is typed following request ENTER. Single line format available wherein request ENTER is not given.

Notable form of the command:

INSERT — "Insert a line after current line."

Applicable forms: 1, 3, 4, 7 - 10.

LINE - Return EDITOR to line mode.

Applicable form: 1.

NOVERIFY - Nullify VERIFY command.

Applicable form: 1.

PASTE - Insert text after next line(s) implied by operand.

Notable form of the command:

PASTE — "Insert text after current line."

Applicable forms: 1, 3, 4, 7 - 10

PRINT - Print line(s) implied by operand. Whole lines are printed, whether in line mode or in string mode.

Notable form of the command:

PRINT;* — If positioned at top-of-file: "print entire file." If not: "print remainder of file."

Applicable forms: 1 - 12.

REPLACE - Replace line(s) or string(s) of characters implied by operand. Replacement text is typed following request ENTER. Single line format available wherein request ENTER is not given.

Notable form of the command:

REPLACE;n "Replace next n lines, starting with current line."

Applicable forms: 1 - 10.

RUNOFF - Access RUNOFF subsystem while in EDITOR.

Applicable form: 1

STANDARD - Nullify CASE command.

Applicable form: 1.

STRING - Place EDITOR in string mode (until a subsequent LINE command is given).

Applicable form: 1.

VERIFY - Verify execution of EDITOR commands. The letter V appended to a verb (e.g., FINDV) is equivalent to VERIFY; NOVERIFY is not required.

RUNOFF Commands and Control Words

The following symbol conventions apply:

n - any number
f1, f2 - any file name

Command	Function
REFORM f1, f2, Count n	Format contents of f1, save it under name f2, insert relative line number of f1 in left margin of f2, left margin equals n spaces.

Command	Function
REFORM f1, , PRINT	Print formatted contents of f1, do not save file in formatted form.
REFORM f1, f2, PRINT	Format contents of f1, save it under name f2, print formatted text.
PRINT f2	Print formatted file saved by previous REFORM command.
SKIP n	Skip specified number of formatted pages. Printing starts on page n+1.
NOSTOP or NOSTOP n	Print continuously, to end of file or n pages without stopping at end of each page.
EDITOR	Access EDITOR subsystem while in RUNOFF.

Control Word	Function
.ALLCAP n	Print next n lines in upper case.
.BEGINPAGE n	End printing on this page and place following text on next page.
.BOLDFACE n	Overprint next n lines.
.BOTTOMMARGIN n	Specify size of bottom margin in terms of print lines.
.BREAK	Do not join following line to previous line (as in .FILL or .JUST).
.CENTER n	Center next n lines.
.COMMENT	Do not print following lines of text until another control word is found.
.DOUBLESPACE	Doublespace formatted text.

Control Word	Function
.FILL	Move words to shorten long lines and lengthen short lines.
.FOOTING x, n	Print n lines of foot line in location specified; x may be: C - centered R - right justified L - left justified A - right justified on odd pages, left justified on even pages.
.HEADER x, n	Specify location of page heading and number of lines. That number of following lines will be printed at the top of each page. See .FOOTING for identification of x.
.IGNORE x, x, ...	Cause symbols listed not to be used as text characters.
.INDENT n	Indent n spaces at beginning of all following lines. N is added to accumulated total.
.JUSTIFY	Insert spaces between words to justify the right margin.
.LEFTDENT n	Subtract n spaces from accumulated indent total for all following lines.
.LINELENGTH n	Specify length of line in terms of character spaces -- 10 per inch.
.LITERAL	Print following RUNOFF control word as part of text.
.MARGIN t, b, l, r	Set designated margins with numeric values.
.MULTISPACE n	N space formatted text.

Control Word	Function
.NODENT	Set accumulated total of indent n's to zero.
.NOFILL	Print all lines as they were entered.
.NOJUST	Do not justify right margin.
.NOTAB	Stop tabulation and return to previous format.
.PAGE x, y, n	Number pages, beginning with n. Print page numbers in location x, y. x may be: B = bottom, T = top x and/or y may be: C = center, L = left, R = right, A = alternating
.PAPERLENGTH n	Specify size of paper in terms of print lines -- 6 per inch.
.PARAGRAPH	After subparagraphing, return to previous line length.
.POINT n	Cause new page to be formatted.
.REFERENCE (x..x)	Print footnote within parentheses at bottom of the page.
.REPLACE x, x, ...	Listed symbols are replaced with space characters.
.SCOREUNDER n	Underscore next n lines.
.SINGLESPACE	Single space formatted text.
.SPACE n	Insert n line spaces before printing next lines.
.SUBFOOTING x, n	Print n lines of subfoot lines in location specified. See .FOOTING for identification of x.

Control Word	Function
.SUBHEADING x,n	Print n lines of subhead line in location specified. See .FOOTING for identification of x.
.SUBPARAGRAPH n	Shorten lines by n character spaces at beginning and end of each line.
.TABULAT n,...,n	Set tabs as specified by n.
.TOPMARGIN n	Specify size of top margin in terms of print lines.
.UNDENT n	Subtract n spaces from indent total for next line only.

TERMINAL/BATCH INTERFACE FACILITY

Categories of Facility

CARDIN	Submission of batch jobs through TSS.
supporting subsystems	SCAN (batch output scanning) FDUMP (file dumping/correction) JOUT (batch output manipulation)
batch-dimension features	ASCASC (file conversion) ASCBCD (file conversion) BCDASC (file conversion) BPUNCH (batch punch) BPRINT (batch print) RBUG (conversational debug routine) Conversational I/O

Question/Answer Sequences

Initiated by RUN or PRINT

CARD FORMAT, DISPOSITION ?

CARD FORMAT,... ? { ASIS | STRIP | MOVE | NORM }

...DISPOSITION ? { null | ,WAIT | ,TALK | ,URGC (xx) | ,JOUT | ,ROUT(xx) } (relevant for RUN only)

ASIS	Pass to batch as is.
STRIP	Strip line numbers. First nonnumeric character of line (if not a tab) goes in column 1.
MOVE	Move line numbers to columns 73-80. First nonnumeric character (if not a tab) goes in column 1.
NORM	Implies MOVE and that standard tab character (colon) and standard settings are used in generating file; i.e., :,8,16,32,73
null	Job is initiated.
WAIT	User notified of batch job completion. Terminal remains passive until job-termination message is received.
TALK	Terminal switched to direct-access with submitted program. Upon completion of conversational I/O, terminal remains passive until job-termination message is received.
URGC (xx)	Assign initial urgency.
JOUT	Save files for examination by JOUT.
ROUT (xx)	Direct output to station xx.

TAB CHARACTER AND SETTINGS ? { null (i.e., no tabs used) | tab-set;...;tab-set }

Asked only if prior response was not NORM.

Initiated by ASCBCD

LABELS? { ASIS | STRIP | MOVE | NORM | alpha (i, j)$_1$;...; alpha (i, j)$_n$ }

where:

alpha$_i$ is 1-to-5 character alphanumeric prefix

(i, j)$_i$ line-number range to be prefixed

TAB CHARACTER AND SETTINGS? (same as for RUN)

Asked only if prior response was not NORM.

Initiated by BPUNCH or BPRINT

LABELS? (same as for ASCBCD)

TAB CHARACTER AND SETTINGS? (same as for RUN)

$ IDENT (information identical to variable-field requirements of $ IDENT card)

Initiated by BCDASC

LINE NUMBERS? { null (i.e., no line numbers) | MOVE | AUTO | AUTO n, m }

Initiated by APRINT

$ IDENT?
RUNOFF FORMAT? (respond YES or NO)

Initiated by SCAN

FILE? filedescr or filedescr; n

where n denotes the nth report on the specified file

FORM? { GMAP | FORT | COBOL | LOAD | DUMP | USER }

If previous response was USER

CODE? line-code (1-5 characters)

EDIT? { YES (multiple-blank suppression) | NO (print blanks 'as is') }

? SCAN verb

SCAN Verbs

FIND/literal string/, n (EDITOR formats accepted)
PRINT n or ALL (EDITOR formats accepted)
LIST
BATCH
STATION CODE? { null (for central site) | ab (remote/batch code) }
$ IDENT? $ ident-information
$ USERID? $ userid-information
SPACE n
BACK n
LINE n
ERROR n
UNDEFINED
FLAG x or null
LOAD
CODE abcde or null
EDIT
DONE
BYE
REM text

SCAN Verbs Defined

FIND/literal string/;n	Position implied line counter to nth line containing literal string.
PRINT { n | ALL }	Print next n lines or all lines from current line.
LIST	Same as PRINT.
BATCH	Initiate a batch-world job.
SPACE n	Space line pointer ahead n lines.
BACK n	Space line pointer back n lines.
LINE n	Reposition line pointer to line n.
ERROR n	List next n errors.

UNDEFINED	List all undefined symbols.
FLAG x or null	List lines of GMAP assembly having error flags. Null error tag implies list of all flagged instructions.

Flags

U - undefined symbol

M - multidefined symbol

A - address illegal

X - Index illegal

R - relocation error

P - phase error

E - even error (col. 7)

C - conversion error

L - location field

O - operation

T - table overflow

LOAD	Print out abbreviated load map.
CODE abcde or null	Change line code. Null "turns off" line code.
EDIT	Return to EDIT? level.
DONE	Return to SYSTEM? level.
BYE	Terminate TSS session.
REM text	Produce "remarks" line.

Initiated by FDUMP

FILE NAME; TYPE - filedescr;t

where t is L (or blank) for a linked file, or R for a random file.

BLOCK TO BE READ - block serial number

FUNCTION ?

Sloc	Snap location Loc
$Sloc_1$-loc_2	Snap locations Loc_1 thru Loc_2
Sloc, n	Snap n locations, starting at loc
Ploc∅data	Patch functions
W	Rewrite function
C filedescr	Copy function
D	Done

Initiated by JOUT

FUNCTION ?

- ACTIVITY n
- DIRECT id or ONL
- EPRINT rc
- LIST
- PRINT rc
- REMOVE
- SCAN rc

where:

n is specified activity
id is remote site
ONL is central site
rc is report code obtained by LIST

Bypassing Question/Answer Sequences

Question/answer sequences, initiated by RUN, PRINT, and ASCBCD may be bypassed by providing first-line reformatting information.

(question) ? [line-number] ## first-response/second-response

where:

first-response and second-response is precisely what is required in question/answer sequence.

RBUG (Conversational Debug Routine)

General form of an RBUG instruction:

aloc (i, j, k)

where:

a is an operation name,

loc is a location or location-symbol,

i is the "at" or "from" address,

j is the "to" address, and

k is a step-value (increment)

subscripts j and k are optional where applicable.

RBUG Operation Names

A - ASCII snap

B - Breakpoint insertion

C - Complex snap

D - Double-precision snap

E - Erase breakpoint

F - Floating-point snap

H - Hollerith (BCl) snap

I - Integer (decimal) snap

L - Logical (T or F) snap

M - Modify register

Form: Mr$\emptyset$data$_8$

O - Octal snap

P - Patch location

Form: Ploc(i)$\emptyset$data$_8$

Q - Quit (Xl abort)

R - Run (i optional)

S - Starting-address (offset)

T - Terminate, normal

W - Where (effective address)

X - Register display

Form: Xr, or X for "all"

Conversational I/O

Job Control Cards:

$ DAC fc, d

where:

fc is remote-terminal file code, and
d is logical-unit-designator (normally blank)

$ USE .RTYP

Applicable Calls:

- OPEN - connect terminal, open file
- GET - read logical record
- PUT - write logical record
- CLOSE - close file, disconnect terminal

Calling sequences are standard, as described in FILE and RECORD CONTROL, BN85

ACCESS

ACCESS Functions

CREATE CATALOG	Create subcatalog.
CREATE FILE	Define file space, attributes for given file name. Does not bring file into AFT.
ACCESS FILE	Bring file into AFT.
DEACCESS FILE	Take file out of AFT.
MODIFY CATALOG	Modify name, password, and/or permissions associated with given catalog.
MODIFY FILE	Modify name, maximum size, password, and/or permissions associated with given file.
PURGE CATALOG	Delete catalog from system along with subordinate sub-catalogs, and files; released file space overwritten.
PURGE FILE	Delete file from system, overwriting released file space.

RELEASE CATALOG	Delete catalog from system along with subordinate sub-catalogs and files; released file space not overwritten.
RELEASE FILE	Delete file from system, without overwriting released file space.
LIST CATALOG	List names of catalogs and files emanating from catalog.
LIST SPECIFIC	List in detail description of catalog or file specified.

User Permissions

READ	EXECUTE
WRITE	MODIFY
APPEND	PURGE
EXCLUDE	CREATE
LOCK	RECOVER

Function Formats

CREATE CATALOG, CREATE FILE

FUNCTION ? { CC / CF }

CATALOG STRUCTURE TO WORKING LEVEL ?

userid/catname$password/... /catname$password

NEW CATALOG NAME ? catname (for CC only)

FILENAME, SIZE (LLINKS), MAX SIZE, MODE ? (for CF only)

filename, initial size (blocks) max. size (blocks)

PASSWORD ? password

GENERAL PERMISSIONS ? permission,...,permission

SPECIFIC PERMISSION?

permission,...,permission/userid/.../userid

null response returns system to NEW CATALOG ? level

MODIFY CATALOG, MODIFY FILE

FUNCTION ? { MC / MF }

CATALOG STRUCTURE TO WORKING LEVEL?

userid/catname$password/filename$password (latter for MF only)

NEW NAME ? new catname or new filename

NEW MAX. SIZE ? new max. size (for MF only)

NEW PASSWORD ? { new password / DELETE }

GENERAL PERMISSIONS ? { permission,...,permission / DELETE }

SPECIFIC PERMISSION ? { permission,...,permission/userid / DELETE/userid/.../userid }

Short-Form Format

FUNCTION ? function name, catalog/file string, option,..,option

where options are: password, permissions, size assignment, mode assignment

TSS MEDIA CONVERSION

TSS media conversion (TSCONV) will perform the following functions:

- INPUT - create standard format, TSS text file from cards.
- OUTPUT - create card deck from standard format, TSS text file.

INPUT { ASIS / MOVE / INSERT / ASCII / COMDK }

INPUT identifies control card requesting file-creating function and takes following mutually exclusive options:

ASIS, i, j	Text file generated from input cards, from columns specified by i to j. Standard columns (default option) for i to j are 1 to 80.
MOVE, i, j, m, n	Text file generated from input cards, from columns specified by i to j. Line numbers taken from columns specified by m to n. Standard columns for i to j are 1 to 72, and form m to n are 73 to 80.
INSERT, i, j, m, n	Text file generated from input cards and from columns specified by i to j. Line sequence-numbered, starting with m and incremented by n. Standard columns for i to j are 1 to 72. Standard values for both m and n are 10.
ASCII	Text file generated from input cards, using binary deck previously punched from this program.
COMDK, option	Text file generated from input cards consisting of a compressed source deck. Option used in conjunction with ASIS, MOVE, or INSERT. If ALTER's are to be made at time file is generated, a $ DATA 1*, , COPY and a $ ENDCOPY card must be employed.

OUTPUT { ASIS / MOVE / STRIP / ASCII }

OUTPUT identifies control card requesting card-deck producing function, and takes following mutually exclusive options:

ASIS, i, j	**Text file read, BCD card deck punched** in columns specified by i to j. Standard columns (default option) for i to j are 1 to 80.
MOVE, i, j, m, n, l	Text file read, BCD card deck punched, moving data to columns specified by i to j. Line numbers moved to columns specified by m to n, right-justified. L specifies label to be punched starting in column 73, left-justified. Standard columns for i to j are 1 to 72 and for m to n, 73 to 80.

STRIP, i, j	**Text file read, check deck punched,** stripping off line numbers, with data moved to the columns specified by i to j. Standard columns for i to j are 1 to 80.

Note

With above output options, data is converted from ASCII to BCD before punching.

ASCII	Text file read, binary deck containing file text punched. (See "Binary Card Format" below.)

Binary Card Format

Word 1	7/9 punch and number of data words (maximum = 21)
Word 2	Checksum
Word 3	Card number, starting at 0
Words 4-24	Text

Abort Codes

SE - A binary card is out of sequence. Card number is printed out.

CK - Checksum of card does not agree with computed checksum.

NB - First data card is not binary, but ASCII was specified on control card.

CP - No control card found (keyword may be misspelled).

Sample Deck Setup

```
$ SNUMB          XXXXX
$ IDENT          account number, name
$ USERID         name$password
$ PROGRAM        TSCONV
$ PRMFL          OT, R/W, L, userid/filename
INPUT, ASIS
   .
   .
(Data deck)
   .
   .
$ ENDJOB
***EOF
```

OCTAL/ASCII CONVERSION EQUIVALENTS

Octal No.	ASCII Char.	Octal No.	ASCII Char.	Octal No.	ASCII Char.	Octal No.	ASCII Char.
000	NULL	040	SP	100	@	140	GRA
001	SOH	041	EXP	101	A	141	a
002	STX	042	"	102	B	142	b
003	ETX	043	#	103	C	143	c
004	EOT	044	$	104	D	144	d
005	ENQ	045	%	105	E	145	e
006	ACK	046	&	106	F	146	f
007	BELL	047	'	107	G	147	g
010	BSP	050	(	110	H	150	h
011	HT	051	)	111	I	151	i
012	LF	052	*	112	J	152	j
013	VT	053	+	113	K	153	k
014	FFD	054	,	114	L	154	l
015	CR	055	-	115	M	155	m
016	SO	056	.	116	N	156	n
017	SI	057	/	117	O	157	o
020	DLE	060	0	120	P	160	p
021	DC1	061	1	121	Q	161	q
022	DC2	062	2	122	R	162	r
023	DC3	063	3	123	S	163	s
024	DC4	064	4	124	T	164	t
025	NAK	065	5	125	U	165	u
026	SYN	066	6	126	V	166	v
027	ETB	067	7	127	W	167	w
030	CAN	070	8	130	X	170	x
031	EM	071	9	131	Y	171	y
032	SUB	072	:	132	Z	172	z
033	ESC	073	;	133	LBK	173	LBR
034	FS	074	<	134	RSL	174	VTL
035	GS	075	=	135	RBK	175	RBR
036	RS	076	>	136	CFX	176	NOT
037	US	077	?	137	_	177	DEL

Honeywell

Honeywell Information Systems
In the U.S.A.: 200 Smith Street, MS 486, Waltham, Massachusetts 02154
In Canada: 2025 Sheppard Avenue East, Willowdale, Ontario M2J 1W5
In Mexico: Avenida Nuevo Leon 250, Mexico 11, D.F.

20802, 2578, Printed in U.S.A. BS12, Rev. 2