CONTROL DATA CORPORATION

BASIC VERSION 3 SUMMARY

Pub. No. 60482800A July 1978

CONTROL DATA® OPERATING SYSTEMS

NOS 1 for the CONTROL DATA® CYBER 170 Models 171, 172, 173, 174, and 175; CYBER 70 Models 71, 72, 73, and 74; and 6000 Series Computer Systems

NOS/BE 1 for the CDC CYBER 170 Series; CYBER 70 Models 71, 72, 73, and 74; and 6000 Series Computer Systems

LANGUAGE ELEMENTS

BASIC CHARACTER SET

Alphabetic:	A thru Z
Numeric:	0 thru 9
Special:	+ ,
	- .
	* "
	/ ^
	(<
	) >
	$?
	= ;
	: #
	' blank

Any character available to the operating system can be used in data and string constants.

VARIABLES

Numeric Variables

A numeric variable consists of a single alphabetic character or a single alphabetic character followed by a numeric character. Numeric variables are preset to zero before program execution.

String Variables

A string variable consists of a single alphabetic character followed by a dollar sign ($) or a single alphabetic character and a numeric character followed by a dollar sign.

Subscripted Variables

A numeric subscripted variable consists of a numeric variable followed by a subscript list bounded by parenthesis.

A string subscripted variable consists of a string variable followed by a subscripted list bounded by parenthesis.

CONSTANTS

Constant	Form
Integer	±n n Decimal digits Range $1.26501 \times 10^{322} \geq \lvert n \rvert \geq 3.13152 \times 10^{-294}$ Accurate to a maximum of 14 digits
Decimal	±n. ±.n ±n.n n Decimal digits Range $1.26501 \times 10^{322} \geq \lvert n \rvert \geq 3.13152 \times 10^{-294}$ Accurate to a maximum of 14 digits
Exponential	±nE±s ±n.nE±s ±.nE±s ±n.E±s n Decimal digits E±s Exponent s Base 10 scale factor Range $1.26501 \times 10^{322} \geq \lvert n \rvert \geq 3.13152 \times 10^{-294}$ Accurate to a maximum of 14 digits
String	"string" string String of alphanumeric characters. If the character " is to appear in the string, it must be specified by two consecutive " marks. A string can contain 0 to 131,070 6-bit characters or 0 to 65,535 12-bit escape code characters.

OPERATORS

Symbol	Meaning
Arithmetic Operators	
^ or **	Exponentiation.
/	Division.
*	Multiplication.
+	Addition; unary plus.
-	Subtraction; unary minus.
Relational Operators	
=	Equal to.
<>or><	Not equal to.
>	Greater than.
>= or =>	Greater than or equal to.
<	Less than.
<= or =<	Less than or equal to.
Logical Operators	
NOT	Logical negation.
AND	Logical multiplication or intersection.
OR	Logical addition or union.
String Operators	
+	String concatenation.

STATEMENT FORMS

[ ]	Enclosed elements are optional.
{ }	Only one element must be selected.
. . .	Repeat elements as needed.

Terms in lowercase represent words or symbols supplied by the programmer.

a	Alphabetic identifier	nc	Numeric constant
c	Numeric string constant	ne	Numeric expression, constant or variable
ch	Any character	r	Relational expression
e	Expression, constant, or variable	sc	String constant
esub	External subroutine name	se	String expression, constant, or variable
i	Integer	snv	Simple numeric variable
im	Format image	stm	Executable statement
lfn	File name	stv	String variable
ln	Line number	sv	Simple variable
m	Matrix (one or two dimensional array)	v	Variable identifier (numeric, string, or subscripted)
na	Numeric array variable		

APPEND #ne

$\text{BASE}\begin{Bmatrix}0\\1\end{Bmatrix}$

CALL esub [(e1,e2, . . . ,e20)]

$$\text{CHAIN}\begin{Bmatrix}\text{lfn}\left[\text{, file-param}\left[\text{, }\begin{Bmatrix}\text{NORMAL}\\\text{ASCII}\\\text{MODE=}\begin{Bmatrix}\text{NORMAL}\\\text{ASCII}\end{Bmatrix}\end{Bmatrix}\right]\right]\\\text{\#ne}\end{Bmatrix}$$

$\text{CHANGE}\begin{Bmatrix}\text{stv TO na}\\\text{na TO stv}\end{Bmatrix}$

CLOSE #ne

DATA c1, c2, . . .

$\text{DEF FNa}\begin{Bmatrix}\text{(sv1,sv2, . . . ,sv20)=ne}\\\text{(sv1,sv2, . . . ,sv20)}\end{Bmatrix}$

$\text{DEF FNa\$}\begin{Bmatrix}\text{(sv1,sv2, . . . ,sv20)=se}\\\text{(sv1,sv2, . . . ,sv20)}\end{Bmatrix}$

$\text{DELIMIT}\left[\begin{Bmatrix}\text{(ch1), . . . ,(ch3)}\\\text{\#ne,(ch1), . . . ,(ch3)}\end{Bmatrix}\right]$

DIM m1(i1, . . . ,i3),m2(i1, . . . ,i3), . . .

END

FILE #ne1=lfn1,#ne2=lfn2, . . .

FNEND

FOR snv = ne1 TO ne2 [STEP ne3]

GOSUB ln

GOTO ln

$\text{IF END \#ne}\begin{Bmatrix}\text{THEN}\\\text{GOTO}\end{Bmatrix}\text{ ln}$

$\text{IF r}\begin{Bmatrix}\text{GOTO ln1}\\\text{THEN}\begin{Bmatrix}\text{ln1}\\\text{stm1}\end{Bmatrix}\end{Bmatrix}\left[\text{ELSE}\begin{Bmatrix}\text{ln2}\\\text{stm2}\end{Bmatrix}\right]$

$\text{IF MORE \#ne}\begin{Bmatrix}\text{THEN}\\\text{GOTO}\end{Bmatrix}\text{ ln}$

INPUT [#ne,] v1,v2, . . .

JUMP ne

[LET] v1 [=v2=v3, . . .]=e

$\text{MARGIN}\begin{Bmatrix}\text{\#ne1,ne2}\\\text{ne}\end{Bmatrix}$

$$\text{MAT m1 = }\begin{Bmatrix}\text{m2}\\\text{m2 + m3}\\\text{m2 - m3}\\\text{m2 * m3}\\\text{(ne) * m2}\\\text{INV(m2)}\\\text{TRN(m2)}\\\text{ZER[(ne1 [,ne2])]}\\\text{CON [(ne1 [,ne2])]}\\\text{IDN[(ne1 [,ne2])]}\end{Bmatrix}$$

MAT READ [#ne,] m1,m2, . . .

$\text{MAT PRINT [\#ne]}\left[\text{USING}\begin{Bmatrix}\text{im,}\\\text{ln,}\end{Bmatrix}\right]$
$\quad\text{m1}\begin{Bmatrix},\\;\end{Bmatrix}\text{m2}\begin{Bmatrix},\\;\end{Bmatrix}\cdots\left[\begin{Bmatrix},\\;\end{Bmatrix}\right]$

MAT INPUT [#ne] m1,m2, . . .

MAT WRITE #ne,m1,m2, . .

NEXT svn

NODATA [#ne,] ln

ON ne GOSUB ln1, ln2, . . .

$\text{ON ne}\begin{Bmatrix}\text{GOTO}\\\text{THEN}\end{Bmatrix}\text{ ln1,ln2, . . .}$

$\text{ON ERROR}\left[\begin{Bmatrix}\text{GOTO ln}\\\text{THEN ln}\end{Bmatrix}\right]$

$\text{PRINT [\#ne]}\left[\text{USING}\begin{Bmatrix}\text{im,}\\\text{ln,}\end{Bmatrix}\right]$
$\quad\text{e1}\begin{Bmatrix},\\;\end{Bmatrix}\text{ e2}\begin{Bmatrix},\\;\end{Bmatrix}\cdots\left[\begin{Bmatrix},\\;\end{Bmatrix}\right]$

READ [#ne,] v1,v2, . . .

$$\text{REM}\left[\begin{Bmatrix}\text{LIST,}\begin{Bmatrix}\text{NONE}\\\text{ALL}\end{Bmatrix}\\\text{TRACE,}\begin{Bmatrix}\text{ALL}\\\text{PART}\\\text{NONE}\end{Bmatrix}\\\text{ch1ch2 . . .}\end{Bmatrix}\right]$$

RESTORE #ne

RETURN

SET #ne1,ne2

SETDIGITS ne

STOP

WRITE #ne,e1,e2, . . .

:im

FORMAT FIELD SPECIFICATION CHARACTERS

INTRINSIC FUNCTIONS

Legend:

nc	Numeric constant	se	String expression, constant, or variable
ne	Numeric expression, constant, or variable	x	Constant, variable, function, or numeric expression
abr	Abbreviation for ASCII character name		
ch	Character	[ ]	Optional
m	Matrix	. . .	Repeat as needed

Function	Form	Comment
Absolute value	ABS(ne)	

ASCII code	ASC(ch)
	ASC(abr)	

Arctangent	ATN(ne)	Range: $-\pi/2$ to $+\pi/2$
Character	CHR$(ne)	Range: $0 \geq ne \leq 127$
Time of day	CLK(ne)	Time in hours and fractions of an hour
Time of day	CLK$	Time as a string constant
One matrix	CON [(ne1 [,ne2])]	
Cosine	COS(ne)	Angle ne is in radians
Cotangent	COT(ne)	Angle ne is in radians
DATE	DAT$	
Determinant	DET	Determinant of most recently inverted matrix
Error statement line	ESL(ne)	
Error statement message	ESM(ne)	
Base e	EXP(ne)	e to power of ne
Identity matrix	IDN [(ne1 [,ne2])]	
Largest integer	INT(ne)	Largest ne
Matrix inversion	INV(m)	
Length of string	LEN(se)	
Base 10 log	LGE(ne)	ne>0
Current word position	LOC(ne)	
Length of file	LOF(ne)	Length in words
Natural log	LOG(ne)	ne>0
Maximum	MAX(ne1, . . .ne20)	
Minimum	MIN(ne1, . . .ne20)	
Next line number	NXL(x)	
Random number	RND(ne)	Number between 1 and 0

Round off	ROF(ne1)
	ROF(ne1, ne2)
		Rounds ne1 off to ne2 decimal places

Sign	SGN(ne)	1 if ne is positive 0 if ne is zero -1 if ne is negative
Sine	SIN(ne)	Angle ne is in radians
Square root	SQR(ne)	ne≥0
String	STR$(ne) STR$(ne,se)	Resultant string formatted according to se

Substring	SUBSTR(se,ne1,ne2)
	SUBSTR(se,ne1)
		n1 beginning character, n2 ending character

Tab	TAB(ne)	
Tangent	TAN(ne)	Angle ne is in radians
CPU elapsed time	TIM(ne)	
Matrix Transposition	TRN(m)	
User number	USR$	Returns user number under NOS; returns USERNUM under NOS/BE
Value	VAL(se)	String of numbers to numeric value
Zero matrix	ZER [(ne1 [,ne2])]	

FORMAT FIELD SPECIFICATION CHARACTERS

Character	Representation
#	Numeric character, alphabetic character, alphanumeric character
$	Currency sign; floating when more than one
*	Check protect; leading blanks replaced by *
<	Left-justify string; right truncate
>	Right-justify string; left truncate
^	Floating point indicator
+	Plus printed for positive values; minus for negative
-	Blank printed for positive values; minus for negative
,	Comma insertion
.	Decimal point insertion
( )	Negative values enclosed in parentheses; positive values in blanks
DB	DB inserted for negative value; two blanks for positive
CR	CR inserted for negative value; two blanks for positive

BASIC CONTROL STATEMENT

BASIC.
BASIC(parameter-list)

AS	omitted AS=0	Source program and data files contain non-ASCII characters.
	AS	Source program and data files contain ASCII characters.
B	omitted B=0	Do not produce relocatable binary.
	B	Write relocatable binary on file BIN.
	B=lfn	Write relocatable binary on file lfn.
BL	omitted	Suppress page ejects on output listing.
	BL	Do not suppress page ejects on output listing.
DB	omitted DB=0	Do not activate the debug and trace features.
	DB	Same as DB=B/DL.
	DB=B	Force binary generation and/or program execution regardless of compilation errors.
	DB=DL	Activate program tracing as controlled by REM TRACE debug lines.
	DB=TR	Trace all statements.
E	omitted	Write compile-time error diagnostics on the file specified by the L parameter. If L=0 then write diagnostics to OUTPUT.
	E	Write compile time error diagnostics on file ERRS
	E=lfn	Write compile time error diagnostics on file lfn.
EL	omitted EL=W	Write warning diagnostics and fatal compile-time diagnostics on the file specified by the E parameter.
	EL=F	Write only fatal compile-time diagnostics on the file specified by the E parameter.
GO	omitted	If no B parameter is specified, execute compiled program without loading. If B parameter is specified, do not execute compiled program.
	GO	Execute compiled program.
	GO=0	Do not execute compiled program.
I	omitted	Compile source program from file INPUT.
	I	Compile source program from file COMPILE.
	I=lfn	Compile source program from file lfn.
J	omitted J	Read data from default file INPUT.
	J=lfn	Read data from default file lfn.
	J=0	No default runtime data file.
K	omitted K	Write execution-time output on default output file OUTPUT.
	K=lfn	Write execution-time output on default output file lfn.
L	omitted	If batch job, write compile-time output on default output file OUTPUT. If interactive job, suppress compile-time output.
	L	Write compile-time output on file OUTPUT.
	L=lfn	Write compile-time output on file lfn.
	L=0	Suppress compile-time output.
LO	omitted LO LO=S	Write source listing on the file specified by the L parameter.
	LO=0	Write object and source listing on the file specified by the L parameter.
	LO=0/0	Write object listing only on the file specified by the L parameter.
	LO=0	Turn off all list options.
PD	omitted	Use the installation default print density on the files specified by the L and K parameters.
	PD=6	Use a print density of 6 lines/-inch on the files specified by the L and K parameters.
	PD PD=8	Use a print density of 8 lines/-inch on the files specified by the L and K parameters.
PS	omitted	If PD is not specified, use installation default page size for the file specified by the L parameter. If PD is specified, use PS=PD*(default PS)/(default PD).
	PS=n	Use a page size of n lines/-page; $4 \leq n \leq 32768$.