3400

COMPUTER SYSTEM

SCOPE

GENERAL INFORMATION MANUAL

CONTROL DATA
CORPORATION

3400

COMPUTER SYSTEM

SCOPE

GENERAL INFORMATION MANUAL

CONTROL DATA
CORPORATION

Additional copies of this manual may be obtained from the nearest Control Data Corporation Sales office listed on the back cover.

CONTROL DATA CORPORATION
Documentation Department
3145 PORTER DRIVE
PALO ALTO, CALIFORNIA

September, 1965
Pub. No. 60055600, Rev. A

CONTENTS

DESCRIPTION 1

SCOPE is an operating system which provides supervisory control of program execution for the CONTROL DATA® 3400 Computer. It facilitates programming and minimizes operating responsibilities, and makes efficient use of computer time. It is designed so that maximum memory area is available for job execution. SCOPE and other programming systems for the 3400 provide compatibility with the Control Data® 3600.

The SCOPE library contains (FORTRAN, COMPASS, ALGOL, COBOL, and SORT programs), therefore, those units required by the library programs are also required by SCOPE. The speed and efficiency of SCOPE is affected by the choice of input/output units.

SCOPE provides:

- Job processing control
- Input/output control
- Interrupt control
- Debugging aids
- Library preparation and maintenance
- Job stacking
- Labeling and tape assignment

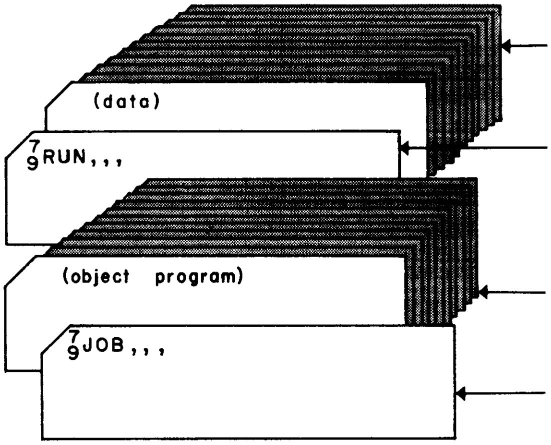
For efficient operation of SCOPE with the FORTRAN and COMPASS translators, the following equipment is required:

Quantity	Model	Description
1	3404	Computer
1	3403	32-K Memory
1	3401	Console with Typewriter
2	3406	Data Channels
1	3421	Magnetic Tape Controller (2 RW Control)
4	604 or 607	Magnetic Tapes
1	3446	Card Punch Controller*†
1	3447	Card Reader Controller with 405*
1	3655	Printer*

*For faster operation magnetic tapes may be substituted for these units.

†Compatible w/IBM 523 or 544.

This illustration demonstrates the simplicity with which a program may be executed under control of the SCOPE operating system.



Data, required by the object program, may follow on the standard input unit.

RUN control statement provides execution time limits and indicates the object program is to be executed. SCOPE transfers control to the object program.

SCOPE loads the relocatable binary object program into memory.

JOB control statement provides identification, accounting information, and a time limit for the job which follows.

More elaborate job stacking is available to suit the needs of a programmer or installation.

MAJOR FEATURES

- Job processing

  maintains accounting information

  initiates compilations and assemblies

  assigns equipment (optional tape labeling)

  loads and links subprograms

  allocates memory

  provides overlay processing

- Debugging aids

  snap dumps

  error dumps (recovery)

  octal corrections

  memory map

  diagnostics

- Program requests

  input/output control

  external interrupt control

  tape handling

  internal interrupt control

  sampling of time, date, equipment status, and available memory

- Library preparation and maintenance

  listing table of contents

  directory formation

  preparing library tape

● JOB PROCESSING

A group of jobs may be stacked on the standard input unit for processing under SCOPE. Jobs are normally processed in the order in which they appear, but the operator may alter this sequence by control statements. A job, delineated by JOB control statements, consists of one or more independent programs under the same account number, for example: named library programs (COMPASS, FORTRAN, ALGOL, COBOL), object programs, and/or library subroutines (SIN).

At the beginning of each job the previous job is closed out, saved tapes are unloaded, scratch and programmer units are released and equipment assignments and declarations from the previous job are cleared. The new job is logged in and the specified time limit is set.

Accounting information includes job identification, date, start, stop, and elapsed time. Job time limits and print line limits may be specified as well as the type of dump to be taken should the job terminate abnormally.

EQUIPMENT ASSIGNMENT

SCOPE locates and assigns physical equipment at run time according to specifications in the control statements and requests. All references to input/output units are by logical unit numbers chosen by the programmer, and independent of any physical unit designation. Logical units are programmer units, scratch units, and system units.

Programmer units, which may be saved, are retained throughout the job for reference by the program and released at the end of the job.

Scratch units, for temporary use during the operation of a program, are released after each program execution.

System units, used by SCOPE and the programmer, are not released until the end of the monitor sequence. The Load-and-Go unit is an exception; it is available for scratch purposes during job execution. System units include:

Standard Input

Standard Output

Input Comment

Output Comment

Accounting

Library

System Scratch

Load-and-Go

Unless otherwise informed, SCOPE assumes all programmer and scratch unit numbers refer to unique, hyper-density unlabeled magnetic tapes, read or written in binary mode. If another hardware type, mode, or density is required, it is specified by either a control statement or a request. When a logical unit is requested, SCOPE searches the table of input/output equipment available to the computer and makes the assignment.

Tapes may have a 10-word BCD record as a label. This record will be the first record on the tape; its format is described in the Appendix.

LOADER

Subprograms and library subroutines, written in different source languages, may be independently assembled into relocatable binary subprograms and subroutines. SCOPE loads the subprograms and subroutines, links them by relating external symbols to entry points, and allocates common areas. After the subprograms are loaded, the loader searches the SCOPE library directory for subroutines having entry points corresponding to the names of all undefined external symbols, then loads the subroutines, linking them to the subprograms. An option allows loading and partial execution of programs with missing subroutines.

MEMORY ALLOCATION

An object program is loaded into the highest locations of available memory, followed by labeled common. Numbered common overlays the loader, beginning at the first loader location and extending upward in memory. The loader area is not cleared before control is transferred to a loaded program.

SCOPE maintains a record of available memory, which consists of those locations which have not been assigned to a subprogram, common block, or monitor routine. The limits of available memory may be obtained or changed by a SCOPE request.

OVERLAYS

Programs which exceed memory capacity may be divided into logically independent parts, stored on a tape, and called into memory when needed. The program is divided into a main program and any number of overlays, which may be further divided into segments. The main program, which resides in memory, transfers control to each overlay as it is called in. Overlays in turn transfer control to each segment as it is called. The locations assigned to the main program, overlays, and segments are contained in the memory map.

● DEBUGGING AIDS

SCOPE contains facilities to debug programs, make corrections without reassembly, dump areas of memory at job termination, and obtain a map of absolute addresses assigned the program by the loader.

SNAP

Snap dumps are periodic dumps of specified areas of memory taken during execution of a running program. At load time the dump area, format, and frequency are specified. A printout of console registers may also be requested. The dump information is written on the standard output unit.

RECOVERY

A recovery dump is taken if a program terminates abnormally. The dump, written in octal on the standard output unit, may consist of a console dump, program dump, labeled or numbered common dump, or a dump of all memory including SCOPE. A dump may also be requested for jobs which terminate normally.

CORRECTIONS

An object subprogram deck may be octally corrected. Instructions and labeled common data may be altered; numbered common and instructions referencing externals may not.

MAP

After the subprogram is loaded, a memory map written on the standard output unit lists the absolute locations of subprograms, labeled and numbered common, and entry points.

DIAGNOSTICS

When the loader detects an error in the loading operation, a diagnostic is written on the standard output unit, and loading continues as though the error had not been encountered. No library subroutines are loaded. SCOPE terminates the run after loading the last subprogram and moves to the next run on the standard input unit.

● PROGRAM REQUESTS

Program requests specify input/output control, tape handling, internal interrupt, and special run information.

INPUT/OUTPUT

SCOPE handles all input/output requests made by a running program.. SCOPE, using the equipment tables which it maintains, determines whether or not a physical unit has been assigned to the request. If a unit is not assigned, SCOPE requests the operator to assign a tape for an input request or it assigns an available magnetic tape for an output request.

If the unit and a data channel are available, the unit is connected and the request is initiated. If the unit is available but no channel is available, the request is rejected. If the unit is not available, the request is rejected. In event of request rejection, control will be transferred to an address specified in the request.

EXTERNAL INTERRUPT

If an interrupt address is specified in the request, control is transferred to an interrupt subroutine at the end of an input/output operation. Registers are stored and control is transferred to the interrupt subroutine, with the status of the unit supplied. At the end of the interrupt subroutine, the programmer returns control to SCOPE, which restores the registers and processes any other interrupts before returning control to the main program.

INTERNAL INTERRUPT

Internal conditions which may be specified to interrupt program operation include arithmetic faults, bounds (memory limits) faults, clock interrupt, and manual intervention interrupts. When the condition occurs, SCOPE saves the registers, clears the interrupt, and transfers control to an interrupt subroutine supplied by the programmer. After processing in the interrupt subroutine is completed, control returns to SCOPE which processes any other interrupts, restores the registers, and returns control to the interrupted program.

SPECIAL REQUESTS

A running program may request SCOPE to position the system library tape at a record or directory on the tape. A running program may call upon the loader and if it is not in memory, SCOPE reads the loader from the system library. It is possible for the programmer to obtain or change the limits of memory by a special request. Another request allows the running program to return control to SCOPE for normal termination of the run.

● LIBRARY

The SCOPE library contains SCOPE routines, assemblers, compilers, and other information required by the installation. It may include absolute binary and relocatable binary programs, binary and BCD data, subroutine directories and user-defined information. A SCOPE library tape must begin with a binary record containing a bootstrap routine and a label. The first three files contain the SCOPE system; the remainder of the tape may contain any routines desired by the installation.

Use of an alternate library tape may be specified by means of a control card. While this assignment is in effect, library loading will be initiated from the specified tape. Only relocatable programs and their directories may be on an alternate library tape.

TABLE OF CONTENTS

The table of contents is a list of the positions of the subroutine directories, end-of-file marks, subroutine records, tables, and data on the library tape. The table of contents may be listed on the standard output unit by using a SCOPE control statement.

DIRECTORY

A subroutine directory consists of a binary "directory record" and the subroutine records which follow it on the library tape. The directory record contains all the information required by the loader to determine whether the routines following on the tape are needed for execution. There may be any number of directories.

PREPARING AND EDITING

A new library tape may be prepared from source library tapes, and/or subroutines and data from the standard input unit or other input units. The arrangement of records on the new library tape is determined by the order of control statements. Subroutine directories may be prepared and inserted on the tape as well as end-of-file marks. Records may be deleted or replaced and new records inserted to edit a library tape.

A programmer's option allows simultaneous preparation of duplicate copies of a new library tape.

SCOPE PROGRAMMING 2

SCOPE processes jobs under directions supplied by the programmer and operator through control statements and program requests. Program requests are statements which may be included in assembly language (COMPASS) programs to perform operations such as input/output control, tape handling, internal interrupt, and equipment status checks.

CONTROL STATEMENTS

Control statements direct SCOPE in job processing.

JOB PROCESSING

The control statements identify the job and initiate compilation, assembly, or execution of the object program. Control statements may also establish equipment assignments and describe overlays.

JOB, charge number, programmer identification, job time limit

A JOB statement signals job beginning, provides accounting information, identifies the programmer, and sets a processing time limit. It precedes all jobs.

EXECUTION

Library programs may be called to assemble or compile subprograms or programs. Object programs from the standard input unit or other input units may be loaded into memory and executed, or records may be transferred to a logical tape unit from the standard input unit.

program name $(s_1, s_2), p_1, p_2, \ldots, p_n$

program name	is the name of the library program to be called
s	is a SCOPE parameter to correct or debug the library program
p	is a parameter interpreted by the library program

A program statement loads and executes library programs, such as FORTRAN, COMPASS, and COBOL.

FILE, logical unit
FILE END

A FILE statement transfers binary records following it on the standard input unit to the specified logical unit until a FILE END statement is encountered.

LOAD, logical unit

This statement causes relocatable binary subprograms to be loaded into memory from a logical unit other than the standard input unit. (A binary object program on the standard input unit is loaded into memory unless a preceding control statement specifies otherwise.) After the program is loaded, control returns to the next SCOPE control statement.

LOADMAIN, logical unit

Loads the main segment of an overlay program into memory for execution.

RUN, time limit, print limit, recovery dump format, memory map, dump

A RUN statement transfers control to the object program already loaded into memory.

EQUIP, logical unit = declaration$_1$, declaration$_2$, . . . , declaration$_n$

declaration — may concern hardware type, usage, logical unit equivalence, and label declarations

AHT, available hardware table entry ordinal, parameter

This statement may be used to:

Obtain a listing of the available hardware table entry
Alter the status of an available hardware table unit
Alter the status of an entire available hardware table entry

FORCERUN

Undefined externals are patched to an error point and loading is terminated normally. This statement allows a program with missing subprograms to be run to partial completion.

DEBUGGING AIDS

SNAP, trigger address, first word to be dumped, last word to be dumpe format designator, indirect designator, initial pass number, last pass number, dump frequency, identification, and console scoop indicator

A SNAP is a periodic dump of specified areas in a running program.

OVERLAYS

A program which is too large for available memory may be divided into a main program and any number of overlays and segments.

MAIN, logical unit

MAIN defines the main portion of the program and the logical unit on which it is stored.

OVERLAY, logical unit, numerical identifier

OVERLAY directs SCOPE to create an overlay section of a program and write it on a logical unit.

SEGMENT, logical unit, numerical identifier

SEGMENT directs SCOPE to create a segment section for an overlay and write it on a logical unit.

LIBRARY

This routine may be used to prepare a new library or list the table of contents of a library. These two control cards bracket all the following library control cards used to create the library tape:

```
PRELIB
*SCOPE
```

PREPARING LIBRARY

Library control cards to prepare a new library tape by extracting records from other tapes:

```
*PREPARE, tape name, number of new tapes (labels of
          source library tapes)

*EXTRACT, source library label (identifiers for the
          records to be transferred to the new library
          tape from the source library tape)
```

Library control cards to specify the mode in which the records are to be written on the new library tape:

```
*REL, name   or   *REL, u, f, name
*LNK, name   or   *LNK, u, f, name
*BIN, name   or   *BIN, u, f, name
*ABS, name   or   *ABS, u, f, name
*BCD, name   or   *BCD, u, f, name
```

u is logical unit number
f is file number

Other library control cards:

```
*STRIP, name   or   *STRIP, logical unit, file number, name
*DIR, name
*END
*EOF, name
*TOC, record number
```

Library control cards to create labels used by the resident portion of SCOPE:

```
*SYS, name
*DNT, name, hardware type
*AHT, available hardware table ordinal (hardware type,
        I/O indicator, channel number, equipment switch
        setting, equipment unit code)
*COT, 7n(name, tape file number, file record number,
        save loader flag)
*IHA, system mnemonic, AHT ordinal of unit
```

Library control cards to create macros used by COMPASS:

```
*MACRO
*SYSMACS
*LIBMACS
```

LISTING LIBRARY

A library tape table of contents may be listed with the library control card shown below. This card must be placed between a PRELIB card and an *SCOPE card.

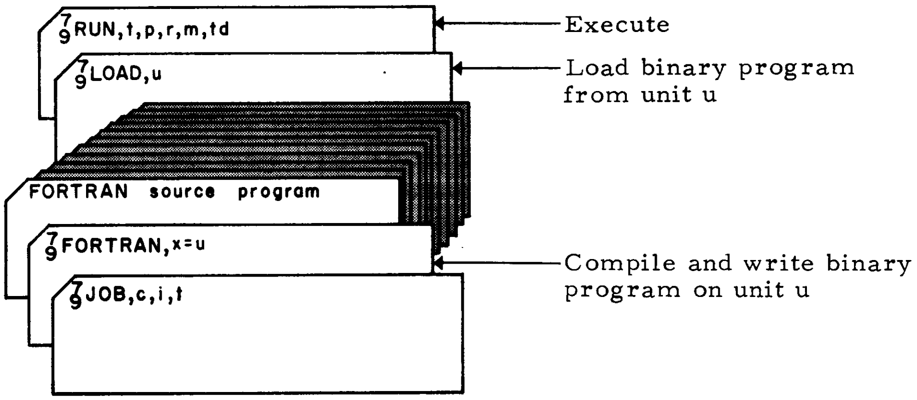
```
*LIST, (tape name)
```

EXAMPLE PROGRAMS

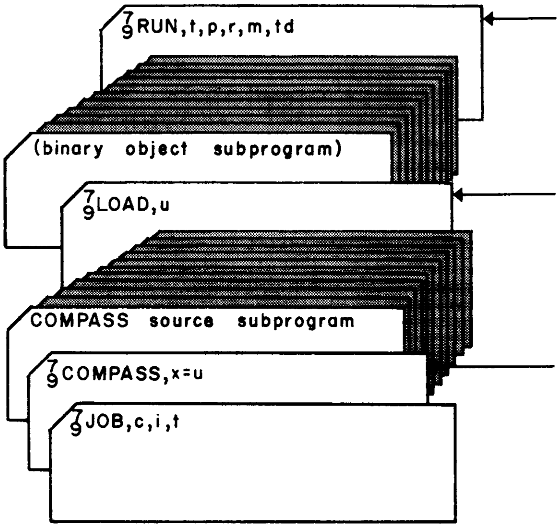
The following programs demonstrate the arrangement of control statements and subprograms on the standard input unit.

COMPILE AND EXECUTE

Compile the FORTRAN source program, writing binary output on unit u. Load and execute the program.



Assemble a COMPASS subprogram, and write binary output on unit u. Load with a binary subprogram and execute.

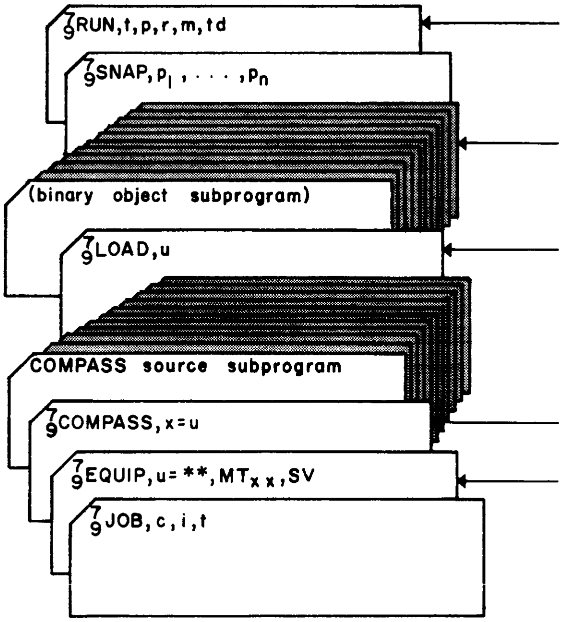


Execute

Load binary subprograms from unit u

Assemble and write the binary program on unit u

Include debugging aids in previous example and save unit u.


$^7_9$RUN,t,p,r,m,td
$^7_9$SNAP,p_1, . . . ,p_n
(binary object subprogram)
$^7_9$LOAD,u
COMPASS source subprogram
$^7_9$COMPASS,x=u
$^7_9$EQUIP,u=**,MT_{xx},SV
$^7_9$JOB,c,i,t

Execute

Load binary subprogram

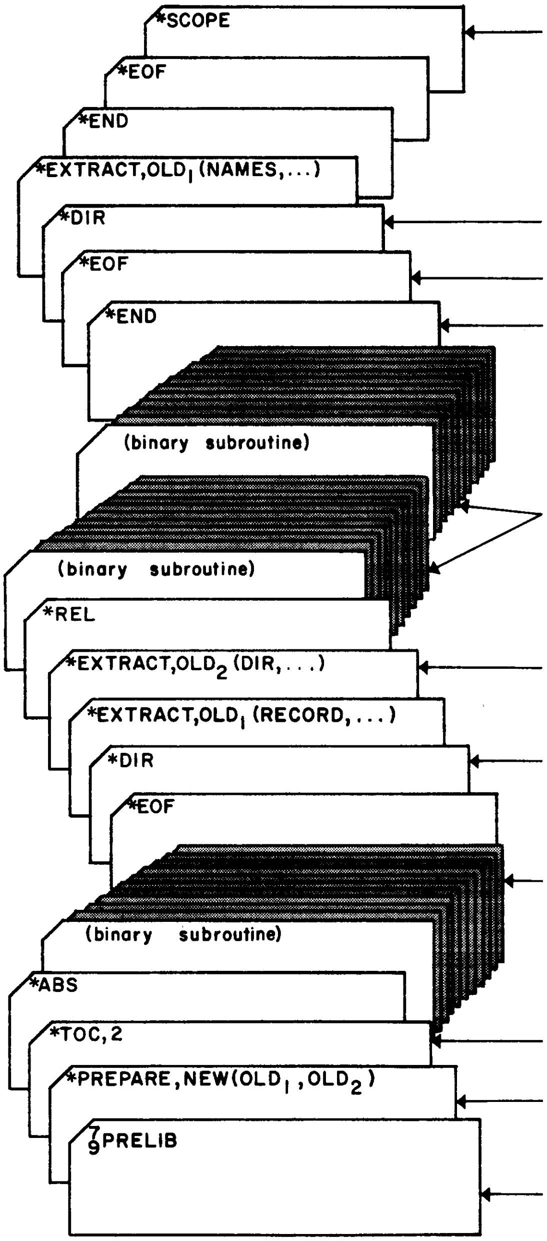
Load binary subprogram from unit u

Assemble and write on unit u

Save logical unit u

CREATE A LIBRARY TAPE

Create a library tape called NEW from two source library tapes called OLD_1 and OLD_2 and subroutines on the standard input unit.



End of library preparation

Directory

Written on new library

End of subroutines to be included in this directory

Information on standard input unit in relocatable binary form is written on new tape

Transfer relocatable records from source tapes to new tape

Place directory of all subsequent relocatable subroutines on new tape until END card is encountered

Write first record on tape in absolute format

Make table of contents second record

Prepare library tape named NEW using source tapes OLD_1 and OLD_2

Call SCOPE library preparation routine

PROGRAM REQUESTS

SCOPE controls the execution of input/output operations and the handling of internal interrupts. Requests are written in the form of COMPASS language macro instructions, which the COMPASS assembler translates into requests to SCOPE. SCOPE may execute the request immediately, stack the request temporarily and return control to the next instruction, or return control to a reject address if the request cannot be executed. When an input/output request is completed, SCOPE gives control to the programmer-defined interrupt subroutine. In the interrupt subroutine, the programmer can determine if the request was completed successfully and initiate remedial procedures if it was not. Requests may be given to establish operating modes or to save tapes at job termination.

Internal interrupt conditions which may be selected to produce program interrupts include errors resulting from shift and arithmetic instructions, memory guard for program operation and data storage, and time limits for program execution.

INPUT/OUTPUT

READ, logical unit, control word address, reject address, interrupt address

WRITE, logical unit, control word address, reject address, interrupt address

STATUS, logical unit, M

or

STATUS, logical unit

MODE, logical unit, reject address, declarations

Up to five mode declarations may be specified in each MODE request. These include declarations to:

Restrict the usage
Specify the recording format
Set the tape recording density
Specify the direction of tape motion

TAPE CONTROL

BSPR	backspace one record
BSPF	backspace one file
REWIND	set to load point
UNLOAD	rewind and unload
SKIP	move to end of file
MARKEF	write file mark
REOT	read after physical end of tape
WEOT	write after physical end of tape
ERASE	erase six inches of tape

SAVE, logical unit	Specify tape disposition
UNSAVE, logical unit	

INTERNAL INTERRUPTS

SELECT, interrupt feature, interrupt address

REMOVE, interrupt feature

Interrupt features:

SHIFT	shift fault
DIVIDE	divide fault
EXOV	exponent overflow fault
EXUN	exponent underflow fault
OVER	fixed point overflow fault
ABNORMAL	transfers control to interrupt address upon abnormal termination
INST	illegal instruction fault
OPER	operand parity fault
MANUAL	manual interrupt alert

BOUND, lower bound, upper bound, reject address, interrupt address

Sets the limits of core; transfer or data storage instructions referencing locations outside these bounds trigger a transfer to the interrupt address.

UNBOUND

Removes the effect of a BOUND request.

CLOCK INTERRUPTS

LIMIT, duration, reject address, interrupt address

Sets time limit, after which control transfers to the interrupt address.

FREE

Removes effect of a LIMIT request.

TIME

Enters time of day into Q register, and job time remaining or time until next LIMIT interrupt into A register.

DATE

Enters date in A register.

LABEL REQUESTS

LABEL, logical unit, location of name, edition, reel, retention code or date written

Specifies the label characteristics.

RDLABEL, logical unit, location of buffer, reject address, interrupt address

If the logical unit is unassigned, all unassigned tapes are searched for a label which is the same as that specified in an EQUIP statement or a LABEL request for the logical unit. If a tape with a matching label is found, it is assigned the logical unit specified by this request. Once the logical unit is assigned and the tape is at load point, the label is read into the buffer area.

WRLABEL, logical unit, location of buffer, reject address, interrupt address

When the tape is at load point, the label information specified in an EQUIP statement or LABEL request for the logical unit is written onto the tape.

SPECIAL REQUESTS

LIBRARY, library unit, reject address, location of record name, record number

Positions the specified library tape.

LOADER

Calls the loader while program is running.

CORE, lower limit, upper limit

If this request is made without parameters, available memory is indicated in the A register. If parameters are specified, available memory is altered.

RELEASE, logical unit, reject address, release parameter

This statement clears system information relating a logical unit to a physical unit. If the physical unit is a tape, additional action is taken, depending on the release parameter; if unit is not tape, the parameter is ignored.

ABEXIT

Control is returned from a running program to SCOPE and a dump will be written on the standard output unit in the format specified on the run card. The dump is preceded by the message:

```
ABNORMAL EXIT TAKEN
```

EXIT

Control returns from a running program to SCOPE. Control may also be returned by transferring to the program entry point.

HERESAQ

The contents of the A and Q registers, which were to be restored following return to the interrupted job are replaced by the contents of A and Q at the time this request is given in an interrupt routine.

OPERATOR CONTROL STATEMENTS

The following control statements are input by the operator.

END SCOPE

This statement signals SCOPE that the processing of the job stack is completed.

PAUSE, job sequence number

SCOPE halts processing of the standard input unit just prior to loading the specified job. Processing is halted until a complete message is received from the operator.

REPEAT

The job in process is terminated abnormally and the standard input unit is repositioned at the beginning of that job. (Not available if input unit is a card reader.)

NEXT, job sequence number + increment

NEXT positions the standard input unit at the beginning of the job specified. SCOPE terminates the job it is processing when the command is given. (Not always valid if input unit is a card reader.)

$$\text{AHT, entry ordinal in AHT, }\left\{\begin{array}{l}\text{unit on which the entry is written}\\ \text{availability of the unit}\\ \text{value to replace the value in the entry}\end{array}\right.$$

This statement lists the contents of the entire Available Hardware Table or of an entry, alters the status of a unit, or establishes or alters an entire table entry.

SSON, $n_1, n_2, \ldots, n_n$

SSOFF, $n_1, n_2, \ldots, n_n$

These statements control the six monitor pseudo sense switches, turning them on or off. The term ALL may be used to specify all the switches.

OPERATING SCOPE 3

SCOPE minimizes operating responsibilities and simplifies operating procedures by maintaining accounting information, listing memory dumps, and recovery information.

INITIALIZATION

The operator loads SCOPE from the library tape by autoload. Switches indicate the channel and equipment to be connected and the function to be executed. The first record loaded from the library contains a bootstrap routine which performs initialization and requests the time and date from the operator. The operator responds via the input comment unit. SCOPE locates the library tape and the standard input unit which contains the job stack. Any changes or requests concerning the physical equipment available to SCOPE may be entered via control statements.

COMMUNICATION

Jobs are processed by SCOPE in the order in which they appear on the standard input unit. If the standard input unit is a tape, the operator may alter the job processing order by control statements.

The operator interrupts processing by pressing the manual interrupt button and entering a message via the input comment unit. The message may be either a control statement for SCOPE, or information for the program.

RECOVERY

If a program terminates abnormally, the recovery routine is called from the SCOPE library tape to make the core dump specified by the programmer. SCOPE is reloaded if necessary and proceeds with the next job, or translator run within the current job.

ORGANIZATION OF SCOPE 4

SCOPE LIBRARY

The SCOPE program contained on the SCOPE library tape is composed of routines and tables which monitor job processing. SCOPE is contained in the first three files of the SCOPE library tape as shown below. The remainder of the tape may contain subroutine directories, library subroutines, data, and tables. The first file contains the bootstrap which reads in SCOPE, recovery which terminates incorrect jobs and continues with the next job, the library table of contents, and label. The label is the last four words of the first record.

RESIDENT

Resident is that portion of SCOPE which remains in memory at run time. Resident is composed of equipment assignment, interrupt control, input/output control, central control routine and equipment drivers (if on-line operation). Resident and the bootstrap with recovery routines occupy the lowest region (00000$_8$) of memory. Other SCOPE programs are loaded next to resident.

		Library Tape
SCOPE	File 1	Bootstrap + Label Recovery Table of Contents
	File 2 (Resident)	Equipment Assignment and Tables Interrupt Control Input/Output Control Driver, if On-Line Central Control
	File 3	Relocatable Binary Loader JOB Processor, SNAP
	File 4-n	Relocatable Library Subroutines within Directories
	Other Files	Data Tables

RELOCATABLE LOADER

The central control routine of SCOPE transfers control to the loader for loading drivers and programs to be executed. The loader may also be called by a running program. When loading is complete, control is returned to the calling program.

EQUIPMENT TABLES

The SCOPE hardware assignment routine maintains tables relating logical units to physical equipment. None of the tables is directly available to the programmer.

The Available Hardware Table is a record of the status and physical characteristics of equipment available to the system at a particular installation. It is part of SCOPE, but it may be altered temporarily in memory by an operator control statement.

LABEL FORMAT APPENDIX

Labeled tapes have an 80-character identification as the first record. The identification format is described below. Tapes without such identification are unlabeled, with the following exception: All library tapes have a label as the last four words of the first record, and this label must be written backwards in order to be recognized through a LABEL request.

FORMAT

Character Position	Description
1	Density: 2 = 200 bpi 5 = 556 bpi 8 = 800 bpi
2-3	Label Identifier -- left and right parentheses ( )
4-5	Unused
6-8	Retention Code -- number of days from date written that tape is to be retained. 000 indicates the tape is available for immediate reuse upon release; 999 indicates tape will be retained indefinitely.
9-22	Name -- 14 alphanumeric characters supplied by the user as the tape identification.
23-24	Reel Number -- two BCD digits to differentiate tapes with identical names.
25-30	Date Written -- month, day, year (mmddyy) in BCD; automatically supplied by SCOPE when writing labels.
31-32	Edition Number -- two BCD digits to designate a specific version of the tape.
33-80	User Information -- any BCD information.

CONTROL DATA

CORPORATION

COMMENT AND EVALUATION SHEET

3400 Computer System
SCOPE General Information Manual
Pub. No. 60055600, Rev. A September, 1965

YOUR EVALUATION OF THIS MANUAL WILL BE WELCOMED BY CONTROL DATA CORPORATION. ANY ERRORS, SUGGESTED ADDITIONS OR DELETIONS, OR GENERAL COMMENTS MAY BE MADE BELOW. PLEASE INCLUDE PAGE NUMBER REFERENCE.

FROM NAME : ____________________

BUSINESS ADDRESS : ____________________

NO POSTAGE STAMP NECESSARY IF MAILED IN U. S. A.

FOLD ON DOTTED LINES AND STAPLE

STAPLE STAPLE

FOLD FOLD

FIRST CLASS
PERMIT NO. 8241

MINNEAPOLIS, MINN.

BUSINESS REPLY MAIL

NO POSTAGE STAMP NECESSARY IF MAILED IN U.S.A.

POSTAGE WILL BE PAID BY

CONTROL DATA CORPORATION
Documentation Department
3145 PORTER DRIVE
PALO ALTO, CALIFORNIA

FOLD FOLD

STAPLE STAPLE

3400 SCOPE GENERAL INFORMATION MANUAL

CONTROL DATA SALES OFFICES

ALAMOGORDO • ALBUQUERQUE • ATLANTA • BILLINGS • BOSTON • CAPE CANAVERAL • CHICAGO • CINCINNATI • CLEVELAND • COLORADO SPRINGS DALLAS • DAYTON • DENVER • DETROIT • DOWNEY, CALIFORNIA • GREENSBORO, NORTH CAROLINA • HONOLULU • HOUSTON • HUNTSVILLE • MIAMI MONTEREY, CALIFORNIA • INDIANAPOLIS • ITHACA • KANSAS CITY, KANSAS LOS ANGELES • MADISON, WISCONSIN • MINNEAPOLIS • NEWARK • NEW ORLEANS • NEW YORK CITY • OAKLAND • OMAHA • PALO ALTO • PHILADELPHIA • PHOENIX • PITTSBURGH • SACRAMENTO • SALT LAKE CITY SAN BERNARDINO • SAN DIEGO • SANTA BARBARA • SAN FRANCISCO SEATTLE • ST. LOUIS • TULSA • WASHINGTON, D. C.

AMSTERDAM • ATHENS • BOMBAY • CANBERRA • DUSSELDORF • FRANKFURT • HAMBURG • JOHANNESBURG • LONDON • LUCERNE • MELBOURNE MEXICO CITY • MILAN • MONTREAL • MUNICH • OSLO • OTTAWA • PARIS TEL AVIV • STOCKHOLM • STUTTGART • SYDNEY • TOKYO (C. ITOH ELECTRONIC COMPUTING SERVICE CO., LTD.) • TORONTO • ZURICH

8100 34th AVE. SO., MINNEAPOLIS, MINN. 55440

CONTROL DATA
CORPORATION

Pub. No. 60055600, Rev. A

LITHO IN U.S.A.