Xerox Control Program for Real-Time (CP-R)

Xerox 550 and Sigma 9 Computers

Operations
Reference Manual

XEROX

Xerox Control Program for Real-Time (CP-R)

Xerox 550 and Sigma 9 Computers

Operations

Reference Manual

90 30 86D

March 1976

Printed in U.S.A.

REVISION

This publication is a major revision of the Xerox Control Program for Real-Time (CP-R)/RT, BP Operations Reference Manual for Xerox 550 and Sigma 9 Publication Number 90 30 86C (dated November 1974). The manual incorporates changes in the text that reflect version D00 of the CP-R operating system. A change in text from that of the previous manual is indicated by a vertical line in the margin of the page.

RELATED PUBLICATIONS

Title	Publication No.
Xerox 550 Computer/Reference Manual	90 30 77
Xerox Sigma 9 Computer/Reference Manual	90 17 33
Xerox Control Program for Real-Time (CP-R)/RT, BP Reference Manual	90 30 85
Xerox Control Program for Real-Time (CP-R)/System Technical Manual	90 30 88
Xerox Control Program for Real-Time (CP-R)/RT, BP User's Guide	90 30 87
Xerox Availability Features (CP-R) Reference Manual	90 31 10
Xerox Sigma Character-Oriented Communications Equipment/Reference Manual (Models 7611-7616/7620-7623	90 09 81
Xerox Sigma Mulitpurpose Keyboard Display/Reference Manual (Models 7550/7555)	90 09 82
Xerox Mathematical Routines/Technical Manual	90 09 06
Xerox Assembly Program (AP)/LN, OPS Reference Manual	90 30 00
Xerox SL-1/Reference Manual	90 16 76
Xerox Extended FORTRAN IV/LN Reference Manual	90 09 56
Xerox Extended FORTRAN IV/OPS Reference Manual	90 11 43
Xerox Extended FORTRAN/Library Technical Manual	90 15 24

Manual Content Codes: BP - batch processing, LN - language, OPS - operations, RP - remote processing, RT - real-time, SM - system management, TS - time-sharing, UT - utilities.

The specifications of the software system described in this publication are subject to change without notice. The availability or performance of some features may depend on a specific configuration of equipment such as additional tape units or larger memory. Customers should consult their Xerox sales representative for details.

CONTENTS

APPENDIXES

FIGURES

TABLES

PREFACE

This manual contains reference information for CP-R computer operators or others working at the console during attended runs. It describes operator communications with the system (i.e., key-ins and messages) system start-up and initialization, job and system control, peripheral device handling, and recovery procedures when applicable. Manuals offering other levels of information regarding CP-R features are outlined below.

- The Control Program for Real-Time/RT, BP Reference Manual, 90 30 83, contains reference information for the real-time and batch processing features of CP-R; (i.e., job control commands, system procedures, I/O procedures, program loading and execution, hardware interrupt and software interface, and service processors). The purpose of the manual is to define the rules for using background processing and real-time features.

- The Control Program for Real-Time/RT, BP User's Guide, 90 30 87, describes how to use the various batch and real-time features that are basic to most installations. It presents the information in a semitutorial format that offers the user of job-oriented approach toward learning the features of the operating system.

- The Control Program for Real-Time/System Technical Manual, 90 30 88, describes the internal function of the operating system. It is intended to be used with the listings supplied with each CP-R system for purposes of system maintenance.

- CP-R Availability Features Reference Manual, 90 31 10, describes the available techniques to rapidly identify a system problem as either a hardware or software malfunction that has already occurred, or to anticipate a potential system alarm. It also describes the techniques to further define the problem via software diagnostic criteria, including the Error Log Lister (ELLA), ANALYZE processor, On-Line Exercisers, and system alarm procedures. The manual is primarily addressed to computer operators, local system programmers and analysts, and Xerox Customer Service personnel.

Information for the language and applications processors that operate under CP-R is also described in separate manuals. These manuals are listed in the Related Publications page of this manual.

COMMAND SYNTAX NOTATION

Notation conventions used in command specifications and examples throughout this manual are listed below.

Notation	Description
lowercase letters	Lowercase letters identify an element that must be replaced with a user-selected value. CRndd could be entered as CRA03.
CAPITAL LETTERS	Capital letters must be entered as shown for input, and will be printed as shown in output. DPndd means "enter DP followed by the values for ndd".
[ ]	An element inside brackets is optional. Several elements placed one under the other inside a pair of brackets means that the user may select any one or none of those elements. [KEYM] means the term "KEYM" may be entered.
{ }	Elements placed one under the other inside a pair of braces identify a required choice. $\left\{\begin{matrix}\text{A}\\\text{id}\end{matrix}\right\}$ means that either the letter A or the value of id must be entered.
...	The horizontal ellipsis indicates that a previous bracketed element may be repeated, or that elements have been omitted. name[,name]... means that one or more name values may be entered, with a comma inserted between each name value.
⋮	The vertical ellipsis indicates that commands or instructions have been omitted. MASK2 DATA,2 X'1EF' ⋮ BYTE DATA,3 BA(L(59)) means that there are one or more statements omitted between the two DATA directives.
Numbers and special characters	Numbers that appear on the line (i.e., not subscripts), special symbols, and punctuation marks other than dotted lines, brackets, braces, and underlines appear as shown in output messages and must be entered as shown when input. (value) means that the proper value must be entered enclosed in parentheses; e.g., (234).
Subscripts	Subscripts indicate a first, second, etc., representation of a parameter that has a different value for each occurrence. sysid$_1$,sysid$_2$,sysid$_3$ means that three successive values for sysid should be entered, separated by commas.
Superscripts	Superscripts indicate shift keys to be used in combination with terminal keys. c is control shift, and s is case shift. L^{cs} means press the control and case shift (CONTROL and SHIFT) and the L key.
Underscore	All terminal output is underscored; terminal input is not. !RUN means that the exclamation point was sent to the terminal, but RUN was typed by the terminal user.
ESC RET LF	These symbols indicate that an ESC (ESC), carriage return (RET), or line feed (LF) character has been sent. !EDIT RET means that, after typing EDIT, a carriage return character has been sent.

GLOSSARY

active foreground program a program is active if it is resident in memory, connected to interrupts, or in the process of being entered into the system via a !RUN control command.

asynchronous events independent events that may be taking place concurrently to task execution (e.g., a read from magnetic tape or via an STIMER service call).

background area that area of core storage allocated to batch processing. This area may be checkpointed for use by foreground programs.

background program any program executed under monitor control in the background area with no external interrupts active. These programs are entered through the batch processing input stream.

binary input input from the device to which the BI (binary input) operational label is assigned.

centrally connected interrupt an interrupt that is connected to a monitor interrupt routine that first saves the environment of the system and then switches the environment to that of the task that gets control when the interrupt occurs.

control command any control message other than a key-in. A control command may be input via any device to which the system command input function has been assigned (normally a card reader).

control message any message received by the monitor that is either a control command or a control key-in (see "key-in").

Data Control Block (DCB) a table in the executing program that contains the information used by the monitor in the performance of an I/O operation.

dedicated memory core memory locations reserved by the monitor for special purposes such as traps, interrupts, and real-time programs.

directly connected interrupt an interrupt which, when it occurs, causes control to go directly to the task; e.g., execution of the XPSD instruction in the interrupt location gives control to the task rather than first going to a monitor interrupt routine.

disk pack a secondary storage system of removable rotating memory. For CP-R purposes, disk pack and RAD are synonymous unless otherwise noted.

dispatcher a set of CP-R routines that schedule secondary task on a software priority basis. There may be more than one dispatcher in a given system.

dummy section a type of program section that provides a means by which more than one subroutine may reference the same data (via an external definition used as a label for the dummy section).

end record the last record to be loaded in an object module or load module.

error severity level code a four-bit code indicating the severity of errors noted by the processor. This code is contained in the final byte of an object module.

execution location a value defining the origin of a relocatable program, to set the address at which program loading is to begin.

external definition a symbolic name that is declared to be "knowable" outside the range of the object module in which it is defined; a "global" symbol. An external definition allows the specified symbolic name to be used in external references (see below).

external reference a reference to a declared symbolic name that is not defined within the object module in which the reference occurs. An external reference can be satisfied only if the referenced name is defined by an external definition in another object module.

foreground area that portion of memory dedicated specifically for foreground tasks and programs.

foreground program a load module that contains one or more foreground tasks.

foreground task a body of procedural code that is associated with (connected to) a particular interrupt and that is executed when the interrupt occurs or when scheduled by a CP-R dispatcher.

Function Parameter Table (FPT) a table through which a user's program communicates with a monitor function (such as an I/O function).

GO file a temporary disk file or relocatable object modules formed by a processor.

granule a block of disk sectors containing a specified number of words.

idle state the state of the Monitor when it is first loaded into core memory or after encountering a !FIN control command. The idle state is ended by means of a !C key-in.

installation control command any control command used during system generation to direct the creation of a CP-R system.

job a job is a collection of one or more related tasks. A foreground job is composed of one or more related foreground tasks. A background job is composed of one or more job steps each comprised of a single task. Tasks executed in the background are not connected to interrupts. Background job steps are processed through the batch input stream. The job concept is used for resources (such as files or devices) that are shared at the job level.

key-in control information entered by the console operator via a keyboard.

keyword a word, consisting of from one to eight characters that identifies a particular operand used in a control command or operator key-in.

load item in object modules, a load-control byte followed by any additional bytes of load information pertaining to the function specified by the control byte.

load map a listing of significant information pertaining to the storage locations used by a program.

load module an executable program formed by using relocatable object modules (ROMs) and/or library object modules as source information. Primary load modules contain one or more primary tasks; secondary load modules contain a single secondary task.

logical device a peripheral device that is represented in a program by an operational label (e.g., BI or BO) rather than by a specific physical device name.

monitor a program that supervises the processing, loading, and execution of other programs, i.e., the control program.

object deck a card deck comprising one or more object modules and control commands.

object language the standard binary language in which the output of a compiler or assembler is expressed.

object module the series of records containing the load information pertaining to a single program or subprogram. Object modules serve as input to the Overlay Loader to form the load module.

operational label a two-character symbolic name used to identify a logical system device.

option an elective operand in a control command or procedure call, or an elective parameter in a Function Parameter Table.

OV file is a temporary dick file that contains an executable program formed by the Overlay Loader if a program file name was not specified at load time. Used primarily to test new programs or new versions of programs.

Overlay Loader a processor that links elements of overlay programs (ROMs) and forms executable programs.

overlay program a segmented program in which the segment currently being executed may overlay the core storage area occupied by a previously executed segment.

parameter presence indicator a bit in word 1 of a Function Parameter Table that indicates whether a particular parameter word is present in the remainder of the table.

physical device a peripheral device that is referred to by a "name" specifying the device type, I/O channel, and device number (also see "logical device").

postmortem dump a listing of the contents of a specified area of core memory, usually following the abortive execution processing of a program.

primary reference an external reference that must be satisfied by a corresponding external definition (capable of causing loading from the system library).

primary task a body of code that is connected to a hardware interrupt and that gains control of the CPU when that interrupt becomes active.

Program Trap Conditions (PTC) two words that indicate trap status (set or reset) and trap exit address respectively.

Rapid Access Data (RAD) storage system a secondary storage system of rotating memory. For disk purposes, RAD and disk pack are synonymous unless otherwise noted.

Relocatable Object Module (ROM) a program, or subprogram, in the form generated by a standard processor such as Macro-Symbol, FORTRAN, etc.

resident program a program that has been loaded into a dedicated area of core memory.

ROM Relocatable Object Module (see above).

secondary reference an external reference that may or may not be satisfied by a corresponding external definition (not capable of causing loading from the system library).

secondary storage any rapid-access storage medium other than memory (e.g., RAD or disk pack).

secondary task a body of code whose execution is scheduled by a CP-R dispatcher.

segment loader a monitor routine that loads overlay segments from disk storage at execution time.

source deck a card deck comprising a complete program or subprogram in symbolic EBCDIC format.

source language a language used to prepare a source program (and therefrom a source deck) suitable for processing by an assembler or compiler.

symbolic input input from the device to which the SI (symbolic input) operational label is assigned.

symbolic name an identifier that is associated with some particular source program statement or item so that symbolic references may be made to it even though its value may be subject to redefinition.

synchronous events events that must take place sequentially.

system library a group of standard routines in object language format, any of which may be included in a program being created.

task see "primary task" and "secondary task".

taskname identical to a load module name (lmn) unless redefined by the SETNAME CAL. All tasks within a load module have the same taskname.

temp stack push-down stacks created by the Overlay Loader and used by the monitor and system library routines.

1. INTRODUCTION

The Control Program for Real-Time (CP-R) for Xerox 550 and Sigma 9 Computers is the major control element of the operating system described in this manual. The total operating system includes the monitor, Overlay Loader, RADEDIT, language processors and user real-time and batch programs.

The content of this manual is operator-oriented in that it is specifically directed toward monitor/operator communications, procedures, control command formats, and device considerations necessary to maintain the system and process program inputs under monitor control. A comprehensive discussion of the internal functions of the monitor and its associated components is given in the Xerox Control Program for Real-Time/RT,BP Reference Manual, 90 30 85.

The CP-R system distinguishes three types of activities: foreground primary tasks, foreground secondary tasks, and batch tasks. The purpose of this division is to guarantee real-time tasks adequate memory and protection in the execution of highly critical processes, and at the same time, offer an efficient method for using the CPU when it is not busy with real-time processes. Limits on the division of system resources among these activities are defined at System Generation time (SYSGEN) in accordance with the needs of the local installation.

Typically, the operator's responsibilities encompass interaction with and control of a number of distinct classes of system activity including: real-time processing, batch processing, media conversion, symbiont operations, and remote terminal operations.

REAL-TIME PROGRAMS

Real-time programs directly or indirectly are connected to hardware interrupts that receive signals from external sources or real-time clocks to trigger execution and respond to these external events within microseconds.

Some real-time programs are loaded and initialized at System Boot time. Others can be run and released by the operator. The first method is used when the real-time process normally remains unchanged and is constantly operative. Typical examples would be a satellite tracking system or the control element of an automated plant or factory. The second approach is sometimes used when real-time operations are executed periodically or irregularly. An example would be a test procedure in an experimental laboratory.

BACKGROUND PROGRAMS

Background (batch) programs are assembled or compiled and executed in the background job. The CP-R operating system allows background programs to use the CPU when the real-time processes are not operative, thus giving greater economy to the system.

In contrast to real-time tasks whose priority sequence and function is controlled by external hardware interrupt or operator key-in, background programs are executed in a serial fashion and their sequence is controlled by control commands inserted in the job stack.

Background compilations or assemblies are initially loaded from some peripheral device (generally a card reader) onto a file in the Background Programs area of the disk and executed in background core storage in serial fashion.

SOFTWARE ENVIRONMENT

The software modules and files with which the operator will find himself directly or indirectly involved are

Monitor

Symbionts

Terminal Job Entry system

Media Conversion Utility

Job Control Processor

Overlay Loader

Permanent disk files

RADEDIT

EDIT

Error Logging file

User Programs (foreground and background)

Language Processors

ELLA Processor

On-line device exercisers

MONITOR

The monitor is the primary control element of the operating system and functions as a supervisor that coordinates and controls a continuous series of foreground and background jobs. It is also the two-way communications link between the operator and the total system. The operator communicates with the system via key-ins which request monitor actions, such as passing messages, triggering interrupts, or running programs. The monitor communicates with the operator via the operator's console (OC device) through error messages, status messages, or requests for operator action such as readying a magnetic tape unit.

Other services the monitor performs includes preserving the relative priority of real-time tasks, protecting the foreground memory and secondary storage from background interference, monitoring available CPU time for background use when real-time interrupts are idle, and providing general I/O services for all tasks.

Parts of the monitor are permanently resident in main memory so that it can respond immediately to a request for service from a real-time task. Response time to such tasks is highly critical and is measured in microseconds. Less critical portions of the monitor such as the Job Control Processor are permanently stored on the Systems area of the disk and brought into main memory as needed.

SYMBIONTS

Symbionts increase the amount of background processing performed in a given amount of time by doing relatively slow card reader and line printer I/O at a time other than when the request is made. Symbiont service is not available to foreground programs.

Most symbionts are automatic and do not require operator intervention in normal operation once they are started. A line printer symbiont is self-starting; a card reader symbiont must be started by the SCRndd, I key-in. Symbionts are CP-R options exercised at SYSGEN time.

TERMINAL JOB ENTRY

TJE allows multiple users of either local Teletypes® attached to an IOP or remote Teletype attached through a COC to log on and use the system interactively. Such users may use EDIT, RADEDIT (a subset), submit batch jobs, and load and debug their load modules concurrently.

Each user is assigned a job name identical to his or her control device (TYxxx or LNxxx). This allows the operator to use all job-oriented key-ins to control the user job in addition to the specific TJE system key-ins. TJE is a CP-R option exercised at SYSGEN time.

MEDIA CONVERSION UTILITY

Media conversion allows the operator to perform a variety of Utility copy functions concurrent with normal real-time and background processing. The functions that are provided are:

File to
- file
- printer
- keyboard printer
- card punch
- magnetic tape

Magnetic Tape to
- printer
- card punch
- magnetic tape (1024 byte records maximum)
- file

Card to
- magnetic tape
- printer
- file
- card punch

The Utility operations are controlled through operator key-ins that are summarized in this manual and are described in more detail in the CP-R Reference Manual, 90 30 85. Media conversion is a CP-R option exercised at SYSGEN time.

JOB CONTROL PROCESSOR

The Job Control Processor (JCP) provides control and utility services for background jobs. The JCP reads and processes control commands input from the Control Command (C) input device. Whenever the JCP encounters a request in the job stack to execute a processor (such as the FORTRAN compiler), or a user program, it causes the processor or program to be loaded into memory from the disk and the JCP to relinquish control. Whenever a processor or program relinquishes control of background, the JCP returns.

The JCP outputs error and status messages on the LL device in contrast to monitor messages that are output on the OC device. Generally, messages output on LL are for programmer reference. All messages of interest to the operator are output on the OC device.

OVERLAY LOADER

The Overlay Loader creates programs in overlay (segmented) form. That is, the Loader converts a program from object module form into an absolute memory image version called a load module for later execution. Segmenting permits programmers to create programs much larger than the available memory size and to use certain memory management functions to optimize real memory allocation.

Like any other processor, the Loader is called in for execution by a control command in a background job. The mnemonics for the control commands used by the Overlay Loader are similar to those used by the monitor but the format of the commands is slightly different (see the Overlay Loader chapter in CP-R RT/BP Reference Manual). The Overlay Loader operates as a background processor and does not interact with the operator.

SECONDARY STORAGE FILES

Since the entire CP-R system is disk oriented, every job run by the system will directly or indirectly involve the use, modification, allocation, or release of permanent disk files. The files of greatest interest to the operator and the mnemonics used to reference them on control commands are as follows:

- Foreground Programs area (FP) contains a collection of foreground programs, an optional User Library, and Public Libraries. Referenced library routines are included in the user load modules at "Load" time. Public Libraries are a group of reentrant routines shared by a number of programs and are called into main memory for execution only when referenced.

- System Programs area (SP) contains the monitor (CP-R), the Account Inventory file, the SCHEDuler file, and (by default) the set of processors used by the local installation, such as AP, FORTRAN, RADEDIT, the Overlay Loader, and EDIT. The area also contains the System Library.

- Background Programs area (BP) is provided to contain the set of permanent user operational programs that execute in the background.

- User areas (any two-character name) allow flexible subdivision of the disk space to aid the individual installation in controlling disk storage use. User disk areas provide both use control and file name uniqueness for the files within them. Use control is determined by the area protection type. This may be public, background, foreground, system, or IOEX. Public and background areas may be read or altered by either background or foreground users. Additionally, public areas provide a pool of space for area-independent file activity. Normally, foreground areas may be altered only by foreground users, and system areas may not be altered. However, both types may be read by any user. IOEX areas may be used only by IOEX (direct device control) services. Files may not be defined in them. File name uniqueness is determined by the combination of file name, disk file account name, and disk area name. This allows several files to be defined with the same name and account, as long as they are in different disk areas. (The data areas, D1 through DF, are considered to be user areas.)

- Background Temp area (BT) contains temporary (scratch) files (X1 through Xn, where n is a SYSGEN parameter) used by background programs for intermediate storage in processing. Their use is identical to scratch tapes on magnetic tape units, and files can be rewound or searched. Note that temp files are erased at the end of each job step by having their pointers reset, unless a SAVE command is present within a job step. Temp files are automatically reset when a new JOB command is encountered in a job stack regardless of SAVE commands, and there is no way to save data on temp files from one job to another.

- The GO and OV files are also in the BT area and are special cases. The GO file contains Relocatable Object Modules (ROMs) formed by a processor if the GO option is specified. The OV file contains the executable program formed by the Overlay Loader if a program file name was not specified at load time.

- The monitor will log hardware and software errors on the ER device. Normally, this will be the ERRFILE file in the SP area, but the ER operational label can be assigned to the card punch or magnetic tape. However, the Error Log Listing and Analysis (ELLA) program will use ERRFILE for the error log data base.

RADEDIT

RADEDIT controls secondary storage allocation for areas containing permanent files and performs utility functions for all areas except the BT area (X1 – Xn, GO, OV).

The operator will encounter a number of jobs involving file manipulation via RADEDIT, including: allotting files, building files, dumping areas or files on user request, copying object modules from libraries, loading new programs into user or system libraries, inhibiting bad disk pack tracks, etc.

File manipulation via RADEDIT is governed by control commands that have a format similar to monitor control commands. RADEDIT operates as a background processor or as an interactive processor in a terminal job (with restricted capabilities). It does not interact directly with the operator.

EDIT

The EDIT processor provides a means of building and updating files of source data such as assembly or compiler language source programs or batch job control command files.

EDIT operates either as a batch processor in the background job stream or as an interactive processor in a terminal job. It does not interact directly with the operator.

LANGUAGE PROCESSORS

The following language processors are available under CP-R any any or all of them may be incorporated in the local system:

Extended FORTRAN IV

SL-1

Assembly Program (AP)

The selected processors are permanently stored in the System Programs area on the disk and called into memory to assemble or compile a user's source program through a control command in a Background control deck. The JCP relinquishes control to the requested processor until its operation is completed.

ERROR LOG LISTING AND ANALYSIS (ELLA)

ELLA may be used for listing and analyzing the error data base generated and updated by the monitor. ELLA furnishes a meaningful and comprehensive diagnostic evaluation of error conditions reported by the system and its related peripherals. An analysis of errors could provide an evaluation of product performance and early detection of product failures, thus increasing the reliability, maintainability, and availability of the operating system and its peripherals.

ON-LINE DEVICE EXERCISER SYSTEM

The On-Line Exerciser system may be used for testing particular sections of the hardware without degrading or changing the integrity of the operating system. Although the On-Line Exerciser system does not diagnose, it may be used to verify that a resource element (such as a peripheral device) is in proper working order without taking the system into an off-line environment.

2. OPERATOR COMMUNICATIONS

Communications between the operator and the system take place through operator key-ins (solicited and unsolicited) and monitor printouts (CP-R messages).

CP-R MESSAGES

CP-R and associated processors (as selected during SYSGEN) output messages on the OC device whenever operator action is required or to inform the operator as to the status of events (including errors) taking place within the system. CP-R output messages are listed and described in Table 1.

TRAP HANDLER MESSAGES

In addition to the messages listed in Table 1, the following messages are output by the trap handler upon occurrence of the various traps if the user does not specify his own trap handling:

!!ARITH. FAULT AT xxxxx ID = xxxxxxxx

!!BREAK ERROR AT xxxxx ID = xxxxxxxx

!!MEM. PARITY ERR AT xxxxx ID = xxxxxxxx

!!MEM. PROT. ERR AT xxxxx ID = xxxxxxxx

!!NONEXIST. ADD. AT xxxxx ID = xxxxxxxx

!!NONEXIST. INST. AT xxxxx ID = xxxxxxxx

!!PRIVILEGE INST. AT xxxxx ID = xxxxxxxx

!!STACK OVERFLOW AT xxxxx ID = xxxxxxxx

!!UNIMPLE. INST. AT xxxxx ID = xxxxxxxx

!!WDOG TIMER RUNOUT AT xxxxx ID = xxxxxxxx

Note that the message "ARITH. FAULT AT xxxxx" is output for the fixed point arithmetic overflow trap, the floating-point fault trap, and the decimal arithmetic fault trap. The message

ERRxx ON CAL AT xxxxx ID = xxxxxxxx

is output if a user program furnishes an invalid parameter while attempting to use a service function. ID identifies the trapped task.

Table 1. CP-R Messages and Responses

Message[†]	Meaning	Operator Action
!!yyndd **** message	"yyndd" is the control device and job name for the terminal job. "Message" is the message sent by the terminal user to the operator.	Action is determined by the content of the terminal user's message.
!!yyndd ATTENTION INTERRUPT	An attention interrupt was received from the specified device.	No action is required.
!!yyndd ERROR[†††]	An irrecoverable error has occurred.	No key-in response required except for card reader. For card reader error, remove last card in output hopper for jam and replace it or a duplicate to input hopper. Key in CRndd RⓃ. If card or reader cannot be fixed, key in CRndd EⓃ to inform the requesting task the card reader errored.

[†] Messages beginning with !!yyndd in the message column will vary according to device type. Therefore the second words in such messages are listed alphabetically.

[††] If background, requires an I/O key-in to continue or retry I/O operation on the device; if foreground, the operation is errored and no key-in is required or expected.

[†††] Key-in may be required depending on device type.

Table 1. CP-R Messages and Responses (cont.)

Message	Meaning	Operator Action
!!yyndd ERROR, NOT OPERATIONAL	Device went not operational during I/O operation.	No action required.
!!yyndd ERROR, POSITION LOST	A magnetic tape unit either returned inconsistent status or the tape position is indeterminant following a tape operation.	No action required.
!!yyndd I/O TIMED OUT[††]	An I/O interrupt failed to return from the device used for an I/O operation in the software timeout period allowed.	Unless the timeout occurred in a foreground task, the message always requires operator action. If the cause and/or correction is unknown, key-in yyndd E(NL). If the cause is known, key in either yyndd R(NL) or yyndd C(NL) to specify whether the I/O operation should be or should not be retried. (R specifies retry; C specifies continue.)
!!yyndd KEY-IN PENDING	An I/O operation is waiting for an operator key-in on the indicated device. This message is repeated at intervals.	Supply the appropriate key-in.
!!yyndd MANUAL	Device was in manual mode at SIO initiation.	Ready the device. No key-in is required.
!!yyndd SIO REJECT, CC = 10 --[††]	SIO instruction returned CC1 and CC2 as nonzero (10).	Correct condition and key in yyndd R(NL). If condition cannot be repaired, key in yyndd E(NL) to inform task the I/O has errored.
!!yyndd SIO REJECT, CC = 01 --[††]	SIO instruction returned CC1 and CC2 as nonzero (01).	Correct condition and key in yyndd R(NL). If condition cannot be repaired, key in yyndd E(NL) to inform task the I/O has errored.
!!yyndd TEST MODE	Device went into test mode during I/O operation.	No action required.
!!yyndd UNRECOGNIZED[††]	SIO instruction returned CC1 and CC2 as nonzero (11).	Correct condition and key-in yyndd R(NL). If condition cannot be repaired, key in yyndd E(NL) to inform task the I/O has errored.
!!yyndd WRITE PROTECTED[††]	An attempt was made to write to a write protected device or disk track.	If the write is to be permitted then change the write protected status of the device or track and key in yyndd R(RET). If the write is not to be permitted, key in yyndd E(RET).
!!ALARM xxxxxxxxxx	A system inconsistency was detected, following the ALARM message. The reason is indicated in the text.	See Chapter 5, "Availability Operating Procedures".
!!BACKGROUND IDLE	Background sequencing has been terminated because JCP read a !FIN command or encountered a critical error.	If more background jobs are to be run, key in C to restart background sequencing.

Table 1. CP-R Messages and Responses (cont.)

Message	Meaning	Operator Action
!!BACKGROUND WAIT	Background has executed a "WAIT" request.	Key in C (NL) to continue background processing.
!!BKG ABORT WAIT	Background has aborted in attended mode.	Key in "C" to continue, or "DB" to dump background memory.
!!BKGD JOB ident ON	Specified background symbiont job has been selected for execution.	No operator action required.
!!CANT OPEN ERRORLOG	The error log file was already in use by error list program, full RFT, no blocking buffer available, or a DED key-in was in effect.	See Chapter 2 Availability Manual.
!!CORE SAVED	Memory has been saved on the CK area following a system fault.	No action required.
!!CORE USED, CAN'T LOAD xxxxxxxx	The specified primary program cannot be loaded for execution because its required core space is already in use.	Key-in X (NL) to abort the job.
!!CP-R RESTARTED	CP-R is fully restarted following a system ALARM and an auto-restart. The date and time information have been maintained and need not be re-entered.	No action required.
!!CRndd CARD NOT FED[††]	The card reader was unable to feed a card correctly.	Correct or replace the card in the read hopper. Push RESET START and key in CRndd R(RET) to retry the operation.
!!DPndd IDLE	Indicated unit has no open files.	May be removed from the spindle.
!!ERRORLOG ERROR	An irrecoverable write error was encountered while attempting to write entries to the ER operational label.	Reassign ER oplabel to another output device or "0".
!!ERRORLOG FULL	While attempting to write entries to the ER operational label, an end-of-file, end-of-data, or end-of-tape was encountered.	See Chapter 2 for Error Log purging procedures, Availability Manual.
!!FILE NAME ERR	A problem has occurred in attempting to open or close a disk file through an STDLB key-in.	Key-in X (NL) to abort the job.
!!I/O ERR, CAN'T LOAD xxxxxxxx	An I/O error occurred or no blocking buffer was available when attempting to load the specified foreground program for execution.	Key-in X (NL) to abort the job.
!!JOB account, name, priority, ident	An input job was added to a list of symbiont files. Message has some format as user's job card except for ident, which is added to the end.	No action required.
!!KEY ERR	CP-R cannot recognize an unsolicited key-in response.	Retry the key-in.

Table 1. CP-R Messages and Responses (cont.)

Message	Meaning	Operator Action
!!LMI FULL, CAN'T LOAD xxxxxxxx	The specified primary program cannot be loaded for execution because no room exists in the Load Module Inventory.	Key-in X NL to abort the job.
!!LOADED PROG xxxxxxxx	Specified primary programs were loaded by Foreground Loader for execution. Up to three program names will be output in one message.	No action required. Outputting of this message to OC may be prevented by setting appropriate assembly time software switch to bypass the code causing the message.
!!MEDIA ABORTED REQ xxxx:yyyy	MEDIA request xxxx was aborted due to reason yyyy where yyyy can be OPER - operator keyed in "X" NOMO - the MO oplabel is not defined OPNI - unable to open input file's DCB OPNO - unable to open output file's DCB PREP - an irrecoverable error occurred during preprocessing BUFS - insufficient job reserved pages for blocking buffers DEV - a fatal error occurred during the copy SPEC - invalid specifications were detected	None. If request xxxx was initiated by an operator or MEDIA key-in and the cause of the abortion removed, the key-in can be reentered.
!!MEDIA MOUNT TAPES FOR xxxx	MEDIA request xxxx is ready to begin.	Mount tape or tapes required. When ready, key-in MEDIA I to initiate copy.
!!MEMORY FAULT	During a memory status scan, memory status was found which indicated that an error was detected by the memory unit. An error log entry was recorded for this problem.	No action required.
!!NO PATCH AREA —CLEAR B $ IGNORES MODIFY COMMAND	If this is OC, only the message portion preceding the hyphen is output. If not OC, the entire message is output.	If this is OC, input next command. If not OC, idle machine, increment address, and RUN.
!!NO TSPACE, CAN'T LOAD xxxxxxxx	A primary program load request could not be satisfied because TSPACE was not available for building FPTs or for reading in the header.	Key in X NL to abort the job.
!!NONEXIST., CAN'T LOAD xxxxxxxx	Specified program cannot be loaded for execution because it does not exist on disk, or required Public Library does not exist on disk.	Retry. If message is output again, key-in X NL to abort the job.

Table 1. CP-R Messages and Responses (cont.)

Message	Meaning	Operator Action
!!PATCH LOC ERR-CORRECT AND CLEAR B $	If this is OC, only the message portion preceding the hyphen is output. If not OC, the entire message is output.	If this is OC, input next command. If not OC, idle machine, increment address, and RUN.
!!PAUSE commands	A !PAUSE command has been read. Comments field will have operator instruction.	Press INTERRUPT switch and key in C Ⓝ to continue reading from the job stack after performing required action.
!!PLEASE KEY-IN DATE-TIME	CP-R has been booted in and requires the date and time before other operations are allowed.	Enter DT ⌴ mm, dd, yy, hh, mm. where mm - month (1-12) dd - day (1-31) yy - year (00-99) hh - hour (0-23) min - minute (0-59)
!!PROCESSOR FAULT	During a processor status scan, processor status was found which indicated that an error was detected by a processor. An error log entry was recorded for this problem.	No action required.
!!PUB LIB, CAN'T LOAD xxxxxxxx	Invalid request to load Public Library for execution. All Public Libraries must be automatically loaded by the system as needed.	Either retry the job or key in X Ⓝ to abort the job.
!!QUEUED AS NUM. xxxx	MEDIA copy request is accepted and is assigned ident number xxxx.	No action required.
!!RELEASED PROG xxxxxxxx	Specified primary program was released.	No action required. Message can be prevented by setting appropriate assembly time software switch to bypass code causing the message.
!!RLS NAME NA	Key-in request to release a foreground program whose name is not recognized by the system.	Retry.
!!SPURIOUS EXTERNAL INTERRUPT	An interrupt has been triggered but is not connected to a task.	No action is required.
!!Syyndd ERR xx	An irrecoverable error occurred. Symbiont activity on specified device was terminated.	No action required.
!!Syyndd IDLE	Specified symbiont device entered idle state.	No action required.
!!Syyndd UNAVAILABLE	Specified device is currently unavailable to the symbiont.	No action required.
!!TERMINAL JOB yyndd account, name OFF	Indicates the terminal user is now inactive. "yyndd" is the control device and job name for terminal job. "Account, name" is logon verified user account, name.	No action required.

Table 1. CP-R Messages and Responses (cont.)

Message	Meaning	Operator Action
!!TERMINAL JOB yyndd account, name ON	Indicates the terminal user is now active. "yyndd" is the control device and job name for terminal job. "Account, name" is logon verified user account and user name.	No action required.
!!UNABLE TO CLOSE DCB xxxxxxxx	The specified DCB was not closed in the task during termination.	No action required.
!!XEROX CP-R VERSION xxxx	Output whenever system is booted in.	Message can be terminated by hitting BREAK key on OC.

OUTPUT MESSAGE FORMATS

Output messages are printed in three different formats on the Operator's Console. These formats are outlined below.

1. Messages without tabs are for operator's reference at a later time and are typically generated by foreground or background programs that wish to communicate with the operator. They are similar to system messages but have no time stamp.

2. System messages which begin with two exclamation marks (!!) will be preceded by a time stamp (hh:mm) in column 1 through 5, followed by a single tab. With normal tab settings the message will be positioned 1 inch to the right of the left margin.

3. Device control messages will have a time stamp in column 1 through 5, followed by four tabs. With normal tab settings the message will be positioned 4 inches to the right of the left margin.

OPERATOR KEY-IN

After the system has been initialized, operator key-ins permit the operator to control the execution of tasks by the system (i.e., foreground tasks, background job stream, and Control Task services).

With the exception of the DT key-in (which is requested by the system at the end of system initialization but prior to any other activity, if job accounting or error-logging has been requested), operator key-ins are unsolicited. That is, an operator key-in is initiated by the operator depressing the INTERRUPT switch on the Control Panel. (On a Xerox 550, the ATTENTION key may be substituted.) This action activates the Control Panel Task which, in turn, triggers the CP-R Control Task. When the Control Task becomes the highest priority task in the system (that is, when all foreground tasks are inactive), the system issues a prompt character (a dash, "-") to the Operator's Console and waits for operator input.

Via the OC device, the operator inputs appropriate information in the following sequence:

1. Optionally, types in an exclamation mark (!).

2. Types in the desired key-in and any associated parameters. (Refer to Tables 2 through 5 for listing and description of Standard, Terminal Job Entry, Symbiont, and Media Conversion key-ins.)

3. Types in a New Line character (NL) to indicate the completion of a key-in.

If a typing error is discovered before the (NL) character has been keyed in, the error may be corrected by the operator in one of the following ways:

1. If the erroneous character is the last character typed, key in one cent sign (¢) character, retype the last character correctly, and continue to type in the remainder of the key-in. Indicate completion of corrected key-in by keying in the (NL) character.

2. If the erroneous character is within a few positions of the last character typed, key in an appropriate number of cent signs (starting with the last character, each cent sign deletes one character and performs a simulated backspace of one position) until the erroneous character has been effectively deleted. Retype correctly, all characters deleted and resume typing rest of key-in. Upon completion of key-in, key in the (NL) character.

3. If the erroneous character is located toward the beginning of the key-in or many positions from the last character typed, it may be more expeditious to key in an EOM (End of Message) character. In this case, the entire key-in is deleted and the monitor is ready for a new key-in.

Table 2. Standard Operator Key-Ins

Key-In	Purpose
yyndd a (NL)	Controls system action following an abnormal condition during an I/O operation, where "yyndd" is the physical device name of the device involved, and "a" is a single character that requests a system action relative to the device as follows: a = C – continue "as is"; a = E – inform user program of the error and transmit record "as is"; a = R – retry the I/O operation; a = X – unconditionally abort the current operation on the device and error the request.
ALARM (NL)	Forces an operator-initiated system alarm. See also REBOOT key-in.
ATT $\left[\begin{Bmatrix}\text{OFF}\\ \text{ON}\end{Bmatrix}\right]$ (NL)	If OFF, the batch control command !ATTEND becomes illegal. If ON, both the !PAUSE and !ATTEND control commands become legal. This command allows the system to permit or reject batch jobs that depend upon the programmer being able to interact with his job. Default is ON. See also PAU key-in.
BMEM[n] (NL)	Change memory allocation for the background job. The "n" specifies the number of pages to be allocated to the background job. If "n" is not specified, its background job memory is restored to its SYSGEN-defined value. Allocations for background tasks using simplified memory management (SMM option on the !OLOAD command) will not take effect until the current background job step terminates.
BREAK jobname (NL)	Transfers control to user's break-receiver routine if the user established break control in the job via an INT service call. jobname is the name of the job to receive the break signal.
C [S/O] (NL)	If no parameters: Continue processing in the background. If the background was in a wait or idle state, the system leaves that state and proceeds. If "S" parameter: Change the background control mode to symbiont, so the symbiont can start background when necessary. If "O" parameter: Change the background control mode to operator, so only the operator may start background. The symbiont can only start background from an idle state. It can not clear a background wait.
CC (NL)	Retransfer control back to the C device from OC. Used in conjunction with TY key-in.
CINT $\begin{Bmatrix}\text{location}\\ \text{label}\end{Bmatrix}, \begin{Bmatrix}\text{D}\\ \text{A}\\ \text{T}\end{Bmatrix}$ (NL)	Disarm, arm and enable, or trigger specified interrupt. The "location" specifies the hex address of the interrupt; "label" specifies an interrupt label; "D" is used to disarm specified interrupt; "A" is used to arm and enable; "T" is used to arm, enable, and trigger the interrupt.
CKD $\begin{Bmatrix}\text{loaddr}\\ \text{loaddr,hiaddr}\end{Bmatrix}$ (NL)	Selectively dump system output saved in the CK area after a system alarm. The "loaddr" parameter specifies the lowest address to dump. The "hiaddr" parameter specifies the highest address to dump. Default cases for "loaddr" and "hiaddr" are 0 and high core respectively. Activation of the Control Panel Interrupt during the MAP portion of the dump stops map output and starts the DUMP portion. Activation of the interrupt during the DUMP portion terminates the dump.
COC (NL)	Job was halted because of error in control command. Continue from OC with correct control command (after depressing the Control Panel Interrupt Key).

Table 2. Standard Operator Key-Ins (cont.)

Key-In	Purpose
CRD $\left\{\begin{matrix}\text{loaddr}\\ \text{loaddr,hiaddr}\end{matrix}\right\}$ NL	Identical to CKD key-in above except that the dump is read from the SE oplabel rather than the CK area (see CRS key-in below).
CRS NL	Preserve an alarm dump by copying it from the CK area to the SE oplabel in 1024-byte records.
DEBUG taskname[,(JOB,jobname)] NL	Causes the named task in the named job to be run under DEBUG control.
$\left\{\begin{matrix}\text{DED}\\ \text{UND}\end{matrix}\right\}$ yyndd, $\left\{\begin{matrix}\left\{\begin{matrix}F\\ X\\ N\\ D\end{matrix}\right\}\left[\left[,\left\{\begin{matrix}I\\ D\end{matrix}\right\}\right]\right]\\ R\end{matrix}\right\}$ NL	DED dedicates a device, device controller, or IOP. UND undedicates a previously dedicated device, device controller, or IOP. F defines the device to be dedicated to the foreground; X prevents device use and aborts any existing requests; N prevents device use but does not abort existing requests for use; D restricts a device for diagnostic use and aborts any existing requests; R specifies that the disk pack will be removed (DED) or has been replaced (UND) from unit DPndd; 1 dedicates or undedicates all devices on IOP n (of yyndd); D dedicates or undedicates only device yyndd. If neither 1 nor D is specified, all devices on the same multiunit controller are dedicated or undedicated.
DISPLAY[LL] NL	Display current status of all tasks in the system on the LL or OC device. Output is to the OC device unless LL is specified. A typical display output is shown in Figure 1.
$\left\{\begin{matrix}\text{DM}\\ \text{DF}\\ \text{DB}\end{matrix}\right\}$[from,to][,T][,S] NL	Dumps the contents of specified memory onto the device that is permanently assigned to the DO oplabel. DM specifies a real-memory dump with default boundaries being zero and the end of the monitor. DF specifies a real-memory dump with default boundaries being the limits of the first Foreground Private Memory partition. DB specifies a virtual-memory postmortem dump of background with default boundaries as the limits of Task Virtual Memory. DF or DM requests are performed immediately. A DB dump request is not performed until the next task termination occurs in background (either for JCP or any other processor or user program). If "from, to" is absent, the entire default area will be dumped; if present, the first word address in hex and last word address in hex of the selected area are defined. If ",T" is absent, core memory is dumped in hexadecimal; if present, core memory is dumped in hexadecimal and EBCDIC. 'S' specifies the dump will be in a short-line format at four words per line.
DT mo, day, yr, hr, min NL	Input of current data and time. Example: DT 8, 17, 69, 22, 30.
ELOG $\left\{\begin{matrix}\text{ON}\\ \text{OFF}\\ \text{PURGE}\end{matrix}\right\}$ NL	Turns error logging procedures on and off. Eliminates the majority of the execution time overhead associated with error logging but does not prevent gathering of the error statistics for the ESUM display. PURGE clears the error log and all error, log and I/O counts.
$\left\{\begin{matrix}\text{ERRSEND}\\ \text{ESEND}\end{matrix}\right\}$ text NL	Creates an error log entry containing the supplied text (56-byte maximum).
ESUM[LL] NL	Display a device error summary on OC or LL. A typical error summary is shown in Figure 2.

Table 2. Standard Operator Key-Ins (cont.)

Key-In	Purpose
EXTM taskname[,(JOB,jobname)] (NL)	Terminate a task in a job. The "taskname" is the name of the task to be terminated. "JOB" is a keyword that indicates that the job under which the task is running will be specified. (If JOB is absent, jobname defaults to the CP-R job.) The "jobname" is the name of the job under which the task to be terminated is running.
FG (NL)	Permit loading of foreground program from background job stack for execution via a !RUN or !INIT control command.
INIT fid[, (JOB, jobname)][, PRI] [,STOP][, (PRIO, xxxx)] [, DEBUG][, TS] (NL)	Read named task into memory and initiate it. The fid is the file identifier for the load module file. "JOB" is a keyword that indicates that the task is to run under a job other than the CP-R job (if "JOB" is not specified, the task runs under the CP-R job). The "jobname" specifies the name of the job under which the task is to be run. PRI specifies the task is primary and if not specified the task is run as secondary. STOP is an option that specifies the task (secondary only) is to be left in suspended state after load. The default is to execute directly after load. PRIO is a keyword to specify task priority (for secondary tasks only). Default is to run under lowest-priority dispatcher at lowest software priority. This is X'EFEF' for nontimesliced tasks or X'FFFF' for timesliced tasks. xxxx is the task's priority value (in hex). The first 2 characters specify the interrupt level minus X'4F' of the task's dispatcher, and the last 2 characters specify the task's software priority. DEBUG indicates the task is to be initially under Debug control. TS indicates that the task is to be time-sliced.
INTLB label, loc (hex) (NL)	Change assignment of interrupt labels.
KJOB jobname (NL)	Terminates all tasks in the named job.
PAU [{ON OFF}] (NL)	If OFF, both the !PAUSE and !ATTEND batch control commands become illegal and any I/O operation which would otherwise have made a request for an operator's key-in to correct an I/O problem is aborted with a TYC4. If ON, only the !PAUSE command becomes legal. This command allows the system to permit or reject batch jobs that depend upon operator interaction (e.g., tape mounting) to continue. Default is ON. See also ATT key-in.
Q [{(SLICE, n) (QMIN, t) (QSWA, t) (QMAX, t)}, ...] (NL)	If no parameters are used, the current settings of the time-slicing controls are reported. These parameters are number of slices per second; and slice length, QMIN, QSWA, and QMAX in milliseconds. If parameters are specified, the corresponding controls are set as specified. The parameter "n" is a decimal number of slices per second that must be between 1 and 500, and must divide 500 evenly. The parameter "t" is milliseconds (decimal) and must be 1 or more.
Q30 {NORETRY RETRY NOPARITY PARITY NOWDT WDT} [, ...] (NL)	This command is available on a Xerox 550 system only. It affects the contents of the Q30 register as follows: NORETRY: set bit 0 No instruction retry RETRY: reset bit 0 NOPARITY: set bit 1 No parity checks PARITY: reset bit 1 NOWDT: set bit 2 No watchdog timer WDT: reset bit 2 Default is no change for each bit.

Table 2. Standard Operator Key-Ins (cont.)

Key-In	Purpose
Q31 [option, . . .] (NL)	This command allows simple examination and modification of the address stop controls (Q31 register) of a Xerox 550 system. It is not available on a Sigma 9 system. The "options" are any of the following: (ADDR, x) Set address to "x" (hexadecimal). (Symbol, x) Set address to "x" plus the value of "symbol". "Symbol" may be any symbol found in the symbol table in module CRS2 or any overlay name. (STM, x) Set address to "x" plus X'6000'. (ROOT, x) Set address to "x" plus X'600E'. INST Compare execution access addresses. WRITE Compare write access addresses. ANY Compare any addresses accessed. ALL (Same as ANY.) PAGE Compare page addresses. WORD Compare word addresses. REAL Compare real addresses. VIRT Compare virtual addresses. STOP Stop execution on compare. HALT (Same as STOP.) BEEP Briefly sound audio alarm on compare. CLEAR Reset all flag bits (but save the address). Any bit or field not specified is not changed. If no options are specified, the Q31 register content is typed as a hexadecimal number.
REBOOT (NL)	Forces an operator-initiated system alarm and an automatic reboot of the system. See also ALARM key-in.
RLS taskname (NL)	Terminate a foreground primary program running under the CP-R job.
RSY [jobname]	Removes SY key-in for the specified job, returning to normal software disk write protection. Default for 'jobname' is 'BKG' which is the background job. Note that the SY key-in is automatically removed from background whenever a !JOB or !FIN command is processed.
RUN taskname[, priority] (NL)	Load and execute a foreground program running under the CP-R job. Only primary tasks can be loaded with this key-in. The name of the foreground file to be loaded must be input.
SCHED fid [, (JOB,jobname)] [, PRI] [, (PRIO, xxxx)] [, (STRT,time)] [, (INTV,interval)] [, DELE[TE]] [, TS]	Schedule a task for periodic INIT. The "fid" is a CP-R file identifier with one variation: the file name is actually a task name that may be the same as the file name or related to it by a SETNAME CAL. If neither an account nor an area is specified, the defaults are area FP and the system account. If an area name is specified, the default is the system account. Specification of a whole area is an error. JOB is a keyword indicating that the task is to be INITed under a job other than the CP-R job. The default is the CP-R job. PRI specifies a primary task; the default is secondary. PRIO is a keyword to specify task priority for secondary tasks. Default is to run under lowest-priority dispatcher at lowest software priority. "XXXX" is the task's priority value in hex. The first two characters specify the interrupt level minus X'4F' of the task's dispatcher, and the last two characters specify the software priority. TS specifies a time-sliced task. DELETE specifies that the task is to be de-scheduled. STRT is a keyword indicating that the time for the first INIT to the task is specified. A value of zero or absence of the keyword causes an immediate INIT.

Table 2. Standard Operator Key-Ins (cont.)

Key-In	Purpose
SCHED fid (cont.)	The format of the time values for start time is as follows: [[[[[[yr,]mm,]dd,]hh,]min,]sec] where yr means year (e.g., 76) mm means month number (e.g., 9) dd means day (e.g., 5) hh means hour (e.g., 14) min means minute (e.g., 30) sec means second (e.g., 10) Values are optionally deletable from left to right; e.g., any value may be omitted provided that all parameters to its left are also omitted. Thus, if hh is omitted, yr, mm, and dd must also be omitted. The SCHED key-in uses current date/time values for omitted values. An example of the STRT option is: ...(STRT, 14, 30, 0)... which would cause the specified task to be INITed at 2:30 p.m. of the current year, month, and day. Please note that omission of the 0 (seconds) value would cause INIT at 14 minutes and 30 seconds of the current year, month, day, and hour. INTV is a keyword indicating that the period between INITs is specified in seconds. If the specified value is not an integral multiple of five, it is rounded up to the next highest integral multiple of five. A value of zero or absence of the parameter causes the INIT to be issued once at the specified STRT time. The presence or absence of the STRT and INTV keywords is interpreted as follows: INTV STRT Result present absent periodic INIT, starting at now + INTV present present periodic INIT, starting at STRT absent absent INIT once immediately absent present INIT once at STRT
SJOB jobname[, (DEBUG, TYndd)] (NL) [,(ACCT, xxx)]	Creates the named foreground job by setting up job controls and table entries but does not initiate any task in the job. DEBUG indicates a Debug control-console device is specified. TYndd is the address used by Debug for communication with the user. ACCT indicates an account is specified. xxx is the account the named JOB is to be associated with.
SNAP[FILE, fid](NL)	Saves core (the monitor) on the specified file or to the SE oplabel by default.
STAT taskname[, (JOB, jobname)]	Output the status of the specified job on OC. The "taskname" defines the name of the task from which status information is desired. The "JOB" is a keyword that indicates that the job under which the task is running will be specified. (If "JOB" is not present, jobname is defaulted to the CP-R job.) The "jobname" is the name of the job under which the task is running. The task status will have the format illustrated in Figure 3.

Table 2. Standard Operator Key-Ins (cont.)

Key-In	Purpose
STDLB label, $\left\{\begin{matrix}\text{fid}\\\text{device}\\\text{oplabel}\end{matrix}\right\}$ (NL)	Changes a current oplabel assignment. The new assignment will stay in effect until changed by another STDLB key-in or system is rebooted. "label" specifies the oplabel to be assigned, which must have been previously assigned at SYSGEN. "fid" specifies a disk file or disk area. "device" specifies a physical device name, or "0" for the null device. "oplabel" specifies one of the SYSGEN-defined oplabels.
SY[jobname] (NL)	Overrides normal software disk file write protection for the specified job. The default for 'jobname' is 'BKG' which is the background job. The SY privilege is removed by the RSY key in or job termination. For background, it is also removed automatically when a !JOB or !FIN command is processed.
TY (NL)	Transfer control from the C device to OC (typewriter) for reading control commands.
W (NL)	Suspend current background job and enter WAIT state.
X (NL)	Abort current background job. Message on OC and LL will show last location executed.
(EOM)	Delete this line. (On a Xerox 550, the combination "CONTROL" and "X" may be substituted.)
(NL)	Ignore operator key-in request.
¢	Delete last character. (On a Xerox 550, a "\" may be substituted.)

Table 3. Terminal Job Entry Key-Ins

Key-In	Purpose
CONTROL yyndd (NL)	Causes a control interrupt (equivalent of Yc) to take place for the named terminal job. It is used for jobs previously created by the LOGON key-in. Active terminal jobs will have been logged on OC as they become active.
LOGON TYndd (NL)	Causes device TYndd to be processed by the logon processor and so perform as a normal TJE user.
OFF (NL)	Prevents any new on-line user from logging on. An OFF key-in and an ON 0 key-in are equivalent.
ON$\left\{\begin{matrix}\text{ALL}\\\text{n}\end{matrix}\right\}$ (NL)	Specifies the number of on-line users allowed on the system at any one time. When n users are on, no additional users are allowed to log on until a current user logs off. All makes all lines available.
SEND$\left\{\begin{matrix}\text{yyndd}\\\text{ALL}\end{matrix}\right\}$, message (NL)	Causes the text message to be sent to the TJE terminal specified by yyndd, or sent to all active terminal users if the ALL option is specified.

Table 4. Symbiont Key-Ins

Key-In	Purpose
BATCH fid (NL)	Places the specified file on the symbiont input queue. If background control is in symbiont mode (see 'C' key-in), background is started. A batch file may contain only one background job.
DELETE ident (NL)	Deletes symbiont files. All input and output symbiont files associated with the specified job ID will be deleted. If the specified job is still active, the DELETE key in has no effect.
DO (NL)	Sets a switch to cause the symbiont system to delete each one of a job's files from the OS area as soon as it has been output. When the DO key-in is in effect, the R and B options of the Syyndd key-in have the same effect as the C option. If DO has not been keyed in, the switch is set such that all a job's output files are deleted when its last file has been output.
PRIORITY ident, priority (NL)	Changes the priority of a job in the symbiont area, where "ident" is the job ID and "priority" is the new priority to be associated with the specified job. Priorities are expressed as hexadecimal values from 0 through 7, where 7 is the highest priority. A priority of 0 inhibits selection of a job for execution in the IS area and prevents output from a job waiting in the OS area.
RDO (NL)	Used in conjunction with the DO key-in and causes deletion of a job's files in the OS area to occur when the job's last file has been output. See DO key-in.
SS (NL)	Initiates symbiont input when only one.symbiont input device exists. The key-in is not allowed if more than one symbiont input device exists. If background control is in symbiont mode (see 'C' key-in), each time symbiont input defines a new job, the symbiont insures that background is started.
Syyndd, option. (NL)	This symbiont key-in gives the operator control of the symbionts, where "yyndd" is the physical address of a symbiont device and "option" specifies the action to be taken and may be one of the following: I — initiate symbiont I/O on the specified device. Output symbionts do not require this key-in as they are self starting unless an "L" or "T" is in effect. S — suspend symbiont activity for the specified device. C — continue symbiont activity for a previously suspended device. B[,n] — continue symbiont activity for a previously suspended device. Before the output is continued, the output file is backspaced n line printer pages. The default will be one line printer page (a line printer page is approximately 37 records). If the device is not a line printer or if the DO key-in is in effect, B has the same effect as C. R — restart symbiont activity for a previously suspended device. Symbiont activity will start from the beginning as if it had not been suspended. If the DO key-in is in effect or if this is an input symbiont, R has the same effect as C. L — lock out the symbiont from future activity after this file. After completing the current file, the symbiont terminates. An input symbiont will terminate when the next !JOB or !FIN card is read. An Syyndd, I key-in is required to restart symbiont activity on the device.

Table 4. Symbiont Key-Ins (cont.)

Key-In	Purpose
Syyndd, option (cont.)	T has the same effect as the "L" option except the device is removed from the symbiont pool if it was not dedicated to symbionts at SYSGEN. Q save the current output file and terminate. What remains of the file is returned to the output queue and the symbiont is locked immediately. The entire file is saved if the symbiont is not in DO mode. When the system assigns a new symbiont output device to the file, the output operation is continued from where it was stopped. Q is useful in moving a file from a down device to one that is working. X release the current job file and begin processing the next job file. If background control is in symbiont mode (see C key-in), each time symbiont input defines a new job, the symbiont insures that background is started.

Table 5. Media Conversion Key-Ins

Key-In	Purpose
$\text{MEDIA}\left\{\begin{matrix}\text{I}\\\text{L}\\\text{S}\\\text{X}\end{matrix}\right\}$ Ⓝ	Either to control the operation of the MEDIA task or to specify a MEDIA copy of an input source to an output destination, where control operations are: I = initiate the MEDIA task or resume operation if it has been stopped. L = prevent the start of any new copy operations after the completion of the current operation. S = suspend the current operation. X = abort the current operation and do not post processing.
$\text{MEDIA}\ (\left\{\begin{matrix}\text{FILE, fid}\\\text{IN,}\left\{\begin{matrix}\text{op}\\\text{yyndd}\end{matrix}\right\}\end{matrix}\right\})$ $\left[,(\left\{\begin{matrix}\text{SFILE, n}\\\text{ALL}\\\text{DEL}\\\text{REW}\\\text{UNLOAD}\end{matrix}\right\}),\ldots\right]$ $,(\left\{\begin{matrix}\text{FILE, fid}\\\text{OUT,}\left\{\begin{matrix}\text{op}\\\text{yyndd}\end{matrix}\right\}\end{matrix}\right\})$ $\left[,(\left\{\begin{matrix}\text{NVFC}\\\text{SPACE, n}\\\text{ADD}\\\text{SFILE, n}\\\text{WEOF, n}\\\text{REW}\\\text{UNLOAD}\end{matrix}\right\}),\ldots\right]$ Ⓝ	Copy operations request MEDIA to copy a file on a disk tape, or cards to another file, which may be on a disk, card punch, tape, printer, or, if the input is on a disk, a keyboard printer. One or more options may be given to specify pre- and post-processing of the input and output mediums, where SFILE, n skips a tape n files forward before the copy begins. ALL continues copying successive files until a double end-of-file is found. DEL deletes a disk file after the copy. REW and UNLOAD rewinds (unloads) a tape after the copy. NVFC inhibits use of the first byte in each record as a VFC byte in printed files. SPACE, n causes n lines to be skipped between each printed line when NVFC is given. ADD causes the input file(s) to be added to the end of a tape already containing files. WEOF, n writes n EOTs on the output tape after the copy.

The display has the following format:

```
PRI  TASK      TASK TASK   JOB    TASK  TASK
SEC  NAME      PRIO STAT   NAME   FWA   LWA
 S CTRLTASK   FEFE  10   CPR     00000 03E4D
 S MMEXEC     FFFF  20   CPR     06000 1DFFF
 S BKG        FFFF  20   BKG     06000 1DFFF
 P CTRLTASK   FEFE  80   CPR     00000 03E4D
```

where

P indicates a primary task.

S indicates a secondary task.

TASK PRIO is the hexadecimal priority value associated with the task.

TASK STAT is a representation of the task's status, as follows:

80 – task is primary
40 – task is rolled out
20 – task is stopped
10 – task is in execution
08 – task is in initialization
04 – task is suspended
02 – task is time-sliced
01 – task is swapped
00 – task is executable

FWA and LWA stand for first-word address and last-word address respectively.

Figure 1. Display Format

The analysis and subsequent action from an operator's key-in is performed at the Control Task priority level. If the operator key-in is not recognized as a valid input, the following message is output on the OC.

!!KEY ERR

In which case, the operator should retype the input correctly. Note that if the typewriter is busy at the time of the Control Panel Interrupt (i.e., waiting for an input to complete), the operator must complete the input before the system will output the prompt character.

COMBINED KEY-INS

To expedite operator key-ins, the following combinations of key-ins are recognized:

Combined Form	Result
FGC	Execute FG and C key-ins.
SYC	Execute SY and C key-ins.
SFC, FSC	Execute FG, SY, and C key-ins.
TYC	Execute TY and C key-ins.

```
09:23     OCT 25,'73
YYNDD     MDL#     ACCESSES      ERRORS      ERR/1000

TYA01     7012           76           0             0
LPA02     7445         1037           0             0
CRA03     7140          110           0             0
CPA04     7160           13           0             0
9TA80     7322         1760           0             0
9TA81     7322            0           0             0
9TA82     7322          273           6            21
9TA83     7322            0           0             0
9TAD1     7333            0           0             0
9TAD2     7333            0           0             0
7TAE0     7372            0           0             0
7TAE1     7372            0           0             0
DPDF0     7242        25752           0             0
DPDF1     7242            0           0             0
DPDF2     7242            0           0             0
DPDF3     7242            0           0             0
DPBE4     7275            0           0             0
DCBF0     7212            0           0             0
DCCF0     7232         1651           0             0
DCCF1     7232            0           0             0
DCCF2     7232            0           0             0

    6 FILED LOGS,        0 LOGS LOST
```

where

YYNDD and MDL# correspond to parameters defined on the :DEVICE control commands input at SYSGEN.

ACCESSES is the number of SIOs issued for each job.

ERRORS is the number of error retries and error completions for each device.

ERR/1000 is the error rate computed as follows:

ERR/1000 = (ERRORS*1000)/ACCESSES if ACCESSES > 0.

ERR/1000 = 0 if ACCESSES = 0.

FILED LOGS is the total number of error log entries that have been successfully filed.

LOGS LOST is the number of log entries lost because error log filing could not take place.

Figure 2. Error Summary Example

```
STATUS xxxxxxxxxxxxxxxx PRIORITY     xxxx
            (binary)              (hexadecimal)
```

where binary status bits are as follows:

Bits	Value	Meaning
0	1	Task in final termination.
1	1	Task connected to CAL2.
2	1	Task connected to CAL3.
3	1	Task connected to CAL4.
4	1	Background task.
5	1	Secondary task.
6	1	Task being aborted.
7	0	Task initiated via RUN.
	1	Task initiated via INIT.
8	1	Load to be performed.
9	1	Public Library used by primary tasks.
10	1	Public Library used by secondary tasks.
11	1	Release to be performed.
12	1	TEL control requested.
13	1	Task is loaded.
14	1	Task is run queued.
15	1	Break control requested.

Hexadecimal priority characters are as follows: First two hexadecimal characters correspond to interrupt level minus X'4F'. For secondary tasks this is the level of the dispatcher for the task. Last two hexadecimal characters are the software priority of a secondary task, or zero for a primary task.

Figure 3. Task Status Format

DEVICE CONTROL

If the system encounters an abnormal condition during an I/O operation, a pertinent message to the operator is output on the OC device. Such a message is of the form

```
!! name message
```

where

name is the physical device name, yyndd, or the disk file name.

message is the message string informing the operator of the specific condition that has been detected; for example:

ERROR (error was detected on operation)

or

MANUAL (device not ready)

I/O messages are discussed below, grouped according to the type of device to which they apply.

I/O KEY-IN FORMAT

After correcting the abnormal conditions, the operator responds by means of a key-in. The format for an I/O key-in is

yyndd a (NL)

where

yyndd is the physical device name of the device involved in the I/O operation.

a is a single character that requests a system action relative to the device, as follows:

C	Continue "as is".
E	Inform the user program of the error and transmit record "as is".
R	Retry the I/O operation.
X	Abort the pending I/O.

(NL) is the NEW LINE code.

CARD READER MESSAGES/KEY-INS

If the card reader fails to read properly, or if a validity error occurs, one of the following messages is issued:

!!CRndd ERROR

!!CRndd CARD NOT FED

A FAULT indicates that the error condition occurred prior to any data being transferred; an ERROR indicates that at least one byte was read. After correcting the condition, the operator responds with an I/O key-in message. The action character selected depends on the circumstances causing the error condition.

If a feed check error or a power failure occurs, CP-R outputs one of the following messages (depending on where in the read cycle the error took place):

!!CRndd ERROR

!!CRndd CARD NOT FED

!!CRndd TIMED OUT

If the card in the hopper is damaged, the operator replaces it with a duplicate, presses the RESET button on the card reader, and responds with one of the following key-ins:

CRndd R (NL)

CRndd C (NL)

In the event of a power failure, the operator presses the RESET button on the card reader and responds with the key-in:

CRndd R (NL)

If the card stacker is full, if the hopper is empty, or if the device is in the manual mode, the following message is issued:

!!CRndd MANUAL

The operator corrects the condition and then presses the START button on the card reader.

CARD PUNCH MESSAGES/KEY-INS

Instead of outputting an error message when a punch error is first detected, the I/O handler attempts to punch a card x times (x = NRT, a DCB parameter specified by the user) before the following message is issued:

!!CPndd ERROR

This message indicates that the card punch is not functioning properly and the operator should reevaluate the job stack based on this knowledge. Improperly punched cards are routed to an alternate stacker.

If the input hopper is empty, the stacker is full, or the chip box is full (some devices), or if the device is in the manual mode, the following message is issued:

!!CPndd MANUAL

The operator corrects the condition and presses the START button on the card punch.

If a power failure or a feed check error occurs, the system outputs one of the following messages (depending on where in the cycle the error took place):

!!CPndd ERROR

!!CPndd TIMED OUT

If the card in the hopper is damaged, the operator removes it, presses the RESET button on the card punch, and responds with the key-in

CPndd R (NL)

In the event of a power failure, the operator presses the RESET button on the card punch and responds with the key-in

CPndd R (NL)

DISK PACK MESSAGES/KEY-INS

If the operator enters the key-in

DED DPndd,R (NL)

and there are no open files on the specified disk pack, CP-R outputs the message

!!IDLE

If there is at least one open file on the specified disk pack at the time the DED key-in is performed, CP-R will output the message

!!DPndd IDLE

when there is no longer an open file on the indicated disk pack. The operator may now remove the pack from the indicated unit, insert a different pack, and key in

UND DPndd,R (NL)

to allow use of the new pack.

DISK DATA PROTECTION

Software protection of the data on disk storage is provided on disk file and area accesses. Areas with write protection code 'S' (system) may not normally be written by any user program. Areas with protection 'F' (foreground) can not normally be written by background programs. Disk area protection is specified when the area is defined by SYSGEN. Write protection does not normally apply to device-access disk operations, but this type of access is normally permitted only to foreground programs. All software restrictions on disk access may be overridden for a specified job (including BKG, the background job) by use of the SY key-in. The message

!!yyndd WRT RESTRICTED

or

!!PAUSE KEY-IN SY

(if included in the background command stream) will be output on OC to inform the operator that access to a protected disk area is requested. The operator would not normally grant system privileges (key-in SY) unless he was assured that it was authorized for the requesting job.

LINE PRINTER MESSAGES/KEY-INS

When an irrecoverable print error is detected, the system outputs the following message:

!!LPndd ERROR

The I/O handler attempts to print a line x times (x = NRT, a DCB variable specified by the user) before outputting the above message. The operator's response after correcting the condition depends on the specific device and circumstances.

If the printer is out of paper, if the carriage is inoperative, or if the device is in the manual mode or off, the following message is issued:

!!LPndd MANUAL

The operator corrects the condition and presses the START button on the line printer.

If the line printer power is off, the system outputs the following message:

!!LPndd UNRECOG

If a printer went into test mode during an I/O operation, the following message is issued:

!!yyndd TEST MODE

The operator should correct the condition and respond with the key-in

LPndd R (NL)

If a printer became nonoperational during an I/O operation, the following message is issued:

!!yyndd NOT-OPERATIONAL

The operator should correct the condition and respond with the key-in:

!!yyndd R (NL)

MAGNETIC TAPE MESSAGES/KEY-INS

If an error occurs during the reading or writing of magnetic tape, the I/O handler attempts a recovery x times (x = NRT, a DCB variable). If the error is irrecoverable, the user is informed via an error return.

If a tape unit is addressed and there is no tape mounted or power is off, the following message is issued:

!!9Tndd UNRECOGNIZED

If an attempt is made to write on a tape unit without a write-permit ring, the following message is issued:

!!9Tndd WRITE PROTECTED

The operator's key-in response depends on the circumstances.

INPUT/OUTPUT SPECIFICATIONS

Throughout the CP-R system, specifications of input data sources and output data destinations are made by giving a physical device name (see Tables 6-8), a disk file identifier (see Table 9), or an operational label (see Table 10).

Table 6. I/O Device Type Codes

yy	Device Type
TY	Line printer.
CR	Card reader.
CP	Card punch.
9T	9-track magnetic tape.

Table 6. I/O Device Type Codes (cont.)

yy	Device Type
7T	7-track magnetic tape.
DC	Fixed-head disk.
DP	Disk pack unit.
PL	Plotter.
NO	Not a standard device. Used as a special purpose device for IOEX.
LD	Logical device.
LN[†]	Remote Terminal.

[†] In this case the device name will be LNxxx where xxx represents the decimal line number.

Table 7. Channel or Cluster/Unit Designation Codes

Specified Character (n)	Significance		
	Sigma 9 Unit Address	Xerox 550	
		Cluster Address	Unit Address
A	0	0	0
B	1	1	0
C	2	1	1
D	3	1	2
E	4	1	3
F	5	1	4
G	6	1	5
H	7	2	0
I	-	2	1
J	-	2	2
K	-	2	3
L	-	2	4
M	-	2	5
N	-	3	0
O	-	3	1
P	-	3	2
Q	-	3	3
R	-	3	4
S	-	3	5
T	-	4	0
U	-	4	1
V	-	4	2
W	-	4	3
X	-	4	4
Y	-	4	5
Z	-	5	0
0	-	5	1
1	-	5	2
2	-	5	3
3	-	5	4
4	-	5	5
5	-	6	0
6	-	6	1
7	-	6	2
8	-	6	3
9	-	6	4
blank	-	6	5
$	-	0	1
#	-	0	2
@	-	0	3
:	-	0	4

Table 8. Device Designation Codes

Hexadecimal Code (dd)	Device Designation
00≤dd≤7F (single devices)	Refers to a device number (00 through 7F).
80≤dd≤FF (multiple devices)	Refers to a device controller number (8 through F) followed by a device number (0 through F).

Table 9. Disk Area Codes

Code	Area
SP	System Program area.
FP	Foreground Program area.

Table 9. Disk Area Codes (cont.)

Code	Area
BP	Background Program area.
BT	Background Temp area.
XA	IOEX Access area.
CK	Checkpoint area.
D1 ⋮ DA ⋮ DF	Data area (number of data areas is defined at SYSGEN).

Table 10. System Operational Labels

Label	Standard or Optional	Reference	Comments
BI	Standard	Binary input	Used to input binary information.
BO	Standard	Binary output	Used by processors to output binary information.
C	Standard	Control command input	Used by the system and processors to read control commands.
CI	Standard	Compressed input	Used by AP.
CO	Standard	Compressed output	Used by AP.
DI	Optional	Debug input	Used by the Debug facility.
DL	Optional	Debug listing	Used by the Debug facility.
DO	Standard	Diagnostic output	Used by the system for postmortem dump and diagnostic messages and by AP for diagnostic messages.
DP	Optional	Debug patch	Used by the Debug facility.
ER	Optional	Error logging	Used by Error Logger.
LL	Standard	Listing log	Used by the system and AP to log control commands and other system messages.
LO	Standard	Listing output	Used for object listings from assemblies and compilations.
MO	Optional	Media output	Used by Media facility.
OC	Standard	Operator's console	Used by the system for key-ins and operator control. (Always assigned to a keyboard/printer.)
P1	Optional	Debug utility 1	Used by the Debug facility.
P2	Optional	Debug utility 2	Used by the Debug facility.

Table 10. System Operational Labels (cont.)

Label	Standard or Optional	Reference	Comments
SE	Optional	Save environment	Used by Alarm Saver.
SI	Standard	Symbolic input	Used by processors to input source (symbolic information).
SO	Standard	Symbolic output	Used by SL-1 and AP.

PHYSICAL DEVICE NAMES

Physical device names are of the form

$$\begin{bmatrix} yy\begin{bmatrix} ndd \\ xxx \end{bmatrix} \\ 0 \end{bmatrix}^{\dagger}$$

where

yy identifies the type of device (see Table 6).

n identifies the channel or cluster/unit number (see Table 7).

dd identifies the device number (see Table 8).

xxx identifies the line number of a communication line.

and

0 identifies the null device.

The null device does not exist as such, but may be used whenever a device of the form yyndd is valid. Input and output requests to the null device cause a normal type completion and zero actual record size to be posted. No data is moved to or from the I/O buffer.

Note that a physical device name may be the name of a logical device (LDndd).

DISK FILE IDENTIFIERS

Disk file identifiers have two basic forms:

1. area, name

where

area is the name of any disk area defined on the system.

name is the name of the file.

This form of file identifier may occur only in a parenthesized group, for example, as in the background command:

```
LOAD (OUT,SP,RADEDIT)
```

or the keyin:

```
BATCH (D1, JOBFILE)
```

Files in accounts other than the system account may not be identified in this form.

2. [name][.[area]][.[account]]

where

name is the name of the file. If omitted, the area must be specified, and the whole area is the identified medium.

area is the name of the disk area in which the file resides, and may be any area defined on the system. If omitted, the file is assumed to reside in a public area and the name will be determined by the system.

account is the name of the account in which the file is defined. Account names must have at least one character, and at most, eight. The account name may be omitted, in which case, a default is provided. If the area name is also omitted, the account name defaults to that in which the task which opens the file is running. If the area name is specified, the account name defaults to the system account.

Files of the same name may be defined as long as they differ in either account name or area name; however, if files of the same name and account are defined in different public areas, one of them will be inaccessible except by explicit specification of its area name, since by default, the one found first is accessed.

The Background Temp area (BT), is a special area in that its file names are restricted to the set, GO, OV, and X1 through X9 and account name is not required. Its primary

[†] See Command Syntax Notation for significance of brackets.

purpose is for work files for the background processors. Its files are temporary and will be discarded at the beginning of the next job.

Files in other areas are permanent files and will be kept by the system until they are explicitly deleted.

OPERATIONAL LABELS

Operational labels (oplabels) are names used to symbolically reference physical devices and disk files. All oplabels, are defined at SYSGEN and are given their permanent default, or system initialization assignments. System oplabels as listed in Table 10, are treated specially:

First, they are created automatically during SYSGEN. The Standard System Operational Labels are always created in all SYSGENS. The optional system operational labels are used by optional features in the CP-R operating system and are created only if their associated feature is included in the system. Any user desired oplabel must be defined in SYSGEN. Default oplabel assignments are also made in SYSGEN.

Second, they are used by the CP-R system and its processors as the standard assignments for standard DCBs. For example, processors producing binary output do so to the M:BO DCB, and unless explicitly changed, the DCB is assigned to the BO oplabel. Where the output actually goes is a function of the assignment of the BO oplabel.

Oplabels are considered job level resources. That is, each job within the system has its own separate set of oplabel assignments. The oplabel assignments for the CP-R job are used as the default assignments for other jobs. When a new job is activated, this set of default assignments becomes the initial assignments in the new job. Any reassignments in that new job affect only that job. The assignments are retained until changed or the job terminates.

The default assignments in the CP-R job may be changed (see STDLB key-in). The resultant set of assignments becomes the default set for any subsequently activated jobs.

RESOLVING I/O MEDIUM NAME AMBIGUITIES

When a command permits more than one I/O medium type to be named in the same field, certain ambiguities may arise as to which type of name is intended; for instance, '9TA80' could be either a file name or a device name. The following rules resolve this type of ambiguity:

1. Names preceded or followed by periods are parts of file identifiers. This means '.SI' is not an operational label, and '9TA80.' is not a device.
2. Names followed but not preceded by periods are file names, for example, '9TA80.' is a file name, not a device. 'SI.' is also a file name, not a disk area name or an operational label.
3. A name is a disk area name if it follows a period, it does not follow another area name in the same file identifier, it has two characters, and it is defined as an area name in the system.
4. A name is a disk file account name if it follows a period and is not selected by rule 3.
5. A name is an operational label if and only if it has two characters, it is defined as an operational label in the system, and it is not excluded by rule 1.
6. A name is a device name if and only if it has five characters, it is defined as a device in the system, and it is not excluded by rule 1.
7. Names which are not typed by rules 1-6 are tested first as area names in file ID format 1, then as file names (with unspecified area and account) in file ID format 2.

3. RUNNING BACKGROUND JOBS

BACKGROUND PROCESSING

CP-R provides a single stream of background batch processing, directed by background control commands, that are read through the C operational label. The control commands are described in the CP-R RT/BP Reference Manual. The general nature of background processing is described in this section.

JOB DEFINITION

The fundamental unit in background processing is the job. A typical job might consist of the following elements:

1. A JOB command card to signal the beginning of a new job. The monitor resets all operational labels to their permanent assignments.

2. Optional ASSIGN or STDLB commands that define to the monitor the peripheral devices and disk files that are to be used for I/O. These commands are only needed to change the inherent I/O assignments in a program.

3. A processor command that calls in the correct system processor (e.g., AP assembler), user program (or combination), or a RUN command to execute a program.

4. Any source data the program is designed to process. The data may be contained in a card deck, magnetic or paper tape, or disk data file.

A number of optional control commands can be included in a job, such as a LIMIT command to set the allowable execution time for a given program. Figure 4 illustrates the simplest case of a deck setup for a job.

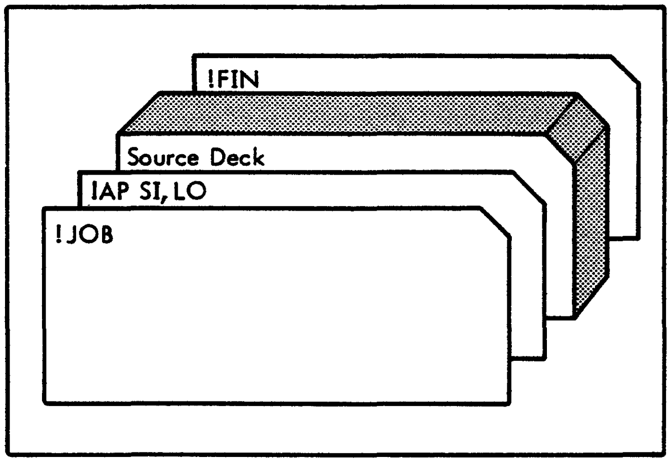


Figure 4. Source Program Assembly Example

In Figure 4, the symbolic input is received from the SI device and the listing output is produced on the LO device.

LOAD AND GO JOBS

"Load and Go" jobs are programs that immediately go into execution mode when the source program is successfully assembled or compiled. That is, the object program is loaded into core from a temporary file on the disk when the assembly is completed instead of being manually loaded from a card or tape device by the operator. Figure 5 illustrates a typical "Load and Go" job.

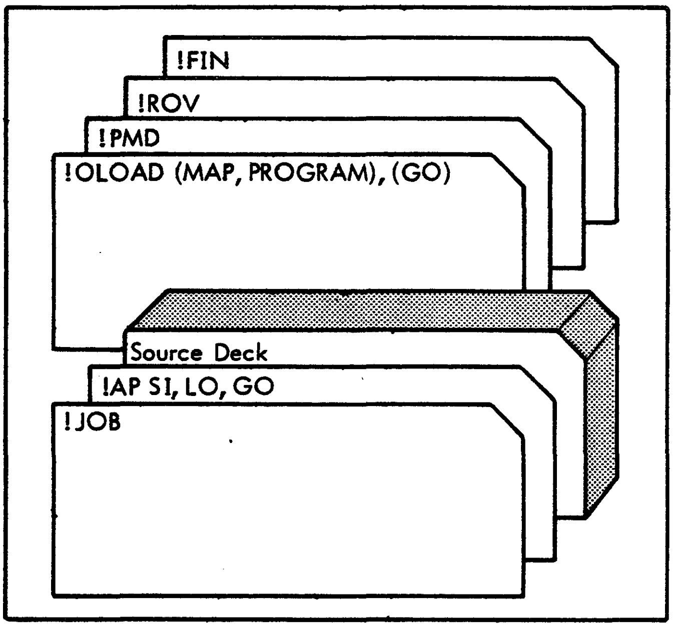


Figure 5. Load and Go Deck Example

In Figure 5, the binary object program produced from the assembly is placed in a temporary (GO) file from which it is later loaded into core for execution. The resultant file is always temporary and can not be retained from one job to another. The Overlay Loader will load the program root into the OV file for execution. A postmortem dump is specified.

JOB STEPS

A job step is the execution of a program (system processor or user program) in the background. The processors that operate under CP-R are service processors (RADEDIT and Overlay Loader) language processors, and user processors. Service and language processors are supplied with the CP-R system. User processors are created at the local installation.

GENERAL OPERATING CONSIDERATIONS

The basic types of background jobs the operator may encounter fall in the following general categories:

1. Assemblies or compilations of original source programs (written in some symbolic language such as FORTRAN IV or AP) into object modules. The output from such a job might include a source listing, an object listing, and an object program deck.

2. Trial executions of partly debugged programs called "test cases". Sample data is loaded and the program processes the data under various conditions. Additional control command options may be used to vary the processing of data or format of the final output.

3. Processing data by an operational program. Such programs generally reside in the Background Programs area of the disk if they are frequently or regularly used. Less frequently used programs may be loaded from a card reader or tape device. Data to be processed can be located in the foreground or background data areas of the disk or input from some other external source.

BACKGROUND JOB RESTRICTIONS

The monitor imposes two fundamental restrictions in processing background programs:

1. Background programs are given CPU time only after real-time hardware interrupts are satisfied. That is, nothing can take place in the background that inhibits interrupts or in any way interferes with real-time responsiveness. Thus, background programs will not be guaranteed any processing time if the foreground is very active.

2. The monitor prevents any attempt by a background program to write into, read from, or execute instructions in foreground core storage, to write into foreground disk file areas, or to utilize devices or services dedicated to real-time tasks. Any attempt by the background program to violate this protection or to execute priviledged instructions, either intentionally or through program error, may result in the monitor aborting the background program.

There is no read protection for foreground disk areas, and background programs can read from secondary file storage without restriction. A typical example of using the read feature would be a real-time data acquisition program that accumulates real-time data and writes it out on the foreground data file area of secondary storage. A background program could then be loaded at a later, less critical period to read and further process the data without disturbing the raw data that might be needed for some further real-time task.

Frequently-used user programs reside on the disk in load module form in the SP area, and are called via a !name command, where name is the name of the file.

OPERATOR INTERRUPT DURING CONCURRENT FOREGROUND/BACKGROUND OPERATIONS

If the operator depresses the INTERRUPT switch during concurrent foreground/background operations (perhaps to change the status of a background job), the monitor will not acknowledge the interrupt with a "–" prompt character until all current and waiting foreground processes are completed. This is because the Control Task that processes the key-in is connected to the lowest priority interrupt in the system. For the same reason, the operator cannot alter the current foreground task.

ATTENDED RUNS

Attended runs are frequently specified when new programs are submitted for processing. An !ATTEND control command in the user's job will inhibit the monitor ABORT routine, and the monitor will go into a WAIT state if it encounters an error it cannot correct. This gives the operator time to take memory dumps or initiate recovery action, if possible. If recovery procedures cannot be successfully completed, the operator's only recourse is to abort the job through an "X" key-in after taking any specified dumps.

SYMBIONT ATTENDED RUNS

Attended runs are of doubtful value in a symbiont system because the operator cannot depend upon the current background input or output to indicate the cause for a background wait. Therefore, the ATT OFF key-in is provided to disallow symbiont background jobs that require operator interaction.

4. RUNNING FOREGROUND

OVERVIEW

Operator control and manipulation of foreground programs requires a knowledge of some of the characteristics of such programs and of how core memory is partitioned. The positioning of memory into various areas such as the systems area, primary area, primary blocking buffering area, secondary area, etc., is installation-determined during SYSGEN and becomes the standard default allocation. In a typical installation, core memory will be allocated as is illustrated in Figure 6.

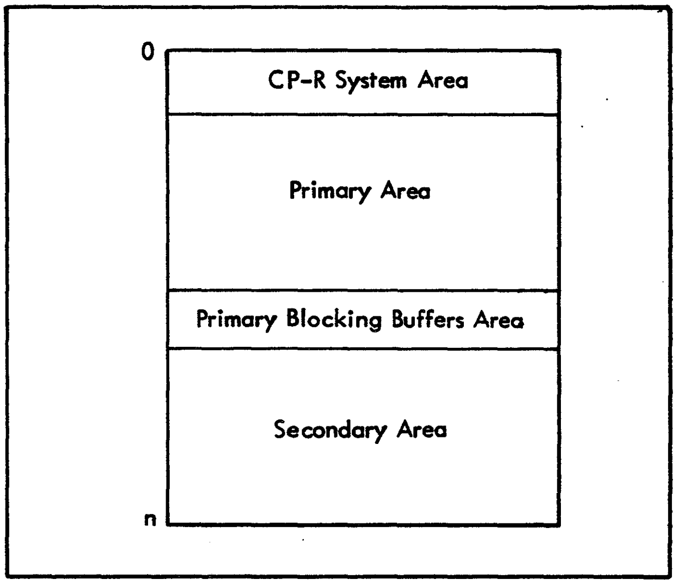


Figure 6. Typical Core Memory Partitioning

Several primary tasks can be resident simultaneously in protected core; the number that can be resident is determined by the number of entries in the Load Module Inventory established during SYSGEN.

REAL-TIME INTERRUPTS

Primary and secondary load modules are frequently loaded and initialized at system boot-time. Program execution is initiated when a signal is received from some external source, such as telemetry equipment, factory equipment, or a medical device in a hospital. The signal causes triggering of an interrupt that initiates execution of its connected task (a body of code within the program).

Each primary load module in core will be connected to one or more interrupts. While these interrupts are generally triggered from outside the system, the signal can be internal; that is, one foreground program can trigger execution of another program or the signal can be received from an internal interval timer (real-time clock) Each interrupt has a different priority level, and when more than one interrupt is in a waiting state, this priority level determines which program will next become active.

CONSOLE INTERRUPT PRIORITY LEVEL

To prevent accidental interference with critical real-time processes, the Console Interrupt Task triggers the CP-R Control Task which is normally the lowest priority interrupt in the system. This prevents the Monitor from processing a key-in until the active interrupt and any pending interrupts have been processed. The system will then acknowledge an operator initiated interrupt with a "-" prompt character on the OC device.

RUNNING FOREGROUND PROGRAMS

Before any foreground program can be loaded into main memory for execution, it must have previously been loaded onto the disk in absolute memory image format.

The SJOB system call or SJOB key-in creates a foreground job by setting up the job controls and table entries but does not initiate any task within the job, which is the function of the CALs, key-ins, and control commands discussed below. To create a foreground job through the operator's console, key-in

SJOB jobname (NL)

where jobname is the name of the foreground job.

The request to run a program is initiated in one of four ways:

RUN taskname[,priority] key-in where taskname is the name of a program in the FP area. The absolute load module will always be run under the CP-R job.

!RUN fid [,priority] control command read in from the background job stack on the C device for the load module specified by fid. The command must be preceded by an FG key-in. The absolute load module will always be run under the CP-R job.

!RUN is also available as a system call within the user's load module.

!ROV control command read in from the background job stack for a job located in the OV file. (Only one job at a time is permitted on the OV file and is always called OV.) The command must be preceded by an FG key-in.

INIT taskname[, (JOB, jobname)], $\left\{\begin{matrix}\text{PRI}\\\text{SEC}\end{matrix}\right\}$

taskname is the file identifier for the load module, with the exception that the specified file name may be converted to an actual file name by lookup in the job program table. If no match is found in the JPT, the specified file name is used.

When the CP-R system encounters the request to run a primary load module, it performs the following steps:

1. LOAD: The CP-R system reads the program header, which specifies the program's required core location and size, and tests if the required core space is available. As an assembly option, the message

   ```
   !!LOADED PROG xxxxxxxx
   ```

   will be output on OC.

2. INITIALIZATION: Control is transferred to the program entry at the CP-R Control Task priority level. The steps the program then follows depend on its structure. Typical steps for execution could be the following:

   a. Arm and enable interrupts.

   b. Set up interval times (the timers define the time delay between the triggering of interrupts).

   c. Connect the tasks to their interrupts.

   d. Exit foreground initialization.

RELEASING FOREGROUND TASKS AND JOBS

Release is accomplished through either operator key-in if he wants to bump the program for a different program that requires some of the same core space, or through a system call from the active foreground program. In any event, the following actions will take place:

1. All interrupts used in the program are disarmed, disabled, and disconnected.
2. The memory space used by the program (both foreground and any background, primary and secondary) is marked as unused.

To release an active foreground program and load another foreground program through the RLS key-in, proceed as follows:

Depress console INTERRUPT key. This causes a flag to be set and the CP-R Control Task to be triggered. Since the Control Task Interrupt is the lowest priority in the system, it will not be recognized until the active foreground interrupt and any waiting interrupts have completed their execution cycle. The system will then output the "-" prompt character on the OC device.

Then, key in RLS or EXTM as follows:

RLS $\left\{\begin{matrix}\text{OV}\\\text{taskname}\end{matrix}\right\}$ (NL)

where

OV is the foreground program currently in the OV file.

taskname is the name of a currently active program that is located in the FP area of the disk.

As an assembly option when the active program is successfully released, the system will output the message

```
!!RELEASED PROG xxxxxxxx
```

on the OC device.

EXTM taskname[, (JOB, jobname)] (NL)

where

taskname is the name of the task to be terminated.

JOB is the keyword and indicates that the job under which the task is running will be specified. If the JOB option is not present, jobname defaults to the CP-R job.

jobname is the name of the job under which the task to be terminated is running.

The operator can terminate an entire job, if desired, by keying in

KJOB jobname (NL)

where jobname is the name of the job to be terminated.

TASK STATUS DETERMINATION

It is frequently useful for the operator to confirm the results of an initialization or releasing key-in as described above. Use of the STAT key-in will cause the current status of a specified task to be output on OC. Key in

STAT taskname[, (JOB, jobname)] (NL)

where

taskname is the name of the task for which status data is desired.

JOB is a keyword and indicates that the job under which the task is running will be specified. If the JOB option is not present, jobname defaults to the CP-R job.

jobname is the name of the job under which the task is running.

The status of the specified task that appears on the operators console will have the format and meaning shown in Figure 3.

OPERATOR INTERVENTION

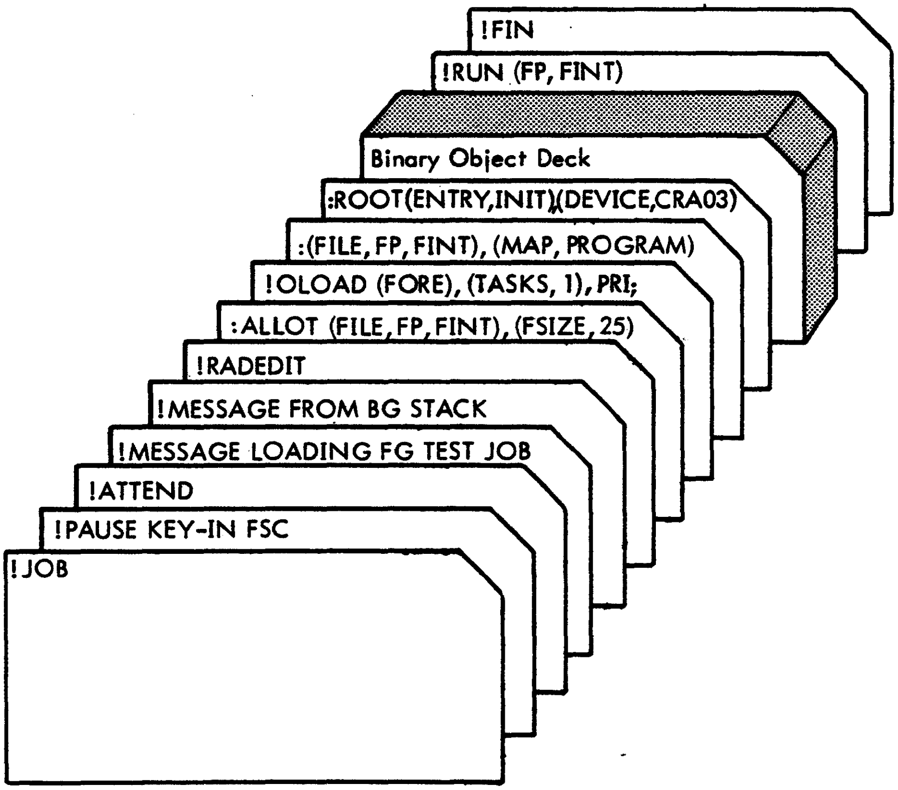
Foreground programs already operational in the system almost never require operator manipulation. Occasionally, the monitor may output a status message to request the operator to ready some special purpose device.

ERROR RECOVERY

Error recovery for operational foreground programs is impossible in most cases, and the operator's only recourse is to call the designated person or Customer Engineer (for a clearly evident machine malfunction).

LOADING NEW FOREGROUND

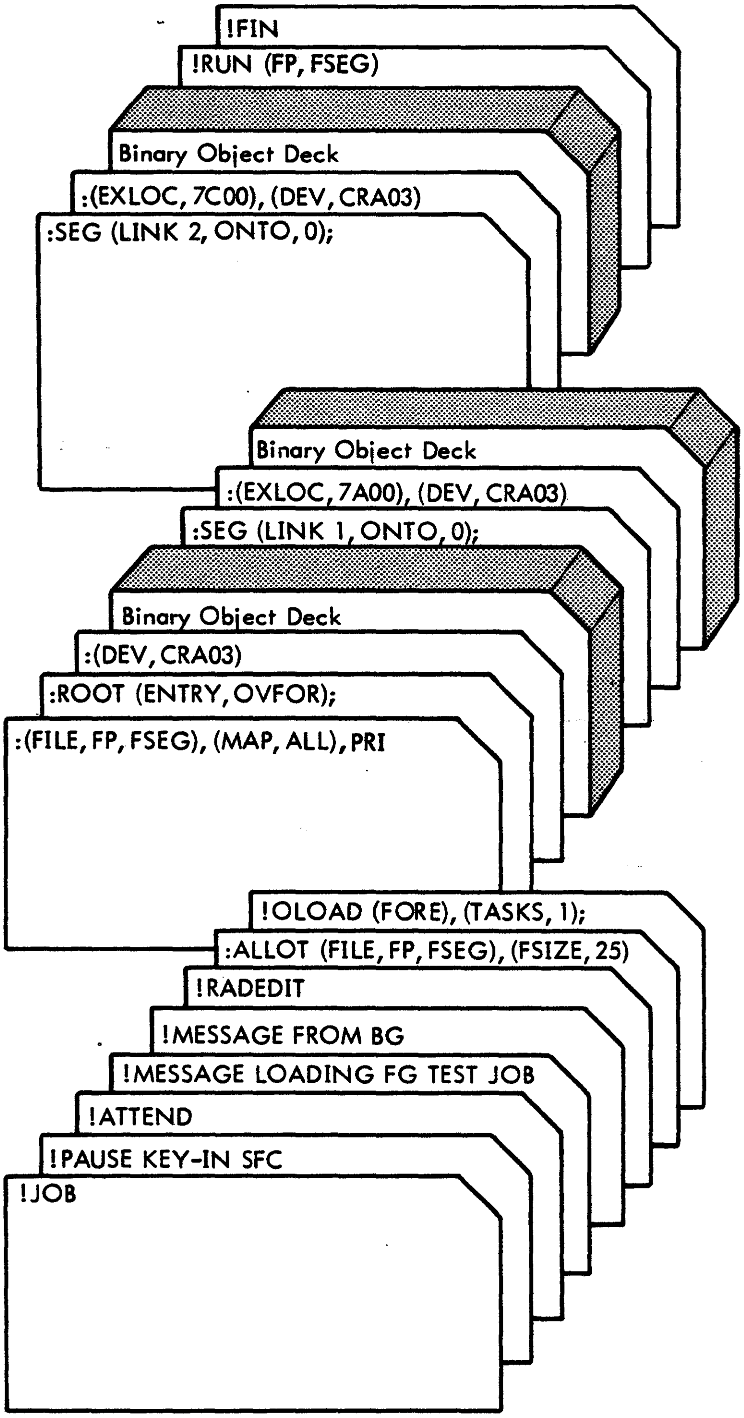
New foreground programs can be loaded into the Foreground Programs area of the disk or disk pack from the background job stack without a new SYSGEN. Operator handling of such programs is identical to background jobs except that loading must be preceded by an SY key-in to access protected disk areas. Figures 7 and 8 illustrate typical examples of loading foreground programs from the background job stream.



```
!FIN
!RUN (FP, FINT)
Binary Object Deck
:ROOT(ENTRY,INIT),(DEVICE,CRA03)
:(FILE, FP, FINT), (MAP, PROGRAM)
!OLOAD (FORE), (TASKS, 1), PRI;
:ALLOT (FILE, FP, FINT), (FSIZE, 25)
!RADEDIT
!MESSAGE FROM BG STACK
!MESSAGE LOADING FG TEST JOB
!ATTEND
!PAUSE KEY-IN FSC
!JOB
```

In this example, the RADEDIT allots a file (FINT) in the Foreground Programs (FP) area. The Overlay Loader loads the binary object deck in the file FINT in core image format. The !RUN control command causes execution of the foreground primary load module. A PROGRAM map is specified.

Figure 7. Load and Execute Foreground Program



```
!FIN
!RUN (FP, FSEG)
Binary Object Deck
:(EXLOC, 7C00), (DEV, CRA03)
:SEG (LINK 2, ONTO, 0);
Binary Object Deck
:(EXLOC, 7A00), (DEV, CRA03)
:SEG (LINK 1, ONTO, 0);
Binary Object Deck
:(DEV, CRA03)
:ROOT (ENTRY, OVFOR);
:(FILE, FP, FSEG), (MAP, ALL), PRI
!OLOAD (FORE), (TASKS, 1);
:ALLOT (FILE, FP, FSEG), (FSIZE, 25)
!RADEDIT
!MESSAGE FROM BG
!MESSAGE LOADING FG TEST JOB
!ATTEND
!PAUSE KEY-IN SFC
!JOB
```

In this example, the RADEDIT allots space for a file called FSEG in the Foreground Programs (FP) area. The Overlay Loader loads a root and two segments into FSEG in core image format. Following an operator FGC key-in, the foreground primary load module is executed via the !RUN control command. An ALL map is requested.

Figure 8. Load and Execute Segmented Foreground Program

5. AVAILABILITY OPERATING PROCEDURES

INTRODUCTION

This chapter gives information and procedures concerning abnormal operations ("abnormal" in the sense that the information and procedures are all related in some degree to the occurrence of abnormal conditions in the system). The operational areas covered are

- System Alarm Conditions:
  - Possible or probable causes/source of error.
  - Possible recovery procedures.
- On-Line Availability Procedures:
  - Error Log management and analysis.
  - Device isolation and substitution.
  - Operation of device exercisers.
  - Dump Analysis.

SYSTEM ALARM CONDITIONS

In the event of a serious system alarm condition such as a critical machine fault, CP-R will attempt an orderly shutdown of the operating system. The shutdown can take either of two forms depending upon whether #CRASH option was included in the system at assembly time.

SYSTEM ALARM WITH #CRASH OPTION

CP-R has a number of software checks that will indicate when an irrecoverable software or hardware fault has occurred. When such a fault is detected, an immediate and controlled system shutdown will be forced and the sequence of events listed below will follow:

1. All inhibits will be set and all registers saved.
2. A one-second wait will occur, during which a descending tone will be output on the console speaker to permit current I/O to run down.
3. All I/O devices will be HIOed through their primary device address.
4. The 1-Khz alarm will be turned on.
5. An alarm message in the form of "!!ALARM text" will be output on OC.
6. If there is a CK area present in the system, memory will be saved on the CK area until either the end of memory or end of area is reached. 1024-byte blocks will be used and a "CORE SAVED" message will be output on OC when end of memory or CK area is reached.
7. The alarm will be turned OFF.
8. The system will then take one of three actions.
   a. Default action is to hang in a one-instruction LOOP.
   b. If REBOOT option was specified at SYSGEN on the :MONITOR card, or if the REBOOT key-in was the cause of the alarm, or if a negative value was set by a task using the RECALARM CAL, the system will automatically reboot itself.
   c. If a valid real address was set by a task using the RECALARM CAL, the system will branch to that address in master-real mode with all interrupts inhibited.

The core saved on the CK area may be interrogated by using the CKD, CRS, or CRD key-ins when the system is brought back up (see Table 2).

Note: The first ROLL-OUT operation will overwrite the CK area and the memory image will be destroyed.

SYSTEM ALARM WITHOUT #CRASH OPTION

The procedures established for the operator's installation might also involve an attempt at a stand-alone memory dump prior to rebooting CP-R from disk:

1. All inhibits will be set and all registers saved.
2. A one-second wait will occur, during which a descending tone will be output on the console speaker to permit current I/O to run down.
3. All I/O devices will be HIOed through their primary device address.
4. The 1-Khz alarm will be turned on.
5. An alarm message in the form of "!!ALARM" will be output on OC.
6. The alarm will be turned OFF.

7. The system will then take one of three actions:

   a. Default action is to hang in a one-instruction LOOP.

   b. If REBOOT option was specified at SYSGEN on the :MONITOR card, or if the REBOOT key-in was the cause of the alarm, or if a negative value were set by a task using the RECALARM CAL, the system will automatically reboot itself.

   c. If a valid real address was set by a task using the RECALARM CAL, the system will branch to that address in master-real mode with all interrupt inhibited.

OPERATOR INITIATED SYSTEM ALARM

To force a shutdown of the operating system, perform the following steps:

1. Place computer in the IDLE state.
2. Clear the data switches.
3. Store INSTR ADDR (this clears next instruction).
4. Set data switches 5, 6, and 7.
5. Insert PSW2 (this sets all inhibit bits).
6. Step and place computer in RUN state (this executes an illegal instruction, zero, while inhibits are set, causing an alarm).

SYSTEM RECOVERY

In all cases, the only recovery action the operator can take is to reboot the system from disk. The CK area may be saved by a CRS key-in (see Table 2) and the ANALYZE processor run to process the memory dump and to provide a formatted dump analysis (see "Dump Analysis" later in this chapter).

If the alarm condition persists through several recovery attempts, a Xerox Field Engineering service representative should be notified.

ON-LINE AVAILABILITY PROCEDURES

ERROR LOG MAINTENANCE AND ANALYSIS

CP-R maintains an error log in which abnormal events such as device errors or machine faults are automatically logged along with normal system information and events. The error log output is used to maintain a record of system operation to expedite maintenance. The error log (operational label ER) can be assigned to a file, magnetic tape, or card punch.

The operator is involved with the error log in five ways at installations where error log is performed:

1. The contents of the on-line error log will need to be purged from disk to an off-line storage medium such as magnetic tape or cards if it overflows during system operation. A CP-R message will be output to the operator to indicate such a condition.

2. The operator may be required to run the ELLA processor, either at predetermined times or as circumstances dictate, or both (see "Error Log Listing" later in this chapter).

3. The operator may display a summary of device errors through use of the ESUM key-in (see Table 2).

4. At run-time the operator can dynamically include or exclude error logging with the ELOG key-in.

5. At run-time the operator may enter annotations into the error log by using the ERRSEND key-in.

ERROR LOG PURGING

The need for purging the on-line error log from disk will be indicated by the following CP-R message on OC:

```
!!ERROR LOG IS FULL
```

The message indicates an actual overflow condition; that is, one or more entries are already lost.

To purge the error log file, the operator performs the following procedure:

Copy the error log file (ERRFILE) to another file or device and then key-in

```
ELOG   PURGE
```

ERROR LOG LISTING

The system error log file is formatted and listed by ELLA, the Error Log Listing and Analysis processor. ELLA may be used either to summarize, sort, and list the entire log file, or list selected portions of the file.

A comprehensive discussion of ELLA is given in the Availability Reference Manual, (CP-R) Publication Number 90 31 10. The step-by-step procedures most frequently useful are detailed below.

ERROR LOG ANALYSIS

Assuming ELLA is already in the system at run-time, enter the following job to obtain a summary analysis of a system/ device error recorded in the current (on-line) error log:

```
!JOB
!ATT
!RUN ELLA.BP

        or

!ELLA (if ELLA is a background processor)
!EOD
!FIN
```

which causes the ELLA processor to be initiated. This job assumes that the error log database is a file named 'ERRFILE'. Otherwise, the SET command must be used to reassign the error log data base[†], (see Availability Reference Manual Publication Number 90 31 10).

This procedure will produce a summary of all system errors, then device errors, if any. If a particular device shows a significant number of errors, normally the device should be isolated and submitted to verification by an on-line device exerciser as soon as possible. (See below for procedures.)

If ELLA is not already in the system at run time, load it using the following commands:

```
!JOB
!PAUSE STDLB BI TO ELLA TAPE, SYC
!RADEDIT
DELETE (FILE, ELLA.SP)
ALLOT (FILE, ELLA.SP), (FSIZE, 100), (RSIZE, 256)
!OLOAD (FILE, ELLA.SP), MAP, LIB, FORE
:ROOT (OPLB, BI, EOD)
:SEG (LINK, 1), (OPLB, BI, EOD)
:SEG (LINK, 2), (OPLB, BI, EOD)
:SEG (LINK, 3), (OPLB, BI, EOD)
:SEG (LINK, 4), (OPLB, BI, EOD)
:SEG (LINK, 5), NONE, (EXLOC, 10000)
:RES (CSTAB, 7680)
:ASSIGN (M:SI, C)
!RADEDIT
TRUNCATE (FILE, ELLA.SP)
!FIN
```

TOTAL ERROR FILE SUMMARY AND LISTING

Run (or key-in after using the TY key-in) the job stack given below to obtain both a summary of the error log file and a formatted dump of its entire contents on the line printer:

```
!JOB
!ELLA
SUM
CLIS
END
!FIN
```

CURRENT DAY ENTRIES SUMMARY AND LISTING

Run (or key-in) the job stack given below to obtain a summary and detailed dump restricted to current day error log entries on the line printer:

```
!JOB
!ELLA
TIME,00:00
SUM
CLIS
END
!FIN
```

DEVICE EXERCISER OPERATIONS

The on-line device exerciser EXER is an optional CP-R processor that verifies the operational status of the following types of devices:

- Card Reader (CR)
- Card Punch (CP)
- Line Printer (LP)
- Magnetic Tape Drive (MT)

The mnemonics in parentheses represent the names of the respective device tests. Device verification is indicated when either of two conditions exist:

1. Faulty device operation is suspected and the need for servicing must be determined.
2. Servicing has been completed and elimination of a device problem requires verification.

DEVICE ISOLATION

An I/O device that is exhibiting an excessive number of errors or is otherwise performing abnormally can be isolated from the system; i.e., be made unavailable, and except for disk units, another like device can be substituted in its

[†] Not initially available.

place, if necessary. Following isolation, the device can be exercised by an on-line diagnostic program or can be serviced without disrupting normal system operation.

A device is isolated by means of the DED (Device Dedicated) key-in:

DED yyndd, D $\left[,\left\{\begin{matrix}I\\D\end{matrix}\right\}\right]$

yyndd refers to a physical device name

where

yy specifies the type of device; CR, MT, etc.

n specifies the IOP number.

dd specifies the device number.

I dedicates all devices on IOP n.

D dedicates only device yyndd.

If neither I nor D is specified, all devices on the same multi-unit controller are dedicated.

When a device is made unavailable; i.e., declared to be "down", any nondiagnostic program that refers to it (via CP-R I/O services) will receive a device-unavailable status or the equivalent. Only diagnostic programs, such as an on-line exerciser, have effective access to a device in this state.

In a symbiont system, a symbiont will not attempt any accesses to a device which has been isolated by the DED key-in.

Diagnostic programs accessing an isolated device will bypass any symbiont and will access the device directly.

This allows normal symbiont operations to be used with EXER Jobs while still allowing direct diagnostic use of the device. For example, the line printer exerciser may be run in a symbiont system. While the line printer is isolated, normal Job output will be suspended by the symbiont and the diagnostic output will go directly to the line printer.

INITIATING A DEVICE EXERCISER

Perform the following procedures to initiate a device exerciser:

1. Isolate the device from system (mark it as "down"), if necessary, using the procedure described above.
2. Enter the following job:

```
!JOB
!STDLB (DO,yyndd)
!EXER
!EOD
!FIN
```

   where yyndd is the address of the device to be tested.

   The above job deck assumes EXER is already present in the system. If it is not present, load the binary object modules from cards into the OV file using the following job stream:

```
!JOB
!OLOAD (UDCB,2),(MAP),LIB,BACK,SMM
!ROOT (DEVICE,CRA03,1)
(ECP object module)
:SEG (LINK,1,ONTO,0),(DEVICE,CRA03,1)
(object module for Card Punch/Card Reader
    Exerciser)
:SEG (LINK2,ONTO,0),(DEVICE,CRA03,1)
(object module for Line Printer Exerciser)
:SEG (LINK3,ONTO,0),(DEVICE,CRA03,1)
(object module for Magnetic Tape Exerciser)
:ASSIGN (F:TD,DO)
!STDLB (DO,yyndd)
!ROV
!FIN
```

   Regardless of which deck is used, the following message will be output on OC:

```
!!EXERCISER LOADED
!!INPUT EXERCISER NAME
```

3. Enter the appropriate two-character test mnemonic on OC to commence device testing.

   Examples:

   CP(NL) (card punch)

   MT(NL) (magnetic tape)

CARD PUNCH VERIFICATION

1. If not already isolated, isolate the card punch from the system with the DED key-in as follows (typical example):

   DED CPA04, D, D

2. Load an adequate supply of blank cards in the input hopper and ready the card punch.

3. Load the job deck as per item 2 under "Initiating a Device Exerciser".

4. When the message

```
!!EXERCISER LOADED
!!INPUT EXERCISER LOADED
```

   appears on OC, type in CP to commence execution.

5. When testing is completed and it is desired to return the card punch to system use, input the UND key-in as follows (typical example):

   UND CPA04, D

MAGNETIC TAPE UNIT VERIFICATION

1. If not already isolated, isolate the unit from the system with the DED key-in as follows (typical example):

   ```
   DED 9TA80,D,D
   ```

2. Mount a scratch tape on the unit to be tested. Bring the tape to load point and ready the device.

3. Load the job deck as per item 2 under "Initiating a Device Exerciser".

4. When the message

   ```
   !!EXERCISER LOADED
   !!INPUT EXERCISER NAME
   ```

   appears on OC, type in MT to commence execution.

5. When testing is completed and it is desired to return the unit to system use, input the UND key-in as follows (typical example):

   ```
   UND 9TA80,D
   ```

CARD READER VERIFICATION

1. If the card reader test deck is not available, create one by running the Card Punch Exerciser using the procedure listed under "Initiating a Device Exerciser" and "Card Punch Verification". The resulting punched deck output will be used as the card reader test deck.

2. Load the test deck, followed by an !EOD card, into the input hopper.

3. If the device is not already isolated, interrupt and isolate the card reader with a DED key-in as follows (typical example):

   ```
   DED CRA03,D,DⓃ
   ```

4. Interrupt and key-in TYC Ⓝ to read control command input from OC.

5. Type in the following control commands from OC:

   ```
   !JOBⓃ
   !STDLB (DO,CRA03)
       where A03 is a typical ndd example
   !EXERⓃ
   ```

6. When the message

   ```
   !!EXERCISER LOADED
   !!INPUT EXERCISER NAME
   ```

   appears on OC, type in CR to commence the exercise.

7. When the exercise is complete, type in !FIN Ⓝ.

8. If the card reader is to remain isolated, interrupt and key-in C Ⓝ to continue normal processing. If the card reader is to be returned to normal system use, interrupt and undedicate the card reader with the UND key-in as follows (typical example):

   ```
   UND CRA03,D
   ```

   Interrupt and key-in CC Ⓝ to return control of the C device from OC to the card reader. Interrupt again and key C Ⓝ to continue normal background processing.

LINE PRINTER VERIFICATION

1. If the device is not already isolated, isolate the line printer with the DED key-in as follows (typical example):

   ```
   DED LPA02,D,D
   ```

2. Ensure the line printer has an adequate supply of paper and ready it.

3. Enter the following job stack:

   ```
   !JOB
   !STDLB (DO, LPA02)
       where A02 is a typical ndd example
   !STD (LL,0)
   !EXER
   !EOD
   !PAUSE KEY-IN UND LPA02,D
   !STD (LL,LPA02)
       where A02 is a typical example
   !FIN
   ```

   The above deck set up assumes EXER is already present in the system. If not, load EXER from cards onto the OV file as described under "Initiating A Device Exerciser" in this chapter, and start execution with an !ROV command.

4. When the message

   ```
   !!EXERCISE LOADED
   !!INPUT EXERCISER NAME
   ```

   appears on OC, type in LP to commence the exercise.

5. If it is desired to keep the line printer in isolation, ignore the message

```
!!PAUSE KEY-IN UND LPA02,D
```

when it appears on OC. Instead, remove the !STD (LL,yyndd) control command from the card reader hopper, interrupt, and key in C. If it isdesired to restore the line printer to system use, honor the !!PAUSE KEY-IN message and do not remove the !STD command.

ERROR REPORTING

Upon completion of any device test, a status report is printed on OC with the following format:

```
dd EXERCISER ERROR SUMMARY
NO. OF UNRECOVERABLE I/O ERRORS = xx
NO. OF DATA ERRORS = yy
NO. OF I/O = zz
```

where

dd is the two-character test mnemonic.

xx is the number of unrecoverable errors detected.

yy is the number of errors detected through data checking by EXER of records that had already passed the system's status checking.

zz is the number of I/O operations executed.

In addition to errors listed in the above error status display, other errors may have been reported by the system to the error log. To list these errors, run ELLA, the Error Log Listing and Analysis processor, using either the procedures given previously under "Error Log Listing" or run the following job (it may be convenient to enter the job from OC by using the TY key-in):

```
!JOB
!ELLA
TIME, hh:mm
CLIS
END
!FIN
```

If the above job is used, select a value for hh:mm that represents the time of day just prior to execution of the device test. For example, if you ran the EXER magnetic tape unit test at 2:30 PM and wished to see if any magnetic tape entries were produced in the error log, select the value 14:29 for the TIME command.

DIAGNOSTIC MESSAGES

Diagnostic messages may be output while running EXER. The occurrence of one of these messages usually indicates that an operational error has been made and testing must be restarted. The messages, their meaning and necessary response are given below.

```
TD IS NOT MARKED AS DOWN
```

The device to be tested has not yet been isolated from the system. Use the DED key-in to remove the device from system use so that EXER may test it.

```
OP LBL TD NOT ASSIGNED TO dd DEVICE
```

where dd is the two-character test mnemonic.

The device assigned for testing cannot be exercised by the request test. Either the !ASSIGN command or the two-character test mnemonic entered on OC is in error. Correct condition and reenter the job.

```
INVALID REQUEST
```

An invalid test parameter was entered on OC. Reenter with correct parameter.

```
WRONG TEST CARD DECK FOR TEST NO. n
```

where n is the test number.

An incomplete or out-of-sequence test deck is being used for card reader testing. Correct condition and re-input the deck.

```
INITIAL REWIND ERROR
```

EXER could not execute the initial tape positioning step of magnetic tape test (MT). Check the readiness of the tape unit being tested.

ANALYZE PROCESSOR

ANALYZE is an availability processor that formats and prints CPR's internal tables from a binary core dump for quick comprehension by system programmers. Three user – created work files on rotating memory are required by ANALYZE for its own use. These are the dump file, the map file and the CPR file. In analyzing a dump of the same system, the SYSGEN-created map file and the CPR file in the SP area are used by ANALYZE by default.

The binary core dumps are automatically placed in the CK area when the operating system experiences an ALARM condition with which it cannot cope. Optionally, an operator can use the SNAP key-in to produce a core dump.

RUNNING ANALYZE

DUMP ANALYSIS FROM LOCAL SYSTEM CK AREA

Enter the following job stream from an analysis of a dump residing in the CK area of the local system:

```
!JOB
!RADEDIT
:ALLOT (FILE,da,dname), (FORMAT,U),
    (FSIZE,dfsize), (GSIZE,256)
!PAUSE KEYIN STD SE,dname.da;CRS;C
!ANALYZE
:DUMP (da,dname)
:DISPLAY
!FIN
```

where the user creates the following values:

da is the area that is to contain the dump file.

dname is the filename given to the dump file.

dfsize is the core size (in 256 word blocks) plus 1.

If the file CRASH.SP is allotted and the SE oplabels assignment not altered since the last system reboot, the following procedure may also be used to produce the dump analysis.

Keyin CRS, then run the job

```
!JOB
!ANALYZE
!FIN
```

SAVING A DUMP ON MAGNETIC TAPE

Enter the following job stream to write a dump on a magnetic tape:

```
!JOB
!REWIND 9TA81
!PFIL 9TA81, m
!PAUSE KEYIN STD SE,9TA81;CRS;C
!FIN
```

where m = 3(n-1) and n is the ordinal number of this dump on the tape starting with 1.

DUMP ANALYSIS FROM TAPE BY ANOTHER SYSTEM

Input the following job stream for an analysis of a dump placed on tape by another system:

```
!JOB
!RADEDIT
:ALLOT (FILE,da,dname),(FORMAT,U),
    (FSIZE,dfsize),(GSIZE,256)
:ALLOT (FILE,ma,mname),(FORMAT,C),
    (FSIZE,3)
:ALLOT (FILE,ca,cname),(FORMAT,U),
    (FSIZE,cfsize),(GSIZE,cgsize)
!ANALYZE
:DUMP (FILE,da,dname)
:MAP (FILE,ma,mname)
:CPR (FILE,ca,cname)
:TAPE (IN,9TA80),(DUMP,n)
:DISPLAY
!FIN
```

where the user creates the following values:

da is the area which is to contain the dump file.

dname is the filename given to the dump file.

dfsize is the number of 256-word blocks in the original computer rounded up to the next larger integer.

ma is the area which is to contain the map file.

mname is the name given to the map file.

ca is the area which is to contain the CPR file.

cname is the name given to the CPR file.

cfsize is 32768 divided by cgsize.

cgsize is the sector size in words of the original system's SP area.

n is the ordinal number of the dump on the tape to be displayed, starting with 1.

6. SYSTEM INITIALIZATION AND PATCHING

Modification of the resident CP-R system (including all system tables), CP-R overlays, and Job Control Processor can be performed at system boot time through patches defined on !MODIFY control commands. Any number of !MODIFY commands can be input, and the stack is terminated by a single !END command. The system is notified that patching is to take place by sense switch settings that are set prior to CP-R initialization.

During CP-R initialization, the Control Panel interrupt is inhibited so that a premature key-in will not prevent the completion of initialization.

At the end of initialization, the message

```
PLEASE KEYIN DATE AND TIME
```

will be output on the OC device. The operator must respond with the DT key-in. Any other key-in is considered to be an error.

As each !MODIFY command is processed, the overlay is read into core, modified, and written out to the system disk. Note that unused disk space between the end of an overlay and the end of an overlay's file (that is, unused space on the last sector of an overlay) can be used for a patch area.

INPUT OPTIONS

Sense switch (SSW) settings are used to select initialization options following a system boot. Because SSW settings are used in SYSGEN also, care must be taken to allow for the system boot which follows the SYSGEN process.

Foreground programs which are designated resident are normally loaded during the initialization process. The periodic scheduler is also normally reactivated to resume these tasks. This may be prevented by setting SSW4. No other functions are affected by the setting of SSW4.

Modify control commands and quick patch commands are input depending on the settings of SSW1, SSW2, and SSW3.

Setting SSW1 selects the C device for MODIFY command input. Setting SSW2 selects the OC device for MODIFY command input. Setting SSW1 and SSW2 causes the initialization process to hang following a WAIT instruction before any processing has been done. This allows the sense switch settings to be changed or any critical hand patching to be done before MODIFY or quick patch commands are read.

SSW3 set causes quick patch commands to be read from the same device as the MODIFY commands. If SSW1 and SSW2 are reset, the setting of SSW3 is ignored.

Setting	Meaning
XXX0	Load Resident foreground programs.
XXX1	Do not load resident foreground programs.
11XX	Hang at the start of initialization.
00XX	Read no MODIFY or quick patch commands.
100X	Read MODIFY commands from C device.
010X	Read MODIFY commands from OC device.
101X	Read MODIFY and quick patch from C device.
011X	Read MODIFY and quick patch from OC device.

QUICK PATCHES

Quick patches may occur at the beginning of the initialization process and use none of the system services. Because of this, they may be used to patch portions of the system used during the reading and processing of MODIFY commands. Also modifications to the early portions of the initialization processor may be made.

SSW3 set selects the reading of quick patches from either the C or OC device. SSW1 and SSW2 are used to indicate which device. Unlike the MODIFY commands, quick patches assume that the C device is a card reader and that the OC is a TTY type device. The reading of quick patches is always followed by the reading of MODIFY commands.

The format for quick patches is:

```
loc[+]    value[+]
```

where

loc specifies the hex location of the location to be patched. A trailing + indicates that CP-R relocation bias is to be added.

value specifies the value to be stored in that location. A trailing + indicates that CP-R relocation bias is to be added.

Quick patching is terminated by a blank card when reading the C device, and a NL when reading from the OC device.

PATCH COMMAND FORMATS

In the !MODIFY control commands listed below, the brackets only indicate options available in the specification field and are not actually used in the MODIFY statements; the indicated parentheses and commas are required. The general format for !MODIFY commands is

```
         ({module,loc})
!MODIFY  ({symbol,loc}),[x]value[,[x]value...]
```

where

symbol specifies either a table name or a resident overlay name.

module specifies one of the modules described below to be patched.

loc specifies the initial hexadecimal location to be modified and is relative to the beginning of the named module (for ABS, loc is relative location 0).

value specifies the hexadecimal value to be inserted in the location. Successive values go in successive locations. The value field may be repeated and has an alternate form.

(value, symbol)

Where symbol is either a table name or an overlay name, whose value will be added to the value field. This alternate form for the value field is also valid for all the examples shown here and on the next page.

x is the relocation flag indicating the address field of the value is to be relocated by adding the beginning address of any of the following:

I	is the Decimal Simulation start address.
J	is the JCP start address.
L	is the Floating-Point Simulation start address.
M	is the Monitor start address.
O	is the CP-R Overlay start address.
P	is the Patch area start address.
R	is the named module start address.
V	is the Convert Simulation start address.
Y	is the Byte-String Simulation start address.
Z	is the module preceding the current module.

PATCH SYSTEM OVERLAY OR JCP

```
!MODIFY (OLAY,name,loc),[x]value[,[x]value,
   ...:[x]value]
```

where name is defined in the SYSGEN map under "CP-R Program Allocation" as the first entries. The size (nonresident) or base address (resident) of each overlay accompanies its name.

PATCH SIMULATION ROUTINE

```
!MODIFY (SROU,name,loc),[x]value[,[x]value,...,
   [x]value]
```

where name = FPSIM, DECSIM, BYTSIM, or CVSIM.

PATCH CP-R MONITOR

```
!MODIFY (CPR,loc),[x]value[,[x]value,...[x]value]
```

PATCH SYSTEM TABLES

```
!MODIFY (ABS,loc),value[,value,... value]
```

MODIFY PATCH AREA

```
!MODIFY (PATCH,loc),value
```

TRACE COMMAND FORMATS

In the !MODIFY control commands listed below, the brackets only indicate options available in the specification field and are not actually used in the modify statements; however, the indicated parenthesis and commas are required. The general format for the !MODIFY command is

```
        {(SETRACE,task[,buffer[,events]])
!MODIFY {
        {(TRACE, {ON })[,entry point,entry point,etc.]
                 {OFF}
```

where

SETRACE changes the three trace parameters that follow. It does not activate or deactivate CP-R tracing.

task may be used to selectively trace a single task. The value is the value in the first byte of word 7 of the user's TCB. The default is zero, which traces all tasks.

buffer is an area to be used by the trace. This area should be large enough for the trace routine, entry point[†] symbol table, and 24 words for each event traced; that is, $\cong (700_{10} + (24 * \text{events}))$. The default is the foreground private area.

events is the hexadecimal number of entries to trace before cycling over the oldest entry (this is a circular trace). The default is X'100' events.

TRACE activates (ON) or deactivates (OFF) the trace. If no entry point parameters exist, "ON" traces all entry points and "OFF" turns the trace completely off.[†]

ON or OFF are described in TRACE, above.

entry point activates (ON) or deactivates (OFF) each entry point listed.[†]

The user may obtain the circular trace stack by dumping memory beginning at "buffer". The EBCDIC option (if using the DM key-in) should be used to provide readable entry points and overlay names[†]. The format of memory is as follows:

ENTRY POINT	OVERLAY	PSD	4E & 4F
(2 words)	(2 words)	(2 words)	(2 words)
(16 registers)			

Four words of EBCDIC Xs will follow the last event traced. If the circular stack has been fully used, the Xs will also appear at the end of the stack.

CLEAR COMMAND FORMAT

The CLEAR command is provided to initialize file area directories to contain no files. It may be of particular use to clear the symbiont areas when an IS/OS inconsistancy is reported at boot time, or to initialize areas SYSGENed but not saved when booting a SAVE tape. It is important to be aware that this command only zeroes the first sector of the area, to give an empty file directory. It does not zero the whole area.

The form of the command is

```
!CLEAR AA[,AA,...]
```

where AA is the name of an area to be given an empty file directory.

!END COMMAND FORMAT

The !MODIFY control command(s) must be followed by a terminating !END command with the form

```
!END [CPR]
```

where CPR indicates that the copy of the CP-R resident on the system disk is to be replaced by the current version of CP-R in core.

SYSTEM PATCHING AND TRACING DIAGNOSTICS

The diagnostic messages in Table 11 will be output on the OC device for a corresponding error in sense switch settings or !MODIFY commands.

[†] See "SYSGEN" for a discussion of entry points and overlay names.

Table 11. System Patching Diagnostics

Message	Meaning	Action
!ERROR ITEM XX-CORRECT AND CLEAR B $	If this is OC, only the message portion preceding the hyphen is output.. If not OC, the entire message is output.	If this is OC, input next command. If not OC, idle machine, increment address, and RUN.
!!PATCH LOC ERR-CORRECT AND CLEAR B $	If this is OC, only the message portion preceding the hyphen is output. If not OC, the entire message is output.	If this is OC, input next command. If not OC, idle machine, increment address, and RUN.
!!NO PATCH AREA - CLEAR B $ IGNORES MODIFY COMMAND	If this is OC, only the message portion preceding the hyphen is output. If not OC, the entire message is output.	If this is OC, input next command. If not OC, idle machine, increment address, and RUN

APPENDIX A. XEROX 550 REMOTE ASSIST STATION

The Xerox 550 provides for remote diagnostic assistance over a dial-up connection to the System Control Processor. (This processor is documented in the Xerox 550 Computer Reference Manual, 90 30 77.) Remote assistance can take two major forms:

1. Off-line assistance in which the diagnostic programs are run in a stand-alone mode (i.e., without the CP-R operating system).

2. On-line assistance in which the diagnostic programs are run under the CP-R operating system as privileged diagnostic user programs.

In either case, the connection procedure is basically the same.

CONNECTION PROCEDURE FOR REMOTE ASSISTANCE

1. Verify these switch settings on the Configuration Control Panel located inside the endbell housing:

   ALTSEL switch – down

   FSELA switch – up

2. Turn the REMOTE CHANNEL rotary switch located on the System Control Panel to the appropriate position:

   SCC for off-line connection

   I/O for on-line connection

3. After establishing voice contact with the Remote Assist individual (by using the data set in the TALK mode), inform him that everything is ready, place the data set in the AUTO mode, and hang up the receiver.

For off-line connection, the Remote Assist Station will have immediate contact with the computer and will be able to perform the necessary control functions just as if it were the primary operator's console.

For on-line connection, the communication with the CP-R must be initiated at the operator's console after the dial-up connection is established. This is accomplished with the key-in

```
LOGON TYA0B (NL)
```

If the connection has been established, LOGON will give the normal CP-R salutation to the Remote Assist Station user and request a log-on identification. When the user is logged on the standard logon report will appear on the operator's console.

TERMINATION OF REMOTE ASSISTANCE

At the completion of a remote assist session, the following actions should be taken:

1. Return the data set to the TALK mode.

2. Return the REMOTE CHANNEL rotary switch to the OFF position.

APPENDIX B. XEROX 550 LOAD-AND-GO DIAGNOSTICS

The Xerox 550 computer system includes hardware diagnostic facilities as a combined hardware and software feature. The diagnostic programs described in the appendix are referred to as the Diagnostic Programming System (DPS). DPS runs off-line and is loaded into the machine in place of the CP-R operating system to verify the operational status of the hardware and to aid in the diagnosis of any problems found.

These diagnostics can be loaded by the CP-R system operator in "load-and-go" mode, an easily operated, essentially automatic mode of operation. The diagnostics may be used periodically to determine the condition of the machine, or may be used whenever hardware problems are suspected.

If no problems are found by the load-and-go diagnostic system, the operator can have a high degree of confidence in the operation of the hardware. When problems are found, however, the operator should contact the service representative and relay the status information reported by the diagnostics (as explained below).

The load-and-go diagnostics are normally loaded in a manner that will not destroy any of the disk files that CP-R was maintaining, though still providing some measure of secondary storage testing. If full testing of rotating memories is desired, the CP-R file structure should be saved with a RADEDIT SAVE ALL procedure, as described in the CP-R Reference Manual, and the diagnostics can then be directed to do full testing.

After using the load-and-go diagnostics, the CP-R operating system may be reloaded simply by booting from disk if normal diagnostic operation was followed. If full testing of secondary storage was performed, the operating system must be loaded by booting the SAVE ALL tape.

LOADING DPS

The procedure listed in Table B-1 is recommended for execution of the load-and-go diagnostics. The procedure will result in

1. Execution of the system microdiagnostics for each system element (i.e., Basic Processor, MIOP, and Processor Interface).

2. Execution of the Basic Processor self-test.

3. Testing of the basic instruction set.

4. Loading and execution of the Diagnostic Programming System.

The console printout created by steps 2 through 8 of Table B-1 is shown in Figure B-1.

Each of the step-by-step actions that are necessary to run these tests are listed in the "Action" column of the table. The results that one should expect from a correctly functioning system are explained next to each "Action" item in the "Expected Results" column. If these results are not obtained and, instead, the results listed under "Abnormal Results" are observed, the operator should note the data described under "Initial Data to Report" and relay that information to the service representative. It is important that the sequence listed in Table B-1 be carefully followed, since failure notification from an altered sequence may result in erroneous failure analysis.

Table B-1. Load-and-Go Diagnostic Initialization

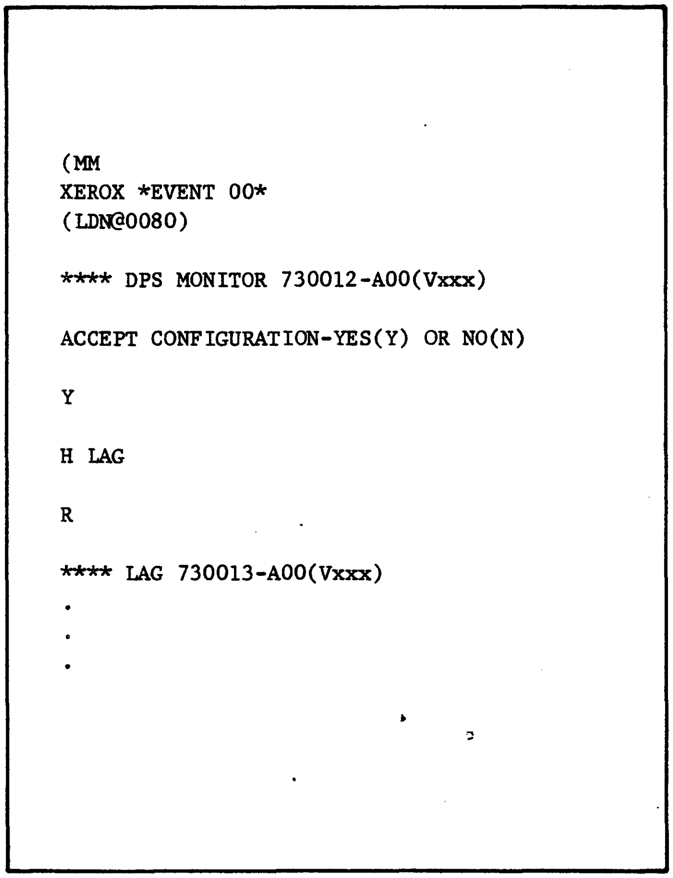
Action	Expected Result	Abnormal Result	Initial Data to Report if Result is Abnormal
1. If CP-R is operational, bring it to a quiescent condition.	DISPLAY keyin indicates no tasks other than CTRLTASK, TEX, MEDIA, SYMBIONT, and MMEXEC.	System cannot be brought to a quiescent condition. (It may eventually be necessary to reboot the system and the file structures. Some files, jobs, and the symbiont may be lost.)	If hardware problems are suspected, no data should be reported at this time. Continue with the procedure in order to load the Diagnostics. If software problems are suspected, this procedure should not be used since it verifies only the hardware.
2. Remove all tapes, other than scratch tapes, from all tape drives, and mount the Diagnostic Program Library tape on an available tape drive.	-	-	-

Table B-1. Load-and-Go Diagnostic Initialization (cont.)

Action	Expected Result	Abnormal Result	Initial Data to Report if Result is Abnormal
3. Bring all devices on the system to the ready state. (Ensure that line printers have an adequate supply of paper, that card punches have a full hopper of cards, that scratch tapes are mounted on tape drives, etc.).	-	-	-
4. On the System Control Panel (SCP), position the MAINT MODE rotary switch to the ON position.	MAINTENANCE MODE status indication illuminates.	No MAINTENANCE MODE indication.	Report that this event occurred.
5. Type: Z^{c}MM4 RET	Hardware responds: XEROX *EVENT 00*.	No XEROX *EVENT 00* message is output.	Report that "Super Reset" (MM4) has failed.
6. Type: Z^{c}LDNdddd RET, where dddd is the 4-digit hexadecimal address of the tape drive on which the Diagnostic Program Library tape was mounted in step 2. This will initiate a 4-step test sequence.	-	-	-
a. Microdiagnostics for all system elements are executed.	No output.	Hardware reports: EVENT F9 followed by the contents of the Single Clock Status Register for the failing system element.	Report the occurrence of EVENT F9 and the displayed contents of the Single Clock Status register.
b. The FROM test (BP self-test is executed.	No output.	The Diagnostic Tape has not moved off of Load Point within 60 seconds, and EVENT F9 has not occurred.	Type: P^{c} and observe the displayed contents of Program Status Word One (PSW1). Report the failure of the FROM test, and the contents of PSW1.
c. The Hardcore Instruction Set test is executed.	Diagnostic Tape has moved off Load Point.	The tape has moved off Load Point, but no messages are printed on the console within 2 minutes.	Type: P^{c} followed by I/O. Observe the displayed contents of Program Status Word One (PSW1) and the contents of General Register One (R1). Report the failure of the Hardcore Instruction Test and the contents of PSW1 and R1.
d. The Diagnostic Programming System Monitor is loaded.	The monitor title message is printed on the operator's console.	-	-

Table B-1. Load-and-Go Diagnostic Initialization (cont.)

Action	Expected Result	Abnormal Result	Initial Data to Report if Result is Abnormal
7. The Diagnostic Programming System (DPS) will request configuration confirmation. Type: Y RET (for "yes").	DPS proceeds to step 8.	Failure to proceed to step 8.	Report that this failure occurred.
8. DPS will type :H> (H indicates the halt state; > is the DPS prompt character). Type: LAG RET (for "load-and-go" operation). DPS will type :R> (indicating that it is now running the test sequence).	See "Running DPS-Normal Operation".	See "Running DPS-Error Detection".	See "Error Reporting".



```
(MM
XEROX *EVENT 00*
(LDN@0080)

**** DPS MONITOR 730012-A00(Vxxx)

ACCEPT CONFIGURATION-YES(Y) OR NO(N)

Y

H LAG

R

**** LAG 730013-A00(Vxxx)
.
.
.
```

Figure B-1. Console Display During Loading of DPS

LOADING DPS FOR MORE THOROUGH TESTING OF ROTATING MEMORY

If it is desired to write on the surfaces of rotating memory devices and thereby obtain a more thorough testing, the procedures outlined in Table B-1 should be modified as follows:

1. After Step 1 is completed and before mounting the Diagnostic Library Tape, the CP-R file structure should be saved through normal file saving methods.

2. Before typing "LAG" in Step 8, the following action should be taken:

   a. Type: REP,OP RET
   (The contents of the Operator Table will be printed.)

   b. Type: 08, RET
   (This sets a flag permitting writing on disk surfaces.)

RUNNING DPS-NORMAL OPERATION

Each test program is read from the Diagnostic Program Library tape and executed sequentially by the Diagnostic Programming System. The main frame is the first segment of the hardware to be tested. Several diagnostic programs will be called from the diagnostic tape for this purpose. After the main frame tests are complete, an applicable peripheral test is run on each peripheral device in turn. The "load device" (the unit on which the diagnostic tape was mounted) will not be tested, however.

When all peripheral units have been tested, a final test called the Systems Exerciser (SYSX) is loaded. Initially, SYSX gives the user the opportunity to change certain conditions of the test by printing status messages and then pausing for input. No action should be taken; the testing sequence will continue after a ten second delay.

When the testing is complete, DPS prints a completion message, halts, and types H> on the console. If no error messages (discussed below) have been produced, the system has passed all testing without error. A typical DPS console listing from an error-free run is shown in Figure B-2.

```
 .
 .
 .
* * * * LAG  730013-A00(Vxxx)

LOAD DEVICE NOT TESTED

SYSX 730010-A00(Vxxx)

IN OPERATOR TABLE, SET SOFTWARE 'SS' #5 ('08')
TO EXERCISE STORAGE DEVICES, AND TO ALLOW A BASE DEVICE

H>

R>

***02F0 READ ONLY
***01F0 READ ONLY
***0080 READ ONLY
***0081 READ ONLY
***0082 READ ONLY

H>

R>

LOAD-AND-GO TESTING COMPLETE

H>
```

Figure B-2. Typical DPS Listing From an Error-Free Run

Running DPS – Error Detection

If at any time during the testing cycle, the Diagnostic Programming System detects an error, it prints the message FDP ERROR or one of the SYSX error messages listed in Table C-2 below, preceded by a one-digit number (an index representing the amount of isolation data the system can produce on the error). Testing will stop at this point and H> will be typed on the operator's console.

Table B-2. SYSX Error Headings

```
DATA ERROR
INST ERROR
I/O TIME OUT
POSITION ERROR
UNEXPECTED I/O INTERRUPT
SIO FAILURE
I/O ERROR
DATA OVERRUN
I/O MEM FLT
IOP ERROR
```

The Diagnostic Programming System selects a line printer on which to list detailed status and error information. Tables are periodically dumped on this logging device as testing progresses. Additionally, the name of each test is printed when it is executed. If an error is detected, as described above, information about its nature is printed on this device prior to the system halt. The format of all error data logged on the printer is similar to the partial illustration given in Figure B-3.

```
.
.
.
  1-FDP ERROR

     SEQ# S  ID    TIME
     0001 4 3000 00:00:00

     PROG  TM  MODE  ERRORS   MOD# UNAD
     AUTO 02FB 0001 00000001 A560 0000

     NAME CH LOC
      BPA 1B A11
      BPB 1B A10
      BPC 1B A08
.
.
.
```

Figure B-3. Error Data Listing

TERMINATING DPS AND RELOADING CP-R

DPS may be terminated prematurely by typing HALT at any time. When DPS completes its test cycle or is halted, it types:

H>

After DPS has halted, return the MAIN MODE rotary switch on the System Control Panel to the OFF position. The MAINTENANCE MODEstatus indicatorwill be extinguished.

If the normal load-and-go diagnostic procedure has been used, CP-R may be reloaded by booting from the disk. If full testing of disk surfaces was performed, CP-R must be reloaded and the file structure must be restored from the SAVE ALL tape.

Telefile
COMPUTER PRODUCTS, INC.

17131 Daimler Street, CA 92714
(714) 250-1830