3100
3200
3300
3500

COMPUTER SYSTEMS
REAL-TIME 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

November, 1965
Pub. No. 60134300

CONTENTS

3100/3200/3300/3500 REAL-TIME SCOPE

Real-Time SCOPE is a monitor system of modular structure which increases programming and operating efficiency, yet requires a minimum of storage space and computing time. Real-Time SCOPE is designed around a resident program which maintains continuity of job processing and provides input/output control, time-sharing capabilities, priority interrupt handling, priority job processing, and data channel reservation for real-time operations. Additional routines for job sequencing, program debugging, library editing, and program loading are stored on the monitor library tape and called into storage when needed.

EQUIPMENT CONFIGURATION

Real-Time SCOPE is designed for the Control Data® 3100/3200/3300/3500 computer systems with the following minimum configuration:

1 3100, 3200, 3300, or 3500 computer (16K or 32K memory)

1 console typewriter

1 card reader or other suitable input device

1 line printer or other suitable output device

3 magnetic tape units

2 data channels

This configuration will allow FORTRAN, COMPASS, etc., to run as stacked jobs, but does not provide peripherals for a background program.

OPERATIONS

Real-Time SCOPE includes processors for the following operations:

- Stacked job processing
- Loading relocatable programs
- Logical to physical unit assignments
- Input/output control
- Priority interrupts
- Debugging aids
- Library preparation
- Overlay program formation and execution
- Calling and executing utility routines
- Time-sharing between foreground and background programs
- Memory protection if 3300/3500 multiprogramming option is selected

STACKED JOBS

Real-Time SCOPE allows stacked job processing; a stacked job is made up of one or more runs. A run consists of the execution of a library program, for example, FORTRAN, COMPASS, or COBOL; or the execution of a program loaded from specified logical units other than the system library. Job sequencing, program loading, and program execution are determined by control cards placed in the job deck.

A job card supplies the following information to the monitor system:

- account number
- programmer identification
- time limit
- option controls for suppressing system I/O protection and suppressing the recovery dump

Stacked job decks are made up of foreground programs and background programs. Foreground programs include normal execution tasks such as compilations and assemblies.

BACKGROUND PROGRAMS

The background program is a special-purpose program which, in general, either requires control at discrete intervals of time or at certain I/O-dependent conditions. It can be loaded between foreground programs and will reside in memory, along with foreground programs, until it is terminated.

The background program may be a standard Control Data product, such as the Simultaneous Peripheral Processor (SIPP), or it may be a user-written program.

LOADER

The modular Real-Time SCOPE loader is called into storage by the executive control routine whenever programs are to be loaded. The loader handles the following functions:

- Loads relocatable binary information, including the background program
- Links independently compiled subprograms
- Loads and links library routines referenced by a loaded program
- Determines the input/output drivers required for program execution and loads them from the library tape
- Assigns equipment to logical units for both the foreground and background programs
- Prepares program overlays and segments for foreground programs
- Selects, loads, and links BCD and floating-point simulation routines as needed
- Records errors detected during loading

Programs may be loaded from specified logical units or from the library tape. Each program may consist of one or more independently compiled subprograms; a subprogram may consist of computer instructions, data loaded into a data block, and reservations for common areas to be used during foreground program execution.

From the information on the loader cards which precede each subprogram, the loader assigns the program storage areas, loads the data and instructions into storage, and links symbols to entry points. After the program has been loaded, the equipment tables are examined to determine if drivers for I/O equipment need to be loaded; entry points for drivers are recorded and the drivers are read into memory from the library along with any other routines containing previously undefined entry point names. Control is then transferred to the loaded program.

TIME-SHARING

Real-Time SCOPE provides time-sharing between the background and foreground programs. Time-sharing is controlled by the background programs. When the background program is loaded, it is given control for initialization. Before giving up control by a normal return to Real-Time SCOPE through its entry point, the background program insures that control will be regained through an interrupt condition.

While foreground programs are being executed, for example, the background program could be waiting to execute peripheral jobs. Or, upon real-time interrupts, the background program could be handling data collection. The adaptability of Real-Time SCOPE to almost any program allows wide variety in selection of background programs by the user.

If the multiprogramming option has been selected for the 3300/3500 computer system, the memory used by the background program and by Real-Time SCOPE will be protected from the foreground programs. The memory protection feature disallows any of the following:

I/O by foreground programs

Use of HLT (halt) instruction

Use of DINT (disable interrupt) instruction

Any modification of Real-Time SCOPE tables, the clock and interrupt count, or the Interrupt Mask Register

SPECIAL FEATURES

LOAD TIME EQUIPMENT ASSIGNMENTS

Equipment assignments for a foreground program will be made at load time if the assignments are included on loader equipment declaration cards. These loader control cards give the logical unit number, hardware type, and particular hardware unit for each equipment assignment. From this information, the loader establishes the necessary linkages within the monitor hardware tables. When equipment assignments are made at load time, job control equipment cards (EQUIP) need not be included by the programmer each time the routine is used.

EXTERNAL SYMBOL DECLARATIONS

A number of external symbols may be equated to a common entry point by declaring them on an external symbol loader card. This eliminates the need to load into storage all subprograms referenced by either the foreground or background programs each time the program is run for debugging.

OVERLAYS AND SEGMENTS

A foreground program which is too large to fit into storage at one time may be divided into a main program and any number of overlays and segments which the loader will prepare and store on magnetic tape. The main program remains in storage for immediate execution, or may be called from the overlay tape on a subsequent run. The main program calls overlays, one at a time, into storage; overlays, in turn, call in segments.

During preparation of the overlay tape, the loader will write a storage map on the standard output unit, showing where the main program, overlays, and segments are loaded into storage. The loader also processes octal correction cards for each subdivision of the entire program.

UTILITY ROUTINES

Utility routines, which perform such operations as card-to-tape, tape copying, backspacing, and rewinding, may be called from the system library tape by foreground programs in the job stack. The routine control statement and its necessary parameters are entered on an input device such as a card reader, typewriter, or tape. The Real-Time SCOPE loader locates the routine, which is then executed under monitor control.

MANUAL INTERRUPT

Manual interrupt may be used for system control. The operator may input a message by first pressing the interrupt button and waiting for the typeload button to light. Pressing the interrupt button a second time signals the end of the message. If the input message is preceded by a slash, the message goes to the current foreground program. If the input message is preceded by an equal sign, the message goes to the background program. If there is no prefix, the message is interpreted by the monitor. For example, the manual interrupt feature could be used to request termination of either a foreground or a background program.

EQUIPMENT ASSIGNMENTS

Programmers refer to input/output equipment with logical unit numbers which the monitor system assigns to particular hardware units at load time or at run time. On a monitor control card, the programmer declares hardware type or a specific unit for each logical unit. The operator may also assign logical units with a typewriter control statement. Logical units are defined by an EQUIP statement preceding the job in which the units are used. Logical units may be assigned to particular physical units or to other logical units. Physical equipment assigned to the background program cannot be referenced by the foreground program.

FOREGROUND PROGRAM EQUIPMENT ASSIGNMENTS

As logical unit numbers are assigned, an entry is made in the Running Hardware Table for that logical unit number. Logical units used for stacked job processing are divided into three groups:

Programmer Units 1-49

Use of these units is unrestricted. They must be defined before they are referenced by the program and are released (and rewound, if magnetic tapes) by the monitor system each time a new job is encountered.

Scratch Units 50-54

Scratch units are used for storing intermediate program data. They must be defined before they are referenced by the program and are released by the monitor system after each program execution.

System Units 55-63

System units are assigned by the monitor system for particular job processing tasks.

BACKGROUND PROGRAM EQUIPMENT ASSIGNMENTS

The background program has no system units or scratch units; all unit numbers, 1-49, are programmer units. Although the standard input unit can be used to input control statements pertaining to the background program and allow a background program to be loaded into memory, the equipment assigned for foreground programs is not available for use by the background program during run time.

As logical unit assignments are made for the background program, an entry is placed in the Background Hardware Table. This separation of logical unit logging between the two programs allows complete freedom in use of logical unit numbers.

INPUT/OUTPUT CONTROL

Real-Time SCOPE allows one or two data channels to be declared as real-time channels. The real-time channels may be used for equipment requiring special handling of interrupts or input/output not provided by the system I/O control routines. All I/O on the real-time channels must be done by the user's own routine.

The Real-Time SCOPE input/output control system processes the input/output requests for the remaining data channels--those not used for real-time operations. The control system locates available channels, initiates the requested functions, and controls external interrupts. The following operations may be requested by assembly language calls:

READ	Read n words/characters; read a control word from disk into core storage.
WRITE	Write n words/characters; write a control word from core storage onto the disk.
READB	Read n words/characters backwards.
WRITECK	Compare n words/characters from computer core with n words/characters from drum.
WRITECHECK	Compare data in computer core with data on a disk.
REWIND	Rewind tape to load point.
UNLOAD	Rewind and unload tape.
BKSP	Backspace one record.
SEFF	Space forward past one end-of-file mark.
SEFB	Space backward past one end-of-file mark.
WEOF	Write end-of-file mark.
ERASE	Erase tape.
STATUS	Return unit status to AQ register.
FORMAT	Select a new format for the operation.
LOCATE	Locate a record in the file (seek).
SEARCH	Make a comparison of data on the disk with data in computer core.

The I/O control system governs the channel on which the input/output operations are performed and prevents any operation from destroying a previous operation still active on a selected channel.

When a system input/output request is made, the Real-Time SCOPE I/O routine performs the following operations:

Saves all necessary registers.

Locates and determines the availability of the unit and a channel. If either is unavailable, the request is rejected.

Connects the unit to the available channel.

Determines if the system unit is one on which the requested operation has been declared illegal. If so, the request is rejected and the job terminated.

Initiates the requested operation.

Restores saved registers.

The I/O control system allows the background program to share data channels with foreground programs in the job stack. The addition of an Equipment Status Table and an Equipment Interrupt Table allows both programs to share channels with maximum efficiency and no equipment status conflicts. As many as eight devices per channel may be active without information loss.

The I/O control system does not allow the background program to share equipment with the foreground programs, with one exception: both programs may reference the same mass storage device (disk, drum).

INTERRUPT CONTROL

The central interrupt control routine controls switching between background and foreground programs and gives priority to the background program. When an interrupt occurs, the Real-Time SCOPE interrupt routine performs the following functions:

Real-Time Operations	Non-Real-Time Operations
Stores the interrupt code	Stores the interrupt code
Disables all further interrupts	Disables all but real-time interrupts
Saves the A and Q registers and the three index registers.	Saves the A and Q register and the three index registers.
Sets necessary time-sharing flags.	Sets necessary time-sharing flags.
Enters the user's interrupt subroutine.	Enters the user's interrupt subroutine.

Real-Time Operations	Non-Real-Time Operations
Clears time-sharing flags.	Clears time-sharing flags.
Restores all registers upon return from the interrupt subroutine.	Restores all registers upon return from the interrupt subroutine.
Enables interrupts.	Enables all previously disabled interrupts.
Returns to the interrupted program.	Returns to the interrupted program.

The user has the option of declaring one or two data channels as real-time channels. These data channels may then be used for high priority interrupts which will be processed by the interrupt control routine before all others. Although this routine provides for interrupt priority on two levels, the user may increase the number of levels of priority through use of the real-time channels.

SYSTEM UNIT PROTECTION

System unit protection for foreground program processing is a monitor option. If protection is selected, the standard input, output, punch, and library units will be protected as summarized below; otherwise this routine may be bypassed:

Standard Input	Prevents positioning of unit outside the range of the current job.
Standard Output	Prevents positioning of the unit before the current job and detects end-of-tape marks. If end of tape is detected, the tape will be rewound and the operator will be alerted to mount a continuation tape.
Library	Prevents writing on this unit.

MEMORY ALLOCATION PRINT

A memory allocation print provides the addresses where a program was loaded into storage. The print includes addresses for the following:

First word of all subprograms in the program

First and last words of the common block

First words of the data block

All program entry points

First word of the program extension area, if assigned

DEBUGGING

OCTAL CORRECTIONS

Corrections may be made in foreground programs by inserting octal correction cards subsequent to loading. These cards contain corrections encoded in octal digits to be inserted into storage to replace locations in the loaded program, and, if desired, to add portions to the program. Corrections may be loaded into any of the subprograms, the data block, or a loader-assigned program extension area.

RECOVERY DUMP

A recovery dump, which is taken when a program terminates abnormally, consists of an octal listing of the console conditions, register file, and all of storage. Also included is the address of the last instruction executed. Recovery dumps, which may be selected for foreground and background programs, depend upon parameter specifications of the JOB and BACK cards.

SNAP DUMPS

Snap dumps are taken of specified areas of storage during foreground program execution. The programmer designates the instruction address which Real-Time SCOPE uses to determine when to take the dump. Immediately before the instruction is executed, the areas specified by the programmer are dumped. These may include the console, register file, and blocks of storage.

FORTDUMP AND PROGDUMP

These routines are similar in effect to the snap dump, but they are called by the programmer at the object program level. FORTDUMP, used in FORTRAN programming, and PROGDUMP, used in COMPASS programming, may be called for debugging foreground programs only.

LIBRARY PREPARATION

New library tapes can be prepared and existing library tapes can be edited through the Real-Time SCOPE PRELIB routine. This routine, which is a part of the Real-Time SCOPE monitor system, is loaded into memory only when requested by the user.

Library preparation and editing are directed by Real-Time SCOPE control cards which identify macros, origin addresses for routines, and records to be deleted, replaced, or added to the library tape. The listing produced of the contents of a library tape may include the following:

All control statements used to prepare the tape

Name and record number of each program

All entry points and their absolute or relative locations

Absolute starting and ending locations or the length of each routine

The listing may be suppressed, but PRELIB diagnostics will always be printed.

FOREGROUND PROGRAM CONTROL CARDS

```
7
9SEQUENCE,j
```

$j \leq 999$, identifies the job

Precedes the card deck for a job. If jump key 6 is set, the system pauses for operator communication statements.

```
7
9ENDSCOPE
```

Signals the end of a stack of Real-Time SCOPE jobs.

```
7
9ENDREEL
```

Terminates a reel of magnetic tape containing a job stack.

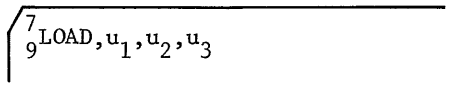
```
7
9JOB,c,i,t,NP,ND
```

Must be placed in front of the job deck. Parameters c and t are for accounting information; t is a time limit. The two optional fields NP and ND indicate system units protection and abnormal termination dump options.

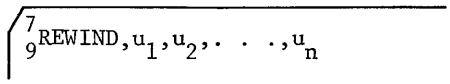
```
7
9EQUIP,u1=q1,. . .,un=qn
```

Used for programmer and scratch unit equipment declarations.

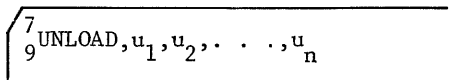
Transfers binary information from the standard input unit to unit u.



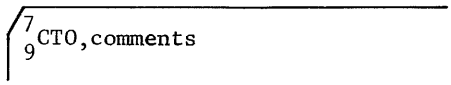
Specifies that binary records from u are to be loaded into memory. Up to three logical units, u_i, can be specified on one card; one file is loaded from each.



Rewinds specified tape units to load point.



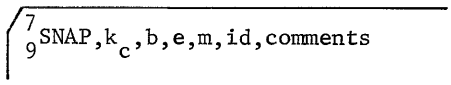
Rewinds and unloads specified tape units.



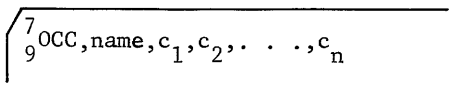
Prints comments to the operator on the console typewriter.

Initiates execution of a loaded program. The execution time limit is t; and NM, if it appears, suppresses the writing of a memory map.

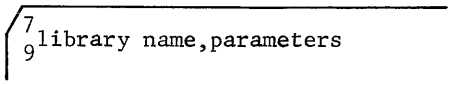
Provides a pause while the operator mounts new tapes, changes printer paper, etc.



Specifies debugging dumps to be taken at selected points during program execution.



Enters octal corrections following program loading. Name is a program address and c_i are octal corrections.

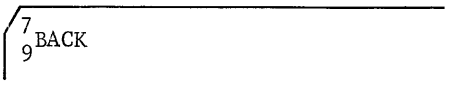


Loads and executes named library routines such as COMPASS and FORTRAN.

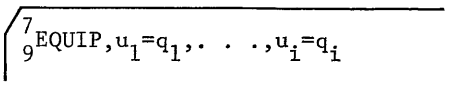
Loader card as a control card:

The programmer may place binary subprogram cards in the appropriate place in his deck. Upon reading a binary card on INP, Real-Time SCOPE calls the loader which reads INP until the next 7,9 control card is encountered.

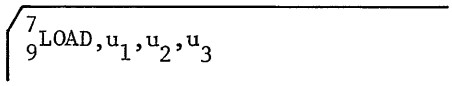
BACKGROUND PROGRAM CONTROL CARDS



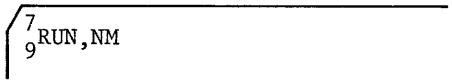
Informs Real-Time SCOPE that subsequent control cards up to the JOB control card pertain to background programs.



Assigns units for processing the background program.



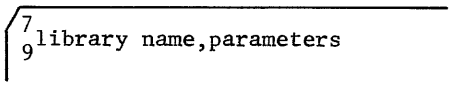
Specifies that the background program in binary form is to be loaded from the units indicated. Up to three units may be indicated; if no units are indicated, the program is loaded from the standard input unit of the foreground programs.



Initiates execution of the loaded background program; NM, if it appears, suppresses the writing of a memory map. Control cards following RUN will again refer to the foreground program.

Loader card as control card:

The background program may be loaded from INP without a LOAD card by placing the program binary deck in the INP deck. Upon reaching a binary card, Real-Time SCOPE calls the loader which reads INP until the next 7,9 control card is encountered.



Loads a named program from the library unit as a background program. This program is given control for initialization of parameters. Upon return, the background program will remain in core and all subsequent control cards pertain to foreground program processing.

OPERATOR COMMUNICATION

Typewriter Statements

AET, p_1, p_2

Changes the system available equipment table or obtains a listing of the current table contents.

SEQUENCE,j

Alters the job processing or repeats the execution of a job.

ENDSCOPE

Signals the end of the stack of jobs under Real-Time SCOPE.

EQUIP, u_1=q_1, ..., u_n=q_n

Used for programmer, scratch, or system unit assignments; when it appears between BACK and JOB cards, it specifies background logical unit assignments.

CALL, library name, p_i, ..., p_n

Calls a named library program for execution; primarily intended for calling utility routines.

REWIND, u_1, ..., u_n

Rewinds specified tape units to load point.

UNLOAD, u_1, ..., u_n

Rewinds and unloads specified tape units.

BACK

May appear only immediately after a SEQUENCE card and indicates that subsequent statements are for processing background program.

LOAD, u_1, u_2, u_3

May be used for loading background program only; and must appear between the BACK and JOB cards after any EQUIP assignments for the background program have been made. Zero to three units may be specified. If zero, loading is from the standard input unit of the foreground programs.

CONTROL DATA
CORPORATION

COMMENT AND EVALUATION SHEET

3100/3200/3300/3500 COMPUTER SYSTEMS
Real-Time SCOPE - General Information Manual
Pub. No. 60134300 November, 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

31/32/33/3500 REAL-TIME 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

CONTROL DATA CORPORATION

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

Pub. No. 60134300.

Litho in U.S.A.