60482600

CONTROL DATA CORPORATION

SYMPL VERSION 1 INSTANT

DAVID E. LEE

CDC® OPERATING SYSTEMS:
NOS 1
NOS/BE 1
SCOPE 2

60482600

CONTROL DATA CORPORATION

SYMPL VERSION 1 INSTANT

CDC® OPERATING SYSTEMS NOS 1 NOS/BE 1 SCOPE 2

REVISION RECORD	
REVISION	DESCRIPTION
A	Original release at PSR level 472.
(5-23-78)	
RECEIVED	
05 MAR 1979	
DAVID E. LEE	
Publication No. 60482600	REVISION LETTERS I, O, Q AND X ARE NOT USED

Address comments concerning this manual to:

CONTROL DATA CORPORATION
Publications and Graphics Division
215 MOFFETT PARK DRIVE
SUNNYVALE, CALIFORNIA 94086

LIST OF EFFECTIVE PAGES

New features, as well as changes, deletions, and additions to information in this manual are indicated by bars in the margins or by a dot near the page number if the entire page is affected. A bar by the page number indicates pagination rather than content has changed.

Page	Revision
Cover	–
Title Page	–
ii	A
iii/iv	A
v/vi	A
vii	A
viii	A
1 thru 27	A
Cover	–

Page	Revision

PREFACE

This instant provides a convenient summary of the SYMPL Version 1.3 language as implemented under the control of the following 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

SCOPE 2 for the Control Data CYBER 170 Model 176; CYBER 70 Model 76; and 7600 Computer Systems

This instant is written for a programmer familiar with the SYMPL 1 language and the operating system under which the SYMPL 1 compiler is operating.

More detailed information can be found in the publications listed below.

Publication	Publication Number
SYMPL Version 1 Reference Manual	60496400
SYMPL Version 1 User's Guide	60499800

CDC manuals can be ordered from Control Data Literature and Distribution Services, 8001 East Bloomington Freeway, Minneapolis, MN 55420.

CONTENTS

Language Elements 1
SYMPL Character Set 1
Comments 1
Identifiers 1
Constants 2
Operators 3
Declarations 4
ARRAY Header Declaration 4
ARRAY ITEM Declaration 4
BASED ARRAY Declaration 5
COMMON Declaration 5
Function as Formal Parameter Declaration 5
ITEM Declaration 6
LABEL Declaration 6
Procedure as Formal Parameter Declaration 6
STATUS Declaration 6
STATUS SWITCH Declaration 7
SWITCH Declaration 7
XDEF Declaration 7
XREF Declaration 8
Executable Statements 9
Exchange Statement 9
FOR Statement 9
GOTO Statement 9
IF Statement 10
Labeled Statement 10
Replacement Statement 10
RETURN Statement 11
STOP Statement 11
TERM Statement 11
TEST Statement 11
Compiler Directives 12
$BEGIN Directive 12
CONTROL Directive 12
Attributes of Variables 12
Core Residence Selection 12
FOR Loop Control 13
Compilation Option Selection 13
Conditional Compilation 14
Listing Control 14
Weak External 14
DEF Directive 15
$END Directive 15
Program Structure 16
Main Program 16
Procedures 16
Functions 17
Alternative Entry Points 18

Execution Time Output Procedures 19
Procedure PRINT 19
Procedure PRINTFL 19
Procedure LIST 19
Procedure ENDL 19
Intrinsic Functions 20
ABS Function 20
B Function 20
C Function 20
LOC Function 21
P Function 21
SYMPL Control Statement 22
Reserved Words 26
Standard Character Sets 27

LANGUAGE ELEMENTS

SYMPL CHARACTER SET

Letters:	A through Z and $	
Digits:	0 through 9	
Marks:	+	(
	-	)
	*	[
	/	]
	=	<
	,	>
	.	"
	:	#
	;	blank

COMMENTS

#string#

string Arbitrary string of characters; semicolon and comment delimiter are not valid characters.

Comments can appear anywhere except within a comment or after the name in a DEF declaration.

IDENTIFIERS

An identifier is a string of 1 through 12 letters or digits beginning with a letter. Programmer-defined identifiers must not duplicate reserved words.

CONSTANTS

Constant	Form
Real	±n.n ±.n ±n. ±n.nE±s ±.nE±s ±n.E±s n Coefficient of decimal digits. E±s Exponent (D can be used instead of E). s Base 10 scale factor. Range 10^{-293} to 10^{+322} Accurate to approximately 15 decimal digits.
Decimal Integer	$\pm n_1 n_2 \ldots n_m$ n Decimal digit. Range $-(2^{47}-1)$ to $2^{47}-1$
Octal Integer	O"string" string String of 1 through 20 octal digits 0 through 7.
Hexadecimal Integer	X"string" string String of 1 through 15 hexadecimal digits 0 through 9 and A through F.
Character	"string" string String of 1 through 240 characters. If the character " is to appear in the string, it must be specified by two consecutive " marks.
Boolean	TRUE FALSE
Status	S"stvalue" stvalue Defined in STATUS declaration.

Symbol	Meaning
Numeric Operators	
+	Addition; unary plus.
-	Subtraction; unary minus.
*	Multiplication.
/	Division.
**	Exponentiation.
Masking Operators	
LNO	Logical NOT (bit-by-bit NOT).
LAN	Logical AND (bit-by-bit AND).
LOR	Logical OR (bit-by-bit OR).
LXR	Logical exclusive OR.
LIM	Logical imply.
LQV	Logical equivalent.
Relational Operators	
EQ	Is equal to.
GR	Is greater than.
GQ	Is greater than or equal to.
LQ	Is less than or equal to.
LS	Is less than.
NQ	Is not equal to.
Boolean Logical Operators	
NOT	Negation.
AND	Conjunction.
OR	Union.

DECLARATIONS

ARRAY HEADER DECLARATION

ARRAY name [low:up, low:up, . . .] alloc(esize);

name Identifier; optional unless in BASED ARRAY, XDEF, or XREF declaration.

low:up Lower and upper array bounds; maximum of seven dimensions.

alloc Allocation.

- P Parallel; first word of each entry is allocated contiguously followed by the second word of each entry, and so forth.
- S Serial; all the words of one entry are allocated contiguously.

esize Number of words in array entry.

ARRAY ITEM DECLARATION

ITEM name type(ep,fbit,size)= [preset] ,
name type(ep,fbit,size)= [preset] , . . . ;

name Identifier.

type Type of array item; optional.

B	Boolean
C	Character
I	Signed integer; default
U	Unsigned integer
R	Real
S:stlist	Status

ep Word number in which item occurs; word numbering starts at 0.

fbit Bit position at which item begins.

size Item length.

preset Initialization value; optional. Not relevant for based arrays.

BASED ARRAY DECLARATION

BASED array-dec

BASED BEGIN array-dec array-dec ... END

array-dec Full ARRAY declaration including the ARRAY header declaration and the ARRAY ITEM declaration.

COMMON DECLARATION

COMMON name; data-dec

COMMON name; BEGIN data-dec data-dec ... END

name Name of common; if omitted, blank common is used.

data-dec Declaration for a scalar, array or based array. Blank common cannot be initialized at load time.

FUNCTION AS FORMAL PARAMETER DECLARATION

FUNC name type;

name Formal name of function.

type Type of function.

B	Boolean
C(lgth)	Character with lgth characters
I	Signed integer
U	Unsigned integer
R	Real
S	Status

ITEM DECLARATION

ITEM name type=preset, name type=preset, . . . ;

name Identifier.

type Type of item; optional.

B	Boolean
C(lgth)	Character with lgth characters
I	Signed integer; default
U	Unsigned integer
R	Real
S:stlist	Status

preset Initialization value; optional.

LABEL DECLARATION

LABEL name, name, . . . ;

name Identifier.

PROCEDURE AS FORMAL PARAMETER DECLARATION

FPRC name;

name Formal name of procedure.

STATUS DECLARATION

STATUS stlist identifier, identifier, . . . ;

stlist Name of list.

identifier Status value; can duplicate a reserved word. The first identifier is assigned a value of 0.

STATUS SWITCH DECLARATION

SWITCH swname:stlist label:stvalue, label:stvalue, . . . ;

swname	Name of switch.
stlist	Name of status list as declared by a STATUS declaration.
label	Label name.
stvalue	Status value from stlist.

SWITCH DECLARATION

SWITCH swname label, label, . . . ;

swname	Name of switch.
label	Label name.

XDEF DECLARATION

XDEF xdec

XDEF BEGIN xdec xdec . . . END

xdec One of the following that is to be referenced in an externally compiled program:

Full data declaration for scalar, array, based array, or switch.

PROC name;

FUNC name type;

LABEL name, name, . . . ;

XREF DECLARATION

XREF xdec

XREF BEGIN xdec xdec ... END

xdec Any of the following whose storage is declared with XDEF:

Data declaration for a scalar without preset.

Data declaration for an array without preset.

Data declaration for a based array.

PROC name;

FUNC name type;

LABEL name, name, ... ;

SWITCH name, name, ... ;

EXCHANGE STATEMENT

v1 == v2;

v1, v2 One of the following:

Scalar
Subscripted array item
P-function
Bead function

FOR STATEMENT

FOR i=aexp1 STEP aexp2 DO statement
FOR i=aexp1 STEP aexp2 UNTIL aexp3 DO statement
FOR i=aexp1 WHILE bexp DO statement
FOR i=aexp1 STEP aexp2 WHILE bexp DO statement
FOR i=aexp1 DO statement

i	Counter for loop.
aexp1	Arithmetic expression indicating initial value of i.
aexp2	Arithmetic expression indicating value to be added to i.
aexp3	Arithmetic expression indicating last value for i.
statement	Statement to be repeatedly executed.
bexp	Boolean expression that must be TRUE for loop execution.

GOTO STATEMENT

GOTO label;

label Name of a label.

GOTO swlist [exp] ;

swlist Name of a switch list.

exp Arithmetic expression whose value is an integer.

IF STATEMENT

IF bexp THEN statement1

IF bexp THEN statement1 ESLE statement2

bexp	Boolean expression.
statement1	Statement executed when bexp is TRUE.
statement2	Statement executed when bexp is FALSE.

LABELED STATEMENT

name: statement

name	Identifier.
statement	Any executable statement; can be a labeled statement.

REPLACEMENT STATEMENT

v = exp;

v	One of the following:
	Scalar
	Subscripted array item
	P-function
	Bead function
	Function name
exp	Arithmetic or relational expression.

RETURN STATEMENT

RETURN;

STOP STATEMENT

STOP;

TERM STATEMENT

TERM

TEST STATEMENT†

TEST name;

name Name of the item used as counter for the loop.

†The TEST statement is used only within a FOR statement.

COMPILER DIRECTIVES

$BEGIN DIRECTIVE†

$BEGIN

CONTROL DIRECTIVE

Attributes of Variables

CONTROL attribute var, var, . . . ;

attribute — One of the following:

OVERLAP	Variable is referenced by more than one name.
DISJOINT	Variable is referenced by only one name.
REACTIVE	Array has overlapping subscript references.
INERT	Array has no overlapping subscript references.

var — Variable with specified attribute; if omitted, all subsequently declared variables are affected until respecified.

Core Residence Selection

CONTROL LEVELn name, name, . . . ;

n — Memory in which common blocks or based arrays are to reside.

For CYBER 70 Model 76 and 7600 and CYBER 170 Model 176

1	Small core memory
2	Large core memory accessed directly
3	Large core memory accessed by block transfer to small core memory

†$BEGIN is used in conjunction with $END.

For CYBER 70 Models 71, 72, 73, and 74; CYBER 170 Models 171, 172, 173, 174, and 175; and 6000 series systems

1,2 Central memory.

3 Extended core storage accessed by block transfer to central memory.

name Name of a common block or based array.

FOR Loop Control

CONTROL looptype;

looptype Type of FOR loop to be generated

FASTLOOP	Test and branch at end of loop.
SLOWLOOP	Test and branch at beginning of loop.

Compilation Option Selection

CONTROL control-word;

control-word One of the following:

PACK	Pack switch code.
PRESET	Preset items in named common.
FTNCALL	Turn on control statement F parameter.
TRACEBACK	Generate traceback code.

Conditional Compilation†

CONTROL condition-word const1, const2;

condition-word	One of the following:	
	IFEQ	const1 equal to const2
	IFLS	const1 less than const2
	IFLQ	const1 less than or equal to const2
	IFGR	const1 greater than const2
	IFGQ	const1 greater than or equal to const2
	IFNQ	const1 not equal to const2
const1, const2	Constants to be tested.	

Listing Control

CONTROL control-word;

control-word	One of the following:	
	EJECT	Skip to new page
	LIST	Resume listing
	NOLIST	Suspend listing
	OBJLIST	List object code

Weak External

CONTROL WEAK name, name, . . . ;

name Name of array, based array, function, label, item, procedure, or switch.

†Conditional statements must be bracketed between the CONTROL directive and either CONTROL FI or CONTROL ENDIF.

DEF DIRECTIVE

DEF name (param,param, . . .) #character-string#;

name	Identifier by which character string is to be referenced.
param	Optional formal parameter.
character-string	DEF body that is to replace references to the DEF name.

$END DIRECTIVE†

$END

†$END is used in conjunction with $BEGIN.

PROGRAM STRUCTURE

MAIN PROGRAM

A main program consists of a program header followed by a series of declarations and statements and ended by a TERM statement.

The format of a main program header is

PRGM name;

name Identifier by which program is known.

PROCEDURES

A procedure consists of a procedure header followed by an optional series of declarations and the procedure body of a single or compound statement.

The format of a procedure header is

PROC name (param, param, . . .);

name Identifier by which procedure is known.

param Formal parameter; optional.

Scalar	Array
Label	Based array
Procedure	Function

A function consists of a function header followed by an optional series of declarations and the function body of a single or compound statement.

The format of a function header is

FUNC name (param, param, . . .) type;

name Identifier by which function is known.

param Formal parameter; optional.

Scalar	Array
Label	Based array
Procedure	Function

type Type of function result; optional.

B	Boolean
I	Integer; default
U	Unsigned integer
R	Real
C(lgth)	Character of length lgth
S	Status

ALTERNATIVE ENTRY POINTS

ENTRY PROC name (param, param, . . .);

ENTRY FUNC name (param,param, . . .) type;

name Identifier by which procedure or function is known.

param Formal parameter; optional.

Scalar	Array
Label	Based array
Procedure	Function

type Type of function result; optional.

B	Boolean
I	Integer; default
U	Unsigned integer
R	Real
C(lgth)	Character of length lgth
S	Status

EXECUTION TIME OUTPUT PROCEDURES

To use the output procedures, a FORTRAN Extended main program must call the SYMPL program. Within the SYMPL program, the output procedures must be declared as external references.

PROCEDURE PRINT

PRINT is used to format information to be written to the file OUTPUT.

PRINT(format-string);

format-string	Literal duplicating the specifications of a FORTRAN format specification.

PROCEDURE PRINTFL

PRINTFL is used to format information to be written to a file with a logical file name other than OUTPUT.

PRINTFL (format-string,file);

format-string	Literal duplicating the specifications of a FORTRAN format specification.
file	File on which the information is to be written, expressed in terms of the file information table for the file.

PROCEDURE LIST

LIST is used to specify an item or array to be printed.

LIST(argument);

argument	Item, expression, subscripted array item, etc. whose value is to be printed.

PROCEDURE ENDL

ENDL is used to terminate an output sequence begun by PRINT or PRINTFL.

ENDL;

INTRINSIC FUNCTIONS

ABS FUNCTION

ABS(exp)

exp Expression whose absolute value is to be returned.

The type of argument determines the type returned.

Argument Type	Type Returned
Real	Real
Integer	Unsigned integer
Other	Same as argument specified in call

B FUNCTION

B ⟨first,lgth⟩ iname

first Arithmetic expression specifying first bit to be extracted.

lgth Optional arithmetic expression specifying the number of bits to be extracted; default is 1.

iname Name of scalar or array item from which bits are to be extracted.

C FUNCTION

C ⟨first,lgth⟩ iname

first Arithmetic expression specifying the first character to be extracted.

lgth Optional arithmetic expression specifying the number of characters to be extracted; default is 1.

iname Name of scalar or array item from which characters are to be extracted.

LOC FUNCTION

LOC(argument)

argument Name of any of the following:

Scalar

Subscripted array item

Procedure

Function

Label

Switch

Array

P-function

P FUNCTION

P⟨barray⟩

barray Name of a based array.

SYMPL CONTROL STATEMENT

SYMPL.
SYMPL,parameter-list.

A (Abort Job After Errors)	omitted	Execute next control statement whether or not errors occurred.
	A	Execute control statement after EXIT(S) control statement if errors occurred.
B (Binary Code File)	omitted B	Write binary output to file LGO.
	B=0	Suppress generation of binary output.
	B=lfn	Write binary output to file lfn.
C (Check Switch Range)	omitted	Do not generate code to check range of switch references.
	C	Generate code to check range of switch references.
D (Pack Switches)	omitted	Generate one word for each switch.
	D	Generate one word with two switch points.
E (Compile $BEGIN/$END Statements)	omitted	Do not compile statements between $BEGIN and $END.
	E	Compile statements between $BEGIN and $END.
F (FORTRAN Calling Sequence)	omitted	Do not compile a word of all zeros at the end of parameter lists.
	F	Compile a word of all zeros at the end of parameter lists.
H (List All Source Statements)	omitted	List statements according to CONTROL NOLIST/ LIST statements.
	H	List all statements.

I (Source Input File)	omitted	Compile card images from file INPUT.
	I	Compile card images from file COMIPLE.
	I=lfn	Compile card images from file lfn.
K (Points-Not-Tested)	omitted	Do not generate points-not-tested interface code.
	K	Generate points-not-tested interface code and trace-back code.
L (Listing File)†	omitted L	Write source statements and diagnostics to file OUTPUT.
	L=1	Write summary of resources used to file OUTPUT.
	L=0	Suppress all listing output except diagnostics.
	L=lfn	Write source statements and diagnostics to file lfn.
N (Cross-Reference Unreferenced Items)	omitted	List only referenced items on the cross reference map.
	N	List all items on the cross reference map.
O (List Object Code)†	omitted	Do not list binary object code.
	O=st/end	List binary object code by range of source statements indicated. st Number of first statement; default is 0. end Number of last statement; default is last statement.
	O=lfn/st/end	List binary object code from specified statements on file lfn.

†Any L, O, R, or X parameters must be concatenated together.

P (Preset Common)	omitted	Data items in common blocks are not to be initialized.
	P	Initialize data items in common blocks according to preset values in the data declarations.
R (List Cross-Reference Map)†	omitted	Do not list cross reference table and common blocks.
	R	List cross reference table and common blocks on file OUTPUT.
	R=lfn	List cross reference table and common blocks on file lfn.
S (Execution Library)	omitted	Compile LDSET tables with references to these libraries: SYMLIB/FORTRAN (NOS and NOS/BE) SYMIO/FORTRAN (SCOPE 2)
	S=0	Suppress LDSET table generation.
	S=lib1/lib2/. . .	Generate LDSET tables with references to libraries lib.
T (Syntax Check)	omitted	Check syntax and generate binary code.
	T	Check syntax but do not generate binary code.

†Any L, O, R, or X parameters must be concatenated together.

W (Single Statement Code Generation)	omitted	Generate object code with multiple source statements intermixed.
	W	Generate object code that maintains a close correspondence with its source statement.
X (List Storage Map)†	omitted	Do not list storage map or common blocks.
	X	List storage map and common blocks on file OUTPUT.
	X=lfn	List storage map and common blocks on file lfn.
Y (Suppress Diagnostic 136)	omitted	List diagnostic 136.
	Y	Suppress diagnostic 136 but take corrective action.

†Any L, O, R, or X parameters must be concatenated together.

RESERVED WORDS

ABS
AND
ARRAY

BASED
BEGIN

COMMON
CONTROL

DEF
DO

ELSE
END
ENTRY
EQ

FALSE
FOR
FPRC
FUNC

GOTO
GQ
GR

IF
ITEM

LABEL
LAN
LIM
LNO
LOC
LOR
LQ
LQV
LS
LXR

NOT
NQ

OR

PRGM
PROC

RETURN

STATUS
STEP
STOP
SWITCH

TERM
TEST
THEN
TRUE

UNTIL

WHILE

XDEF
XREF

$BEGIN
$END

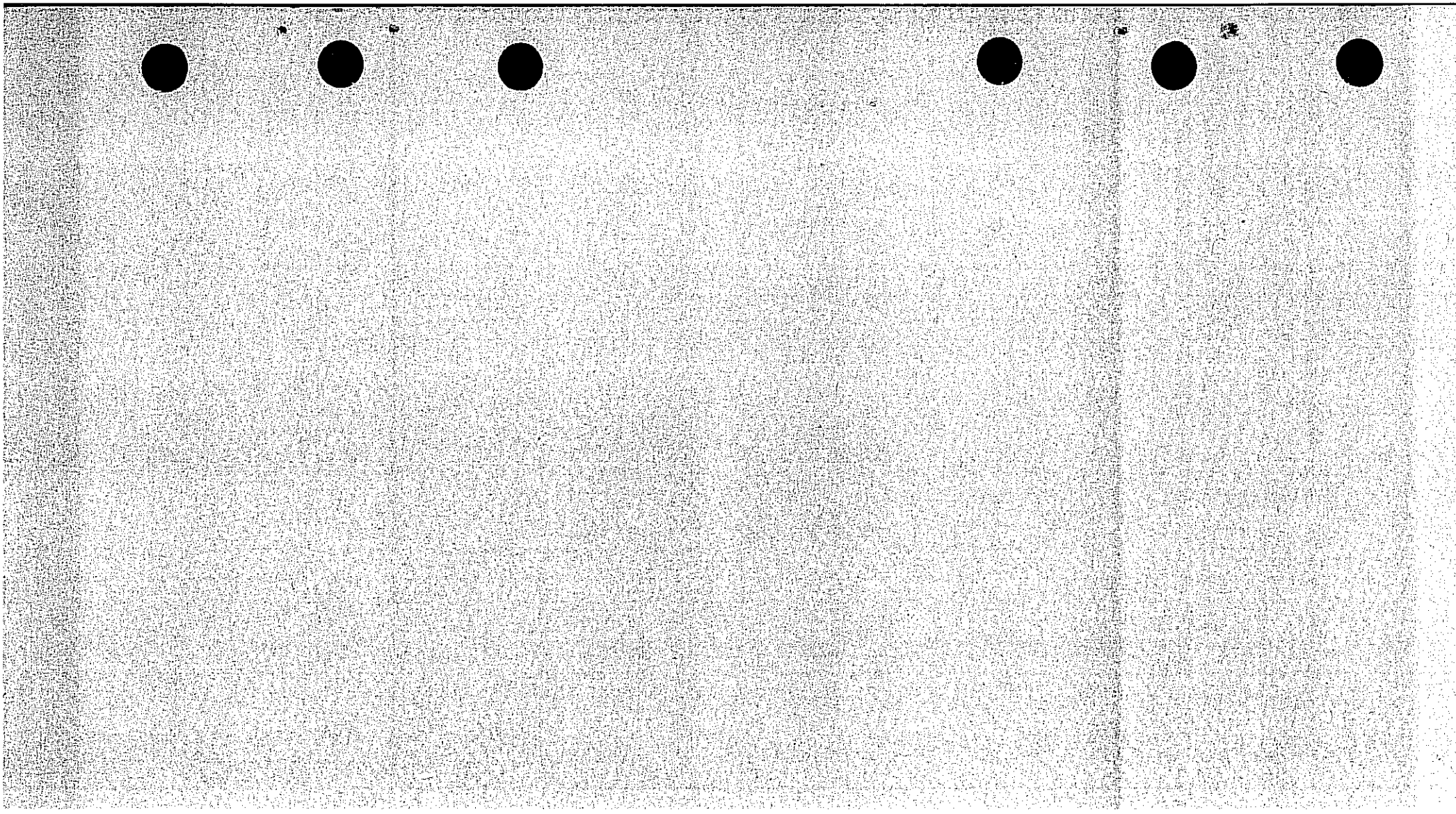
STANDARD CHARACTER SETS

Display Code (octal)	CDC Graphic	CDC Hollerith Punch (026)	CDC External BCD Code	ASCII Graphic Subset	ASCII Punch (029)	ASCII Code (octal)
00[†]	: (colon)[††]	8-2	00	: (colon)[††]	8-2	072
01	A	12-1	61	A	12-1	101
02	B	12-2	62	B	12-2	102
03	C	12-3	63	C	12-3	103
04	D	12-4	64	D	12-4	104
05	E	12-5	65	E	12-5	105
06	F	12-6	66	F	12-6	106
07	G	12-7	67	G	12-7	107
10	H	12-8	70	H	12-8	110
11	I	12-9	71	I	12-9	111
12	J	11-1	41	J	11-1	112
13	K	11-2	42	K	11-2	113
14	L	11-3	43	L	11-3	114
15	M	11-4	44	M	11-4	115
16	N	11-5	45	N	11-5	116
17	O	11-6	46	O	11-6	117
20	P	11-7	47	P	11-7	120
21	Q	11-8	50	Q	11-8	121
22	R	11-9	51	R	11-9	122
23	S	0-2	22	S	0-2	123
24	T	0-3	23	T	0-3	124
25	U	0-4	24	U	0-4	125
26	V	0-5	25	V	0-5	126
27	W	0-6	26	W	0-6	127
30	X	0-7	27	X	0-7	130
31	Y	0-8	30	Y	0-8	131
32	Z	0-9	31	Z	0-9	132
33	0	0	12	0	0	060
34	1	1	01	1	1	061
35	2	2	02	2	2	062
36	3	3	03	3	3	063
37	4	4	04	4	4	064
40	5	5	05	5	5	065
41	6	6	06	6	6	066
42	7	7	07	7	7	067
43	8	8	10	8	8	070
44	9	9	11	9	9	071
45	+	12	60	+	12-8-6	053
46	-	11	40	-	11	055
47	*	11-8-4	54	*	11-8-4	052
50	/	0-1	21	/	0-1	057
51	(	0-8-4	34	(	12-8-5	050
52	)	12-8-4	74	)	11-8-5	051
53	$	11-8-3	53	$	11-8-3	044
54	=	8-3	13	=	8-6	075
55	blank	no punch	20	blank	no punch	040
56	, (comma)	0-8-3	33	, (comma)	0-8-3	054
57	. (period)	12-8-3	73	. (period)	12-8-3	056
60	≡	0-8-6	36	#	8-3	043
61	[	8-7	17	[	12-8-2	133
62	]	0-8-2	32	]	11-8-2	135
63	%[††]	8-6	16	%[††]	0-8-4	045
64	≠	8-4	14	" (quote)	8-7	042
65	¬	0-8-5	35	_ (underline)	0-8-5	137
66	∨	11-0 or 11-8-2[†††]	52	|	12-8-7 or 11-0[†††]	041
67	∧	0-8-7	37	&	12	046
70	↑	11-8-5	55	' (apostrophe)	8-5	047
71	↓	11-8-6	56	?	0-8-7	077
72	<	12-0 or 12-8-2[†††]	72	<	12-8-4 or 12-0[†††]	074
73	>	11-8-7	57	>	0-8-6	076
74	≤	8-5	15	@	8-4	100
75	≥	12-8-5	75	\	0-8-2	134
76	¬	12-8-6	76	~ (circumflex)	11-8-7	136
77	; (semicolon)	12-8-7	77	; (semicolon)	11-8-6	073

[†]Twelve zero bits at the end of a 60-bit word in a zero byte record are an end of record mark rather than two colons.

[††]In installations using a 63-graphic set, display code 00 has no associated graphic or card code; display code 63 is the colon (8-2 punch). The % graphic and related card codes do not exist and translations yield a blank (55_8).

[†††]The alternate Hollerith (026) and ASCII (029) punches are accepted for input only.



CORPORATE HEADQUARTERS
P.O. BOX O
MINNEAPOLIS, MINNESOTA 55440

SALES OFFICES AND SERVICE CENTERS
IN MAJOR CITIES
THROUGHOUT THE WORLD

PRINTED IN U.S.A.

CONTROL DATA CORPORATION