60455980

CONTROL DATA

CONCURRENT MAINTENANCE LIBRARY REFERENCE MANUAL

CDC® OPERATING SYSTEMS:
NOS
NOS/BE

REVISION RECORD

REVISION	DESCRIPTION
A (06-15-78)	Manual released.
B (08-14-79)	Manual revised. Because of extensive changes to this manual, chart tape and dots have not been used, and all pages reflect the latest revision level. This revision obsoletes all previous editions.
C (12-21-79)	Manual revised. Corrections made to the ASSIGN and PARAM directives. Corrections also made to the error messages.
D (10-15-80)	Manual revised. Additions include execution directives for TFL, HPA option, and DSA; a description of error code report; a description of detailed status analysis; and SKEDULR. Miscellaneous editorial and technical corrections have also been made.
E (05-15-81)	Manual revised at CML 3.2 level 126. Additions include HPA support of Detailed Status Analyzer, Multimainframe, LCN, and MSS; and support of FSC and ESM. Corrections have also been made to the engineering file entries. This revision obsoletes all previous editions.
F (09-24-81)	Manual revised at CML 3.2 level 130. Additions include HPA support of MMF, LCN and FSC. This manual supports the CYBER 170 model 825. This revision obsoletes all previous editions.
G (04-15-82)	Manual revised at CML 3.2 level 136. Miscellaneous corrections made to HPA test examples. This manual supports CYBER 170 Model 800 Computer Systems. Trade-secret proprietary designation is removed and manual is returned to hard copy. Due to extensive changes, change bars and dots are not used, and all pages reflect the latest revision. This revision obsoletes all previous editions and replaces publication 60458370.
H (09-28-82)	Manual revised at CML 3.2 level 142. HPA section removed and RDF section added. FLM, DH8, FEP, FSM, FHC, and FHD examples of running jobs added. This revision obsoletes all previous editions.
J (05-16-83)	Manual revised at CML 3.2 level 150. Sections 1 and 2 are extensively revised. Appendixes A, B, and C are added.
K (10-10-83)	Manual revised at CML level 155. New CML installation procedures added to section 1. Miscellaneous technical changes also made.
L (05-11-84)	Manual revised at CML level 161. Section 5, Engineering File, has been eliminated; the information in that section is included in the HPA User Reference Manual. Additions include a new section 5, Microcode Verification Process, and appendix D, MALET Output Listing Example. MTPLOT information is included in appendix E of the HPA User Reference Manual. Miscellaneous technical changes also made.
M (10-10-84)	Manual revised at CML level 167. Section 3, Remote Diagnostic Facility (RDF) has been eliminated; the information in that section is included in the NOS and NOS/BE On-line Maintenance Software reference manuals. CML Installation Utilities are added to section 1. Miscellaneous technical changes also made.
N (06-13-85)	Manual revised at CML level 173. Appendix B extensively revised. Miscellaneous technical changes also made.
P (11-07-85)	Manual revised at CML level 179. Validation denied error messages added. Miscellaneous technical changes also made.

Publication No.
60455980

REVISION LETTERS I, O, Q, S, X AND Z ARE NOT USED.

©1978, 1979, 1980, 1981, 1982, 1983, 1984, 1985
by Control Data Corporation
All rights reserved
Printed in the United States of America

Address comments concerning this manual to:

Control Data Corporation
Publications and Graphics Division
4201 North Lexington Avenue
St. Paul, Minnesota 55126-6198

or use Comment Sheet in the back of this manual.

LIST OF EFFECTIVE PAGES

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

PAGE	REV
Front Cover	-
Title Page	-
2	P
3/4	P
5	P
6	N
7	P
8	P
9	N
1-1	J
1-2	J
1-3	J
1-4	L
1-5	N
1-6	N
1-7	L
1-8	K
1-9	N
1-10	L
1-11	K
1-12	K
1-13	L
1-14	L
1-15	P
1-16	P
1-16.1/1-16.2	P
1-17	L
1-18	N
1-19	N
1-20	N
1-20.1/1-20.2	N
1-21	P
1-22	M
1-23	P
1-24	L
1-25	L
1-26	N
1-27	N
1-28	N
1-29	M
1-30	M
1-31	M
1-32	M
1-33	M
1-34	P
1-35	P
1-36	P
1-37	P
1-38	P
1-39	P
2-1	J
2-2	N
2-3	N
2-4	J
2-5	J
2-6	J
2-7	J
2-8	J
2-9	J

PAGE	REV
2-10	J
2-11	J
2-12	L
2-12.1/2-12.2	L
2-13	J
2-14	N
2-15	P
2-16	J
2-17	J
2-18	J
2-19	P
2-20	J
2-21	J
2-22	N
2-23	J
2-24	J
2-25	J
2-26	L
2-27	L
2-28	J
2-29	M
2-30	J
2-31	J
2-32	J
2-33	J
2-34	J
2-35	J
2-36	J
2-37	L
2-38	L
4-1	M
4-2	G
4-3	N
4-4	N
4-5	G
4-6	N
4-7	L
4-8	G
4-9	G
4-10	G
4-11	G
4-12	G
4-13	G
4-14	G
4-15	G
4-16	G
4-17	G
4-18	G
4-19	G
4-20	G
4-21	G
4-22	G
4-23	G
4-24	G
4-25	G
4-26	G
4-27	G
4-28	G
4-29	G

PAGE	REV
4-30	G
4-31	G
5-1	N
5-2	N
5-3	N
5-4	N
6-1	L
6-2	L
6-3	L
6-4	L
6-5	L
A-1	J
A-2	J
A-3	J
A-4	P
A-5	P
A-6	P
A-7	P
A-8	P
A-9	P
A-10	P
A-11	P
B-1	N
B-2	N
B-3	N
B-4	N
B-5	P
B-6	N
B-7	N
B-8	N
B-9	P
C-1	K
C-2	K
D-1	L
D-2	L
D-3	L
D-4	L
D-5	L
D-6	L
Comment Sheet	P
Back Cover	-

PAGE	REV

PAGE	REV

PREFACE

The CONTROL DATA® Concurrent Maintenance Library (CML) is a collection of programs developed to improve the support functions for Engineering Services personnel in the diagnosis and repair of malfunctions occurring in CDC® CYBER 70, CDC CYBER 170/180 Computer Systems, and 6000 Computer Systems. Revisions to existing programs and new programs can be added as required.

RELATED PUBLICATIONS

The following publications are related to CML and are available through the Literature Distribution Services catalog.

Control Data Publication	Publication Number
CIP User's Handbook	60457180
COMPASS Version 3 Reference Manual	60492600
Concurrent Maintenance Library (CML) Customer's Guide	60456860
CYBER Record Manager Advanced Access Methods Version 2 Reference Manual	60499300
FORTRAN Extended Version 5 Reference Manual	60481300
Hardware Performance Analyzer (HPA) User Reference Manual	60459460
INTERCOM Interactive Procedures Guide	60495200
INTERCOM Version 4 Reference Manual	60494600
INTERCOM Version 5 Interactive Command Summary	60455840
INTERCOM Version 5 Reference Manual	60455010
MALET Reference Manual, NOS and NOS/BE	60456020
Network Products Interactive Facility Version 1 Reference Manual	60455250
Network Products Interactive Facility Version 1 User's Guide	60455260
Network Products Network Terminal User's Instant Manual	60455270
NOS On-Line Maintenance Software Reference Manual	60454200
NOS Version 1 Installation Handbook	60435700
NOS Version 1 Operator's Guide	60435600

Control Data Publication	Publication Number
NOS Version 1 Reference Manual, Volume 1	60435400
NOS Version 1 Reference Manual, Volume 2	60445300
NOS Version 1 Terminal User's Instant Manual	60435800
NOS Version 1 Time-Sharing User's Guide	60436400
NOS Version 1 Time-Sharing User's Reference Manual	60435500
NOS Version 2 Installation Handbook	60459320
NOS Version 2 Operations Handbook	60459310
NOS Version 2 Reference Set, Volume 1 Introduction to Interactive Usage	60459660
NOS Version 2 Reference Set, Volume 2 Guide to System Usage	60459670
NOS Version 2 Reference Set, Volume 3 System Commands	60459680
NOS Version 2 Reference Set, Volume 4 Program Interface	60459690
NOS Version 2 Analysis Handbook	60459300
NOS Version 2 Systems Programmer's Instant	60459370
NOS/BE On-Line Maintenance Software Reference Manual	60453900
NOS/BE Version 1 Installation Handbook	60494300
NOS/BE Version 1 Operator's Guide	60493900
NOS/BE Version 1 Reference Manual	60493800
NOS/BE Version 1 System Programmer's Reference Manual, Volume 1	60494100
NOS/BE Version 1 System Programmer's Reference Manual, Volume 2	60457370
FORTRAN Version 5 Reference Manual	60481300
UPDATE Version 1 Reference Manual	60449900
Binary Maintenance Log (BML) Message Formats	60459940

DISCLAIMER

These products are intended for use only as described in this document. Control Data cannot be responsible for the proper functioning of undescribed features or undefined parameters.

CONTENTS

1. INTRODUCTION 1-1

Traditional Maintenance Approach 1-1
On-Line Maintenance Approach 1-2
Concurrent Maintenance Library (CML) 1-4
Maintenance Application Language for Equipment Testing (MALET) 1-4
NOS, NOS/BE Remote Diagnostic Facility (RDF) Interactive Procedures 1-4
Input/Output Confidence Level Tests 1-5
Engineering File 1-5
CML Tape 1-5
CML Installation 1-6
Resource Requirements (NOS) 1-7
Magnetic Tape/Permanent File Requirements 1-7
Validation Requirements 1-8
Resource Requirements (NOS/BE) 1-10
Magnetic Tape/Permanent File Requirements 1-10
Field Length and Mass Storage Requirements 1-10
Preparatory Procedures 1-11
Gather Installation Materials 1-11
Read Documentation 1-11
Plan CML Installation 1-11
Create NOS or NOS/BE Job Deck 1-13
Perform Local Modifications/ Corrections Code 1-14
CML Installation Procedures 1-16
CML Installation Notes (NOS) 1-16
CML Installation Notes (NOS/BE) 1-16.1
Post-Installation Procedures 1-17
Permanent File Cleanup 1-17
Installation Parameters 1-17
Parameter Formats 1-17
Hardware Configuration Parameters 1-18
Software/Operational Environment Parameters 1-21
CML Installation Jobs Descriptions (CML1 through CML6) 1-24
Binary Installation 1-25
Interactive Mode 1-26
Batch Mode 1-27
From the System Console 1-28
CML Installation Utilities 1-28
Execution Modes 1-29
Interactive Execution 1-29
Batch Job Execution 1-32

2. MALET LANGUAGE 2-1

MALET Elements 2-1
Executive 2-2
Compiler 2-3
Executive/PP Driver Interface 2-4
PP Driver 2-4
Product Overlays to PP Driver Interface 2-5
Low-Level I/O Language Commands 2-5
High-Level I/O Language Commands 2-6
Non-I/O Language Commands 2-6
MALET Local Files 2-6
Structure and Organization 2-7
Module 2-7
Registers and Buffers 2-8
Executive and Diagnostic Switch Registers 2-9
Memory Requirements 2-10
Data Organization 2-11
MALET Program Relationships 2-12
MALET Language Format 2-12.1
Operational Procedure 2-13
Batch Mode Example 2-15
System Console Mode Example 2-15
Terminal Mode Example 2-16
MALET Executive Directives 2-16
ASSIGN or A Directive 2-16
AUTO Directive 2-19
CLRDS Directive 2-19
CLRSW Directive 2-20
COMPILE or C Directive 2-20
DEVICES Directive 2-21
DROP Directive 2-21
DUMP Directive 2-23
GO or G Directive 2-23
LINE or L Directive 2-23
LIST Directive 2-24
LOAD Directive 2-24
MALET Directive 2-25
MODULES Directive 2-25
MOVE or M Directive 2-27
PARAM or P Directive 2-28

PICTURE Directive 2-30
PPU Directive 2-31
PUNCH Directive 2-31
PURGE Directive 2-31
RUN or R Directive 2-32
SCRATCH Directive 2-34
SEND Directive 2-34
SETDS Directive 2-35
SETSW Directive 2-36
SOURCE Directive 2-36
STOP or S Directive 2-38
+ Directive 2-38
- Directive 2-38

3. REMOTE DIAGNOSTIC FACILITY - Section Deleted (See NOS OLM RM Pub. No. 60454200 and NOS/BE OLMS RM, Pub. No. 60453900) 3-1

4. INPUT/OUTPUT CONFIDENCE LEVEL TESTS 4-1

TIO 4-1
Operating Procedures 4-1
Parameters 4-1
Sample Jobs 4-3
Parameter Error Messages 4-4
SEQ and RAN Error Messages 4-5
Installation Deck 4-6
Test File Record Format 4-6
REGEN 4-7
Modes of Operation 4-13
Generate Mode 4-13
Copy Mode 4-15
Regenerate Mode 4-17
Return Mode 4-19
Runpass Mode 4-19
Isolate Mode 4-19
Operating Procedures and Execution Description 4-20
Sense Switches 4-21
Sample Jobs 4-22
Messages 4-24
Dayfile Messages 4-24
Output File Messages 4-24
Error Conditions 4-27
Error Messages 4-28
Error Log 4-31

5. MICROCODE VERIFICATION PROCESS 5-1

MCDUMP Utility 5-1
MCDUMP Formats 5-2
MCCOMP Utility 5-3
MCCOMP Examples 5-3
Verification Process 5-4

6. SKEDULR 6-1

Installing SKEDULR 6-1
Creating Jobs to Execute 6-2
Creating the Job Schedule 6-3
Executing SKEDULR - NOS 6-4
Executing SKEDULR - NOS/BE 6-4

APPENDIXES

A. EXECUTIVE RUNNING AND ERROR MESSAGES A-1

MLD Messages A-9
SBP Messages A-10
CVL Messages A-10

B. MALET UTILITY DECK STRUCTURES B-1

MALET Modules - NOS B-1
MALET Modules Documentation - NOS B-2
MALET Modules - NOS/BE B-2
MALET Modules Documentation - NOS/BE B-3
Examples of Running Jobs B-3

C. ACCESS LEVELS C-1

D. MALET OUTPUT LISTING SAMPLE D-1

FIGURES

1-1 User Time Versus Maintenance Time 1-1
1-2 User Time Versus On-Line Diagnostics Time 1-2
1-3 Maintenance Flowchart 1-3
1-4 CMLUTIL Menu 1-30
1-5 Sample Batch Job 1-32
2-1 Module Buffer Memory Format 2-3
2-2 MALET CPU Memory Requirements 2-10
2-3 MALET PP Memory Requirements 2-11
2-4 MALET Program Relationships 2-12
2-5 Sample K/L Display for DEVICES Directive 2-22
2-6 Sample K/L Display for MALET Directive 2-26
2-7 Sample K/L Display for MODULES Directive 2-27
2-8 Sample K/L Display for PARAM Directive 2-29
2-9 Sample K/L Display for SOURCE Directive 2-37
4-1 REGEN Modes 4-8
4-2 PVC Checksumming 4-10
4-3 FST Flowchart 4-11
4-4 ALX Flowchart 4-12
4-5 Generate Mode Flowchart 4-14
4-6 Copy Mode Flowchart 4-16
4-7 Regenerate Mode Flowchart 4-18
D-1 MALET Source Listing: List Option I D-3
D-2 MALET Source Listing: List Option L D-4
D-3 MALET Global Cross-Reference Map: List Options L and MAP D-6

TABLES

1-1 NOS Installation Jobs Resource Requirements 1-7
1-2 NOS/BE Installation Jobs Resource Requirements 1-10
1-3 Hardware Configuration Parameters 1-18
1-4 Software/Operational Environment Parameters 1-21
1-5 Installation Utilities and Functions 1-31
1-6 Batch Job Parameters 1-33
2-1 Executive/PP Driver Interface Memory Distribution 2-4
2-2 MALET Test Execution Sequence 2-13
4-1 Input Control Cards 4-20
4-2 Error Messages 4-29
4-3 Key to Error Messages 4-30
C-1 Access Levels C-1

INTRODUCTION 1

To understand the purpose and advantages of on-line maintenance, it is necessary to understand basic system maintenance concepts. This section presents these concepts, lists the goals of the hardware maintenance features, and shows how these goals are accomplished.

TRADITIONAL MAINTENANCE APPROACH

The traditional approach to system maintenance assumes that user time and preventive maintenance (PM) time are employed solely for the purposes indicated by their names. It further assumes that failures will occur, for which emergency maintenance (EM) will be necessary.

However, time taken for EM is irrecoverable user time, as is the time between system failure and failure detection [mean time to detection (MTTD)]. MTTD represents rerun time for the user and is added to EM time to find total time lost to the user (figure 1-1).

Part of the PM time is traditionally used running diagnostic programs that have error detection as the goal. It is hoped that enough PM time remains for error correction. If not, more user time must be taken for PM to ensure effective system operation.

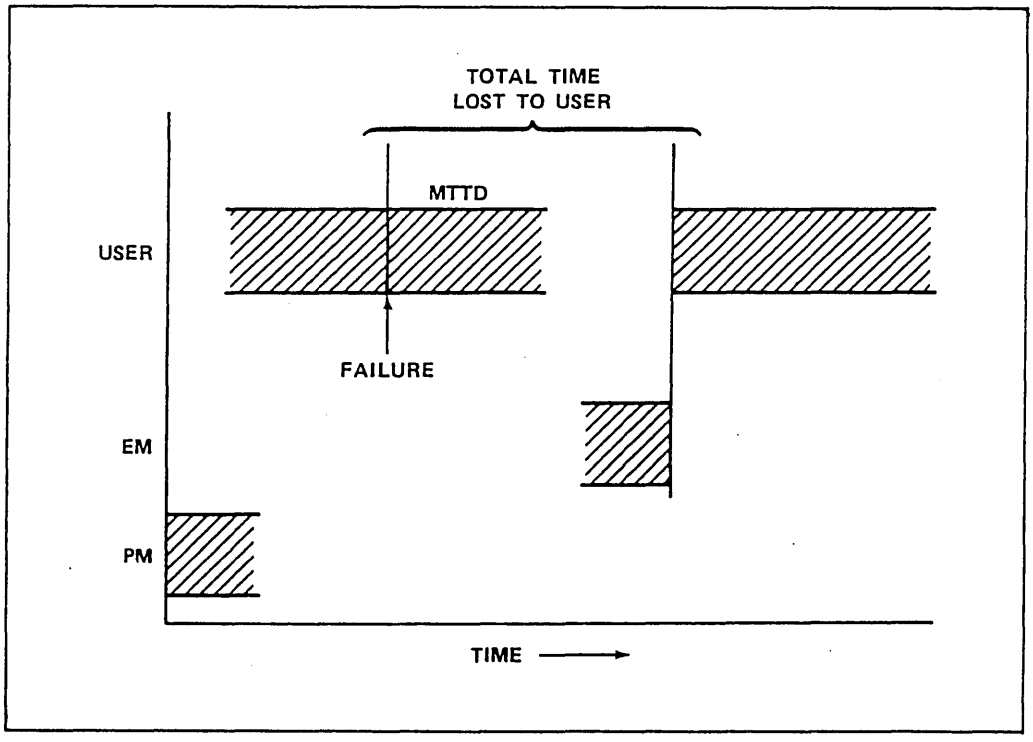

Figure 1-1. User Time Versus Maintenance Time

ON-LINE MAINTENANCE APPROACH

The philosophy of concurrent or on-line maintenance concerning user time and PM time differs from the traditional approach. PM time is used for prevention of failures rather than detection. A small amount of user time is employed for checking correct system operation by the automatic execution of on-line diagnostics at preselected intervals (figure 1-2).

The following three goals summarize the purpose of on-line maintenance.

- Increased system availability: the ability to detect and repair faults in a real-time manner allows greater user access.
- Increased system reliability: constant checking of system viability shakes loose potential failures and hidden problems, resulting in an increasingly reliable system.
- Increased user confidence: the user always knows the level of system operability.

These goals are achieved because the on-line maintenance system includes the following interacting features.

- Real-time failure detection, isolation, and correction.
- More efficient use of PM periods.
- Minimal emergency downtime.

On-line diagnostics, which are run at preselected intervals, detect failures shortly after they occur, significantly reducing the time between failure and detection. Failure information is entered in the dayfile and presented to system personnel in the form of a printout. It is also entered in the engineering file for future reference and analysis. The file can be used as a running account of system operation, enabling the user to easily find the time and nature of failures. These factors combine to save a considerable amount of rerun time.

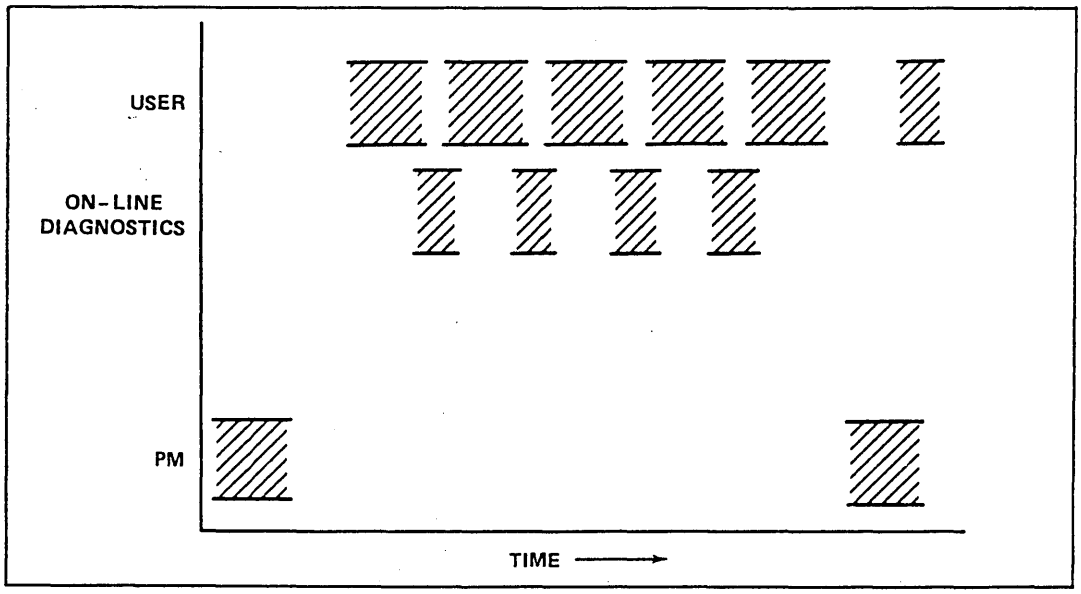

Figure 1-2. User Time Versus On-Line Diagnostics Time

When a failure is detected, isolation and correction can be attempted using additional diagnostics. The information logged in the engineering file can be used to aid in isolating the failure. If it is impossible to isolate and correct the failure on-line, it then must be determined whether the system can continue to operate effectively with the failure.

As the flowchart (figure 1-3) illustrates, the goal is to maintain effective system operability despite occasional hardware malfunctions.

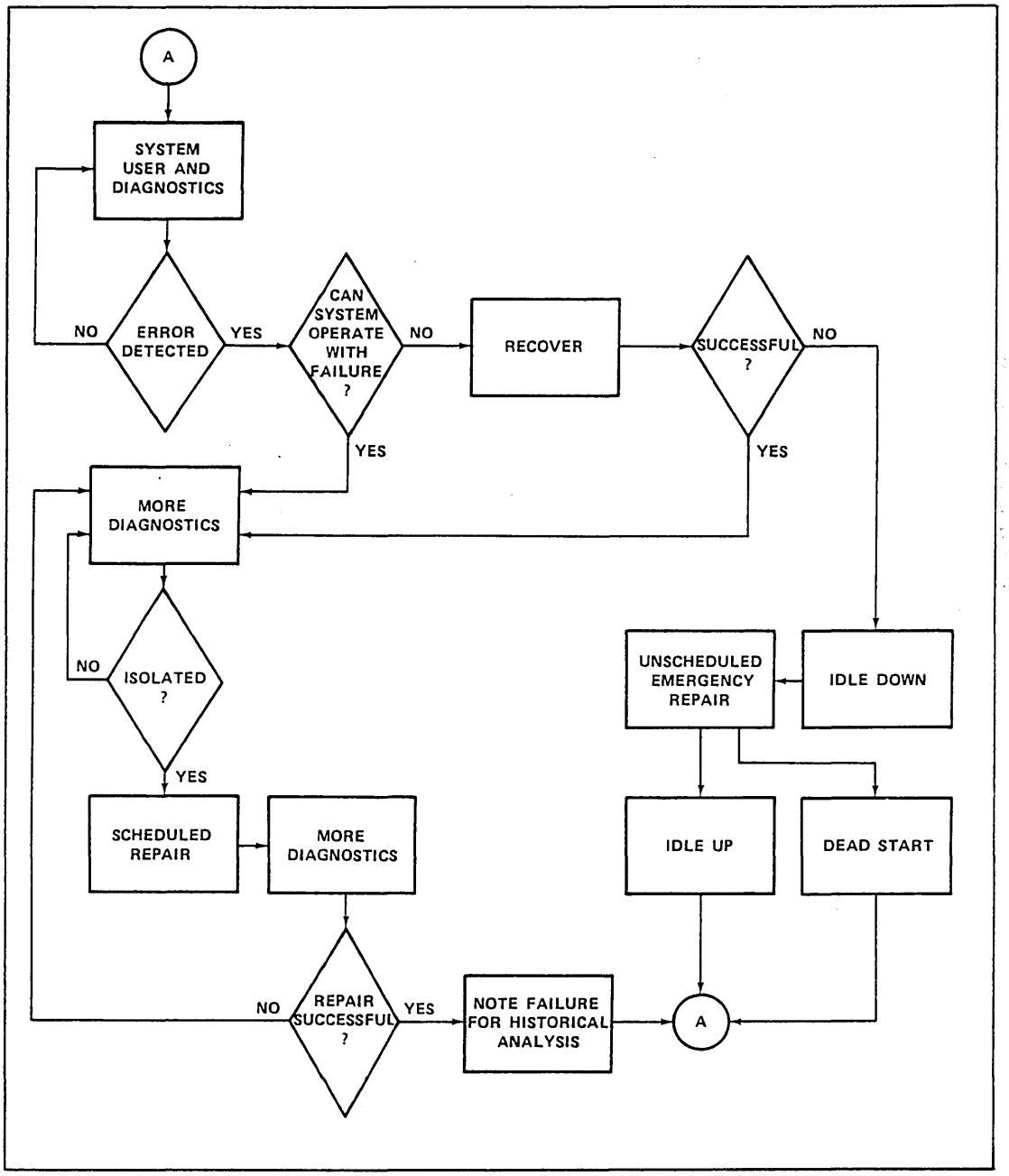

Figure 1-3. Maintenance Flowchart

PM time is not used for EM. PM time is used to prevent failures by performing regularly scheduled procedures, including replacement of suspected components. On-line maintenance releases PM time from the running of diagnostics (failure detection). It also provides, through its logging capability, system status information valuable to efficient planning and use of PM periods.

On-line maintenance recognizes and curbs system degradation before collapse occurs. It detects minor malfunctions and takes proper action before a fatal system failure results. Degradation in a system is inevitable, but it can be controlled by proper application of on-line maintenance.

If system failure does occur and EM is required, the engineering file contains information that aids system personnel in the isolation and correction of the failure.

In summary, user confidence in the system depends upon the knowledge that the system is reliably available and operating at maximum efficiency.

On-line maintenance ensures confidence. Down-time is significantly reduced, if not eliminated, because the system recognizes and reports degradation before system failure. PM time is used for failure prevention, as intended, rather than for failure detection, as in off-line diagnostics. Failures, whether minor or major, are detected much sooner than they would be by the PM diagnostics. Early detection greatly reduces the necessity of job reruns, which increases availability. Because on-line maintenance can log malfunction information in the engineering file and can point out the information shortly after the failure occurs, the state of the system is known at all times. Waiting until PM to learn if jobs are successful is not necessary. Test results are more accurate and believable because the system is tested in the user environment under the operating system rather than in an off-line mode.

CONCURRENT MAINTENANCE LIBRARY (CML)

CML comprises four major types of diagnostics described in the following sections of this manual.

Section 2 MALET Language

Section 3 Remote Diagnostic Facility

Section 4 Input/Output Confidence Level Tests

Section 5 Microcode Verification Process

MAINTENANCE APPLICATION LANGUAGE FOR EQUIPMENT TESTING (MALET)

MALET is a computer maintenance language used to write diagnostic application-type programs that perform peripheral device fault isolation.

NOS, NOS/BE REMOTE DIAGNOSTIC FACILITY (RDF) INTERACTIVE PROCEDURES

NOS, NOS/BE provides customer engineers with the ability to perform maintenance software functions interactively. RDF can be used from any location by dialing the computer's telephone number and is available only at sites with models 810, 815, 825, 830, 835, 845, or 855 computer systems.

INPUT/OUTPUT CONFIDENCE LEVEL TESTS

The tests comprise the TIO on-line check of I/O operations and the REGEN on-line test program. The TIO assesses the performance of mass storage devices, tapes, and disks. The REGEN program provides validation of system data paths, verification of data relative to hardware and software, and verification of the central memory and the capability of the central processor unit to execute further diagnostic tests.

ENGINEERING FILE

The engineering file provides data for effective preventive maintenance through a running account of system performance. This is provided by collecting and recording data on system hardware malfunctions detected by the operating system.

CML TAPE

The Concurrent Maintenance Library (CML) executes under NOS and NOS/BE operating systems.

The CML tape is labeled (VSN=CMLVSN), recorded in Scope Internal (SI) format, and consists of the following logical files:

FILE 1	CML installation procedures, which are written in CYBER Control Language (CCL).
FILE 2	CML Program Library, in Update format with an asterisk (*) as its master control character.
FILE 3	BUCAL binary of the buffer controller diagnostic MY8 (MY8BC) in a form compatible with the downline load requirements of MALET MY8.
FILE 4	BUCAL binary of the buffer controller diagnostic BCX (BCXBC) in a form compatible with the downline load requirements of MALET BCX.
FILE 5	TFL 2550 scratch file binaries.
FILE 6	CR1 scratch file binaries.
FILE 7	Binaries of the LCN tests (NDM, NDP, NDT) in a form compatible with the downline load requirements.
FILE 8	Binary file of MCCOMP control store conversion program suitable for execution under NOS.
FILE 9	Binary file of MCDUMP control store conversion program suitable for execution under NOS.
FILE 10	Binary file of MTPLOT and DSPLOT suitable for execution under NOS.
FILE 11	Binary file of SKEDULR suitable for execution under NOS.
FILE 12	Binary file of MLTDOC suitable for execution under NOS.
FILE 13	MALET diagnostic binaries suitable for loading under NOS.

CML INSTALLATION

The CML installation procedures contained on deck CMLINST are designed to perform an initial installation of the CML components into the NOS or NOS/BE operating system, or upgrade the level of existing CML components contained on the system. The major functions performed by these procedures include:

- Assembly of MALET, HPA, REGEN, and TIO,
- Assembly of the PP routines MLD and SBP,
- Optional incorporation of these binaries into the running system,
- The building of a new deadstart tape that includes these binaries (NOS only),
- Assembly of SKEDULR, DSPLOT, MLTDOC, and MTPLOT, and retention of the binaries on permanent files,
- Optional compilation of microcode utilities MCCOMP and MCDUMP and retention as permanent files,
- Compilation of the MALET diagnostics necessary to support the specified equipment configuration,
- Optional generation of hardcopy documentation for all compiled MALET diagnostics, and
- Incorporation of local modifications or corrective code into any of the assembled/compiled binaries

When called into execution, the procedures create installation jobs (CML1 through CML6) that are tailored to the specific hardware, software, and operational environment that the user defines. The jobs are saved as permanent files for subsequent execution.

The procedures are called into execution directly from the CML tape by using a BEGIN statement to pass the hardware, software, and operational environments to the procedures. The procedures then execute and create the CML installation jobs, which are tailored to meet the specific requirements of the installation. Following this, a ROUTE statement starts execution of the first installation job.

Execution of the installation jobs is begun by routing the first job (CML1) to the input queue. The remaining installation jobs are routed at the proper time in the sequence. These jobs are designed to make optimum use of system resources, thus minimizing installation time.

Typical execution times for NOS, reflecting no other activities in the system, are as follows:

CPU	Wall-Clock Time
CYBER 170-825	49 minutes

RESOURCE REQUIREMENTS (NOS)

Magnetic Tape/Permanent File Requirements

Installation jobs require the magnetic tapes or their permanent file equivalents listed in table 1-1 to be available. All magnetic tape requests issued by the installation jobs contain a VSN parameter, which causes the job to be rolled out until the request is satisfied. The tape requests can then be viewed via the E,P console display. All magnetic tapes can be mounted and the VSN assigned prior to the initiation of the installation process, thus avoiding the need to use the E,P display.

Table 1-1. NOS Installation Jobs Resource Requirements (Sheet 1 of 2)

Magnetic Tape VSN	Permanent File	Used In Job	Comments
CMLVSN	N/A	CML1	CML Program Library. Tape must be 1600 bpi, labeled, SCOPE Internal format, without the Write Enable ring. Permanent file CMLPL is created from the Update Program Library (file 2) for use by subsequent installation jobs. Depending upon system configuration, various other files are copied from the tape and retained as permanent files. File CMLPL may be purged when installation is complete.
REL1A	User Defined	CML2, CML3	REL1A Program Library. Tape must be NOS Internal format, labeled, without the Write Enable ring. The default VSN of REL1A and the default density of 1600 bpi may be overridden through use of the OPLVSN and OPLDEN parameters. The installation job (CML2 or CML3) will create random access permanent file SYSOPL from the tape, which may be purged when installation is complete. Once permanent file SYSOPL is created, a rerun of the creating job (CML2 or CML3) will determine that the file already exists and will not request the tape again. If REL1A (frequently referred to as SYSOPL or OPL) is available on disk, it may be used in lieu of the magnetic tape. It must be a random access program library file in MODIFY format. The disk copy can be used by passing the permanent file name in parameter OPLPFN. The default user name for this file is LIBRARY, which may be changed through use of the parameter OPLUNID. If a password is required, it can be passed in parameter OPLPW. If the file exists on an auxiliary device, the pack name and device type are passed in parameters OPLPAK and OPLDTSET, respectively. If OPLDTSET is not specified, the default auxiliary device type defined at system installation will be assumed.

Table 1-1. NOS Installation Jobs Resource Requirements (Sheet 2 of 2)

Magnetic Tape VSN	Permanent File	Used In Job	Comments
OLDDST	N/A	CML4	Old Deadstart Tape. Tape must be NOS Internal format, unlabeled, with no Write Enable ring. The requested density is the installation default. This tape is used as the basis for the generation of a new deadstart tape (NEWDST), which will contain the CML software.
NEWDST	N/A	CML4	New Deadstart Tape. The tape is written in NOS Internal format at the installation-defined default density. The tape must be unlabeled and have a Write Enable ring installed.

If a SYSEDIT is to be performed, job CML4 issues a dummy request for lfn S. This request is simply a warning to the operator of an impending SYSEDIT, and allows the network to be dropped and the system placed in DEBUG mode. When this has been accomplished, the request can be assigned to a disk or a null device.

Validation Requirements

Execution of the CML installation procedures requires the user to have the following minimum validation limits and permissions. All values are based on the released default values contained in system common deck COMSACC. All entries and numeric values listed are in octal notation. Not all entries apply to every operating system version and level. Refer to the system maintenance reference manual applicable to the system level on which the installation is to be performed.

MODVAL Parameters	Validation or Permission
AP=IAF, AP=RBF	User must be validated for IAF and/or RBF if installation is to be accomplished using interactive or batch terminals.
AW=CLPF, AW=CSPF, AW=CSOJ, AW=CASF, AW=CAND, AW=CSRP	The user must be able to create direct (CLPF) and indirect (CSPF) access permanent files, access system files (CASF), and request nonallocatable devices (CAND). If REL1A (SYSOPL) resides on an auxiliary device, the user must be validated to issue auxiliary device commands. If a SYSEDIT is to be performed, the user must also be validated for system origin privileges.
CC=77B	Validation for unlimited batch command processing.
CM=34B	Validation for a maximum of 163,700 words of central memory (CM).

MODVAL Parameters	Validation or Permission
	NOTE This is the maximum amount of central memory anticipated to be required on a very large operating system (many products installed). Less field length may be required on smaller systems. If the user maximum CM validation is less than that indicated here, the installation job CML4 will dynamically determine the maximum amount available and request that amount.
CS=4	Validation for a cumulative maximum of 10,000 PRUs of indirect access permanent file space.
DB=7	Validation for an unlimited number of jobs to exist concurrently in system.
DS=5	Validation for an individual direct access permanent file size of 50,000 PRUs.
FC=7	Validation for an unlimited number of permanent files.
FS=6	Validation for a maximum indirect access permanent file size of 300 PRUs. (The FS parameter should not be set equal to 7 as this may result in inefficient use of system resources.)
LP=77B	Validation for unlimited print output.
MS=20B	Validation for a maximum of 201,000 additional PRUs the user may allocate per job.
MT=1	Validation for one magnetic tape unit.
RP=1	Validation for one removable disk pack. [This validation is required only if RELIA (SYSOPL) is to be accessed and it resides on a removable device.]
SAV=CULT	Validation to allow the user to write unlabeled magnetic tapes. Required only on NOS 2.2 and later versions.
SAV=COLD	Validation to allow the user to use on-line diagnostics.
SL=77B	Validation for unlimited SRU accumulation.
TL=7	Validation for 7100 (octal) seconds of CPU time.

RESOURCE REQUIREMENTS (NOS/BE)

Magnetic Tape/Permanent File Requirements

The installation jobs require the magnetic tapes or their permanent file equivalents listed in table 1-2 to be available. All magnetic tape requests issued by the installation jobs contain a VSN parameter, which causes the requesting job to be rolled out until the request is satisfied. The tape requests can then be viewed via the P console display. All magnetic tapes can be mounted and the VSN assigned prior to the initiation of the installation process.

Table 1-2. NOS/BE Installation Jobs Resource Requirements

Magnetic Tape VSN	Permanent File	Used In Job	Comments
CMLVSN	N/A	CML1	CML Program Library. Tape must be 1600 bpi, labeled, with no Write Enable ring. Permanent file CMLPL is created from the Update Program Library (file 2) for use by subsequent installation jobs. Depending upon system configuration, various other files are copied from the tape and retained as permanent files.
PL1B	User Defined	CML3	PL1B Program Library. Tape must be of VSN and density as specified by parameters OPLVSN and OPLDEN, respectively, with no Write Enable ring. A permanent file copy (random program library) is made of the tape and retained as permanent file OPL1B under the IDENT specified by parameter CEID. This file may be purged when installation process is complete. Once permanent file OPL1B is created, any rerun of CML3 will determine that the file alreay exists and will not request the tape again. If PL1B is available on disk, it may be used in lieu of the magnetic tape. It must be a random access program library file in Update format. The disk copy can be used by passing the permanent file name and IDENT in parameters OPLPFN and OPLUNID, respectively. If a password is required, it can be passed in parameter OPLPW. If the file resides on a private pack, the VSN and setname can be passed in parameters OPLPAK and OPLDTSET, respectively.

Field Length and Mass Storage Requirements

Execution of the installation jobs requires a maximum of 120,000 words of central memory and up to 7000 (octal) central processor unit (CPU) seconds depending on the specific CPU model in use. Sufficient mass storage space should be available as follows.

Temporary file space (released at job termination): 200,000 PRUs

Permanent file space (released following installation): 200,000 PRUs

Retained permanent file space (MALET diagnostics, and so on): 5000 PRUs

These figures are estimates and may vary significantly depending on system configuration and specific parameters used during the installation process. The figures stated should be adequate for most installations.

The installation process also requires the use of one magnetic tape transport.

PREPARATORY PROCEDURES

Perform the following preparatory procedures before installing CML.

1. Gather Installation Materials.
2. Read Documentation.
3. Plan CML Installation.
4. Create NOS or NOS/BE Job Deck.
5. Perform Local Modifications/Corrections Code.

Gather Installation Materials

1. Locate the CML maintenance software kit provided by manufacturing for the applicable computer system. This kit contains a diagnostic software release bulletin (DSRB), a CML Program Library Tape, and documentation.
2. For NOS, a REL1A tape, system deadstart tape, and a scratch tape (which is suitable for use as a new system deadstart tape) must also be available. For NOS/BE, a PL1B tape must also be available.

Read Documentation

1. Read the DSRB furnished in the CML kit.
2. Read through the CML installation procedures later in this section.
3. Read service bulletins.

Plan CML Installation

Following are step-by-step examples that may help simplify the planning process. The examples indicate how the various parameters are entered in the BEGIN statement. Refer to the detailed descriptions of parameters to determine the impact of each. (None of the parameters is carried forward in the examples. Parameters are shown only to illustrate the specific parameter format.)

Plan your CML installation using the following guidelines.

		EXAMPLE:
1.	Determine the desired hardware configuration and enter the required keywords in the BEGIN statement.	BEGIN,CMLINST,CMLTAPE,,885,67X,2550,580,800,
2.	If the internal sort feature of HPA must be used, add HPSORT; else, leave blank.	BEGIN,CMLINST,CMLTAPE,885,67X,2550,580,800, HPSORT,
3.	If running in an ECS/ESM-based multimainframe configuration, add MMF; else, leave blank. (NOS only.)	BEGIN,CMLINST,CMLTAPE,885,67X,2550,580,800, MMF,
4.	If MALET mass storage access levels require modification, add NOTSECURE; else, leave blank.	BEGIN,CMLINST,CMLTAPE,885,67X,2550,580,800, NOTSECURE,
5.	If 8.5-inch printer paper is in use, add PAPER85; if standard 11-inch paper is in use, leave blank.	BEGIN,CMLINST,CMLTAPE,885,67X,2550,580,800, PAPER85,
6.	If the CML binaries are to be added to the running system, add SYSEDIT; else, leave blank.	BEGIN,CMLINST,CMLTAPE,885,67X,2550,580,800, PAPER85,SYSEDIT,
7.	If 8 lines-per-inch print density is desired for MALET, add 8LPI; else, leave blank.	BEGIN,CMLINST,CMLTAPE,885,67X,2550,580,800, PAPER85,SYSEDIT,8LPI,
8.	If hardcopy documentation is desired for MALET diagnostics, specify the type in MALDOC.	BEGIN,CMLINST,CMLTAPE,885,67X,2550,580,800, PAPER85,SYSEDIT,8LPI,MALDOC=EXT,
9.	For NOS/BE only, enter in parameter CEID the permanent file ident under which all permanent files are to be cataloged.	BEGIN,CMLINST,CMLTAPE,885,67X,2550,580,800, PAPER85,SYSEDIT,8LPI,MALDOC=EXT,CEID=CDCCE,
10.	If the system OPL (RELIA for NOS, PL1B for NOS/BE) must be loaded from tape, specify the VSN and density in OPLVSN and OPLDEN.	BEGIN,CMLINST,CMLTAPE,885,67X,2550,580,800, PAPER85,SYSEDIT,8LPI,MALDOC=EXT, OPLVSN=REL1A,OPLDEN=GE,
11.	If the system OPL is available on disk, specify the location using parameters OPLPAK, OPLPFN, OPLPW, OPLUNID, and OPLDTSET. Omit any parameter not required.	BEGIN,CMLINST,CMLTAPE,885,67X,2550,580,800, PAPER85,SYSEDIT,8LPI,MALDOC=EXT,OPLPFN=OPL, OPLUNID=LIBRARY,
12.	Enter the system version number in SYSVER.	BEGIN,CMLINST,CMLTAPE,885,67X,2550, 580,800,PAPER85,SYSEDIT,8LPI, MALDOC=EXT,OPLPFN=OPL,OPLUNID=LIBRARY, SYSVER=NOS6,

		EXAMPLE:
13.	Enter the required USER/CHANGE statement images. Under NOS/BE, accounting information, if required, is passed by the USER parameter and is incorporated into the installation jobs as the second control statement, right after the JOB statement. Additional accounting can be passed as the third statement in the installation job via the CHARGE parameter.	BEGIN,CMLINST,CMLTAPE,885,67X,2550,580,800, PAPER85,SYSEDIT,8LPI,MALDOC=EXT,OPLPFN=OPL, OPLUNID=LIBRARY,SYSVER=NOS6,USER=$USER, user,pass.$.

At this point, the BEGIN statement is complete and can be incorporated into a card deck or job file as indicated in the sample NOS installation job.

Create NOS or NOS/BE Job Deck

The following NOS and NOS/BE sample jobs illustrate how to create installation jobs. While the examples show how this can be done using a card deck, it can also be done by creating a similar job or procedure interactively or by using 026 at the system console.

NOS

In this NOS example, the hardware configuration has been defined to include 844 and 885 disk drives, a 2550 communications controller, a 67X magnetic tape subsystem, 580 line printers, and a 405 card reader. The mainframe is also identified as a CYBER 170/180 Models 810, 815, 825, 830, 835, 845, and 855 type (800 parameter). The software environment is defined as NOS 2.0 (NOS6 parameter). The operational environment is defined as using 8.5-inch printer paper, a SYSEDIT will be performed during the installation, and the user number, password, family, charge number, and project number have been supplied. The REL1A program library is available to the job on permanent file OPL under the user name LIBRARY.

```
JOB,T200.
  -User/Charge statements-
LABEL,CMLTAPE,VSN=CMLVSN,D=PE,F=SI,PO=U.
COPYBF,CMLTAPE,CMLPROC.
BEGIN,CMLINST,CMLPROC,844,885,2550,67X,580,405,800,
PAPER85,SYSEDIT,SYSVER=NOS6,
USER=$USER,1234567,password,FM.$,
CHARGE=$CHARGE,7654321,CDC.$,
OPLPFN=OPL,OPLUNID=LIBRARY.
ROUTE,CML1,DC=IN.†
6/7/8/9
```

†If execution of the created installation jobs is not desired at this time, omit this statement.

NOS/BE

In this NOS/BE example, the hardware configuration has been defined to include 844 and 885 disk drives, a 2550 communications controller, a 67X magnetic tape subsystem, 580 line printers, and a 405 card reader. The software environment is defined as NOS/BE 1.5 (NOSBE6 parameter). The operational environment is defined as using 8.5-inch printer paper, a system EDITLIB will be performed (SYSEDIT parameter) during the installation, and job accounting information has been passed via the USER parameter. The PL1B program library is available to the job on permanent file PL1B, under the permanent file IDENT of INSTALL.

```
JOB,T200,PE1.
  -Accounting Information-
LABEL,CMLTAPE,VSN=CMLVSN,R,D=PE,X=IU,NORING.
COPYBF,CMLTAPE,CMLPROC.
REWIND,CMLPROC.
BEGIN,CMLINST,CMLPROC,844,885,2550,67X,580,405,800,
PAPER85,SYSEDIT,SYSVER=NOSBE6,
USER=$ACCOUNT,1234567,password.$,
OPLPFN=PL1B,OPLUNID=INSTALL.
BEGIN,NEXTJOB,CMLINST,JOB=CML1,CEID=CDCCE.†
6/7/8/9
```

NOTE

Execution of the installation procedures and jobs requires that the procedures reside on permanent file CMLINST. The permanent file IDENT under which they are catalogued must be the same IDENT that is passed to the procedures via parameter CEID. If CEID is not specified, all permanent files created by the installation processes are catalogued using an IDENT of CDCCE.

Perform Local Modifications/Corrective Code

Local modifications or corrective code can be included in any of the assembled and/or compiled binaries. The modifications must be correction sets in Update format and exist in permanent files (indirect access files on NOS) according to the following specifications. The files must reside under the user's user index (NOS) or the permanent file ident specified by CEID (NOS/BE).

† If execution of the created installation jobs is not desired at this time, omit this statement.

Modifications to	Must Reside on	
CMLINST	INSTMDS	
MLD,SBP	MLDMODS	
MALET	MALMODS	
HPA	HPAMODS	
REGEN	REGMODS	
TIO	TIOMODS	
SKEDULR	SKEDMDS	
MLTDOC	MLTMODS	
MTPLOT	MTPMODS	
MCCOMP	MCCMODS	
MCDUMP	MCDMODS	
DSPLOT	DSPMODS	
DKDATA	DKDMODS	
(MALET diagnostics)		
ATC	ATCMODS	
BCX	BCXMODS	
CCM	CCMMODS	
CIU	CIUMODS	
CLM	CLMMODS	
CPE	CPEMODS	
CP1	CP1MODS	
CRE	CREMODS	
CRP	CRPMODS	
CR1	CR1MODS	
DFU	DFUMODS	
DMA	DMAMODS	
DTC	DTCMODS	
DTI	DTIMODS	
D44	D44MODS	
D88/DH8/DL8	D88MODS	(DH8 on NOS only)
ELR	ELRMODS	
FEP	FEPMODS	
FFU	FFUMODS	
FHC	FHCMODS	
FLM	FLMMODS	
FMC	FMCMODS	
FMD/FHD/FLD	FMDMODS	(FHD on NOS only)
FMU	FMUMODS	
FSD	FSDMODS	
FSM	FSMMODS	
FTP	FTPMODS	
F44	F44MODS	
F7X	F7XMODS	
F88	F88MODS	
ITU	ITUMODS	
LCI	LCIMODS	
LCM	LCMMODS	
LCN	LCNMODS	
LPE	LPEMODS	
LP1	LP1MODS	
MY8	MY8MODS	
NIP	NIPMODS	
NLM	NLMMODS	
PDP	PDPMODS	
RT3	RT3MODS	

Modifications to	Must Reside on
RT5	RT5MODS
TFE	TFEMODS
TFF	TFFMODS
TFL	TFLMODS
TT3	TT3MODS
T5X	T5XMODS
T6X	T6XMODS
T7X	T7XMODS

CML INSTALLATION PROCEDURES

1. Mount CML tape.

2. Initiate your NOS or NOS/BE job using a card deck or interactively through a terminal. The ROUTE statement starts execution of the first installation job. The remaining jobs are routed at the proper time in sequence.

After execution of one of these jobs is accomplished, all that is required is to mount and assign magnetic tapes, and to enter a GO if a SYSEDIT (EDITLIB) is to be performed.

Should errors occur during the execution of any of the installation jobs, the sequence can be restarted by manually routing to the input queue the job that failed. Care must be taken when restarting any job due to job interdependencies. If in doubt, it is recommended that the entire sequence be restarted. Note that the program libraries REL1A or PL1B are saved as permanent files by the jobs. Once they have been successfully loaded from tape, a restart of the job will use the permanent files in lieu of reloading them from tape.

CML Installation Notes (NOS)

If a new deadstart tape is not desired, job CML4 may be dropped after the old deadstart tape (VSN=OLDDST) has been requested. All remaining CML installation processes will have been initiated prior to this tape request. If the new deadstart tape is not created by CML4, it must be built through a user-supplied process that incorporates the routines contained on permanent files CMLBIN and MLDBIN.

The LIBEDIT directives used to build the new deadstart tape delete groups of records by using the from/to directive format. This requires the CML-related records on the old deadstart tape to be in contiguous blocks. Users are cautioned that adding nonstandard records into these blocks or rearranging the CML-related records on the deadstart tape could result in creation of an invalid new deadstart tape. Always review the list output of jobs CML2, CML3, and CML4 to verify that no non-CML-related records were deleted.

The CML installation job that creates the MALET diagnostics does not carry the MALET source code with the module binaries. Source code is not generally required for normal maintenance activities and it can waste permanent file space. Use the CML installation utility MALDIAG to compile any diagnostics for which source code is to be carried with the module binary.

The CML installation process replaces all existing copies of MALET diagnostics with the new versions. These new versions may not execute properly if used with the older version of MALET. Should the new version of CML not be implemented immediately, the customer engineer may be without any executable MALET diagnostics. In this situation, it is best to install CML using another username, then move the new MALET diagnostics to the old username when the new CML is implemented.

CML Installation Notes (NOS/BE)

The CML installation process purges the latest cycle of the MALET diagnostics before cataloging the new version of the diagnostic. These new versions may not execute properly if used with the older version of MALET. If the new version of CML is not implemented immediately, the customer engineer may be without any executable MALET diagnostics. In this situation, it is best to install CML using another permanent file ident, then purge the old diagnostics and change the ident of the new diagnostics when the new CML is implemented.

POST-INSTALLATION PROCEDURES

Permanent File Cleanup

The following permanent files may be purged after CML installation is complete.

CMLPL
DIAGLIB
SYSOPL (NOS only)
OPL1B (NOS/BE only)
CML1
CML2
CML3
CML4
CML5
CML6

INSTALLATION PARAMETERS

Parameter Formats

The parameters passed to the procedures already discussed must conform to the specific formats as follows.

Format	Applies to	
keyword	All hardware configuration parameters	
	HPSORT	
	MMF	(NOS only)
	NOTSECURE	
	PAPER85	
	SYSEDIT	
	8LPI	
keyword=value	MALDOC	
	OPLDEN	
	OPLPAK	
	OPLPFN	
	OPLPW	
	OPLUNID	
	OPLVSN	
	SYSVER	
	CEID	(NOS/BE only)
	OPLDTSET	
keyword=$value$	CHARGE	
	USER	

Hardware Configuration Parameters

Table 1-3 includes a description of the hardware configuration parameters used in the procedures discussed earlier in this section.

Table 1-3. Hardware Configuration Parameters (Sheet 1 of 4)

Parameter Name	Used In Job	Function
CYBERPLUS (NOS only)	CML5	Specifies that CYBERPLUS hardware is included in the configuration. Causes MALET diagnostic CRP to be compiled.
DEMA (NOS only)	CML2, CML5	Specifies that DEMA hardware is included in the configuration. Causes MALET diagnostics FHD, FHC, DH8, FEP, and FMU to be compiled. Also causes 885 detailed status analysis tables to be included in HPA. NOTE DEMA is not supported on NOS/BE.
ESM	CML5	Specifies that ESM hardware is included in the configuration. If SYSVER is NOS4, NOS5, or NOSBE6, causes MALET diagnostic ELR to be compiled.
FSCDISK	CML1, CML5	Specifies that a Federal Standard Channel driving disks is included in the configuration. Causes MALET diagnostics F44, F88, FLM, and FFU to be compiled. Also, NAD diagnostics NDM and NDP are loaded from the CML tape and saved by CML1.
FSCTAPE	CML1, CML5	Specifies that a Federal Standard Channel driving tape drives is included in the configuration. Causes MALET diagnostics F7X, FLM, and FSM to be compiled. Also, NAD diagnostics NDM and NDP are loaded from the CML tape and saved by CML1.
LCN	CML1, CML5	Specifies that LCN is included in the hardware configuration. Causes MALET diagnostics LCI, LCN, LCM, and NLM to be compiled. Also causes CML1 to load NDM, NDP, and NDT from the CML tape and save them.
PLATO (NOS only)	CML5	Specifies that the system is running PLATO and causes MALET diagnostic CIU to be compiled.
MASSTOR	CML1, CML5	Specifies that a CYBER Channel Coupler and MASSTOR are included in the configuration. Causes MALET diagnostics CCM and CLM to be compiled. Also, NAD diagnostics NDM and NDP are loaded from the CML tape and saved by CML1.

Table 1-3. Hardware Configuration Parameters (Sheet 2 of 4)

Parameter Name	Used In Job	Function
UNIBUS	CML5	Specifies that a CYBER/UNIBUS interface and DEC PDP11 are included in the configuration. Causes MALET diagnostic PDP to be compiled.
405	CML1, CML5	Specifies that a 405 card reader is included in the configuration. Causes MALET diagnostics CRE and CR1 to be compiled. Also causes CML1 to load and save CR1SCR from the CML tape.
415	CML5	Specifies that a 415 card punch is included in the configuration. Causes MALET diagnostics CPE and CP1 to be compiled.
512	CML5	Specifies that a 512 line printer is included in the configuration. Causes MALET diagnostics LPE and LP1 to be compiled.
580	CML5	Specifies that a 580 line printer is included in the configuration. Causes MALET diagnostics FTP and LPE to be compiled.
639 (NOS only)	CML5	Specifies that a 639 magnetic tape subsystem is included in the system configuration. Causes MALET utility ITU to be loaded and saved.
65X	CML5	Specifies that a 60X or 65X magnetic tape subsystem is included in the configuration. Causes MALET diagnostic T5X to be compiled.
66X	CML1, CML2, CML5	Specifies that a 66X magnetic tape subsystem is included in the configuration. Causes the MALET diagnostics T6X, BCX, and MY8 to be compiled. Also, causes BCXBC and MY8BC to be loaded from the CML tape and saved by CML1. Also causes the detail status analysis tables for 66X to be included in HPA.
67X	CML5	Specifies that a 67X magnetic tape subsystem is included in the configuration. Causes MALET diagnostics T7X and ATC to be compiled. Also causes the detail status analysis tables for 67X to be included in HPA.
800	CML6	Specifies that a CYBER 170-800 computer system (except models 865 and 875) is included in the configuration. Causes microcode utilities MCDUMP and MCCOMP to be compiled and retained as permanent files.
83X (NOS only)	CML5	Specifies that 834 (ISD1) or 836 (ISD2) disk drives are included in the configuration. Causes MALET diagnostics FSD and DFU to be compiled.

Table 1-3. Hardware Configuration Parameters (Sheet 3 of 4)

Parameter Name	Used In Job	Function
844	CML2, CML5	Specifies that 844 disk drives are included in the configuration. Causes the MALET diagnostic D44 to be compiled. If 7155 is also defined, MALET diagnostic DTC is also compiled. Also causes the detail analysis tables for 844s to be included in HPA.
885	CML2, CML5	Specifies that 885 disk drives are included in the configuration. Causes MALET diagnostics D88, FMC, FMD, and FMU to be compiled. Also causes the detail status analysis tables for 885s to be included in HPA.
885LS (NOS only)	CML5	Specifies that large sector 885 disk drives are included in the system configuration. Causes MALET diagnostics FLD and DL8 to be loaded and compiled. Will also cause FMC and FMU to be loaded and compiled if not done through the inclusion of other hardware parameters.
895	CML1, CML5	Specifies that an 895 disk and a CYBER Channel Coupler are included in the configuration. Causes MALET diagnostics DFU, CCM, and CLM to be compiled. Also, NAD diagnostics NDM and NDP are loaded from the CML and saved by CML1.
2550	CML1, CML5	Specifies that a 2550 communications controller is included in the configuration. Causes MALET diagnostics TFF and TFL to be compiled. Also, the ODS diagnostics will be loaded from the CML tape and saved by CML1.
2550100	CML5	Specifies that a 2550-100 communications controller is included in the configuration. Causes MALET diagnostic TFE to be compiled.
5X70	CML1, CML5	Specifies that a CYBER Channel Coupler and 5870 or 5970 Nonimpact Printer are included in the configuration. Causes MALET diagnostics NIP, CCM, and CLM to be compiled. Also NAD diagnostics NDM and NDP are loaded from the CML tape and are saved by CML1.
667X	CML5	Specifies that either a 6671, 6673, 6674, or 6676 multiplexer is included in the configuration. Causes MALET diagnostics TT3 and RT5 to be compiled.
7X54	CML1, CML5	Specifies that either a 7054, 7154, or 7152 controller is included in the configuration. Causes MALET diagnostics BCX and MY8 to be compiled. Also, BCXBC and MY8BC will be loaded from the CML tape and saved by CML1.

Table 1-3. Hardware Configuration Parameters (Sheet 4 of 4)

Parameter Name	Used In Job	Function
7155	CML5	Specifies that a 7155 disk controller is included in the configuration. This parameter is only required when the 844 disk drives are connected to the 7155 controller. If both 844 and 7155 are defined, MALET diagnostic DTC will be compiled in addition to FMD, FMC, and FMU.
7155C	CML5	Specifies that a 7155C disk controller is included in the configuration. This parameter is required when a 7155C type controller is used and DEMA is not defined.

Software/Operational Environment Parameters

Table 1-4 includes descriptions of the software/operational environment parameters used in the procedures discussed earlier in this section.

Table 1-4. Software/Operational Environment Parameters (Sheet 1 of 3)

Parameter Name	Used In Job	Function
CEID (NOS/BE only)	ALL	The NOS/BE permanent file IDENT under which all permanent files are to be catalogued.
CHARGE	CML1, CML2, CML3, CML4, CML5, CML5A, CML6	Charge statement image. Parameter must be passed as a literal (enclosed in $ $). Parameter required only if system requires charge numbers. Default: COMMENT.-NO CHARGE NUMBER SUPPLIED-
HPSORT	CML2	Keyword passed to evoke the internal HPA sort routine in lieu of SORT/MERGE 5. Omitted: HPA calls SORT/MERGE 5 to sort the data. Specified: HPA calls the internal routines to sort the data.
MALDOC	CML5, CML5A	Type of hardcopy documentation desired for compiled MALET diagnostics. Legal values: NONE (No documentation produced; default). EXT (External documentation produced). INT (Internal documentation produced). BOTH (Both internal and external documentation produced).
MMF (NOS only)	CML6	Keyword passed to cause SKEDULR to be assembled for use in a multimainframe environment. (Used in NOS only.) Default: SKEDULR assembled for use in single or nonshared mainframe environment.
NOTSECURE	CML3	Keyword passed to evoke a *YANK,SECURE directive during the Update of MLD. This alters the MALET mass storage access level capabilities. Omitted: *DEFINE,SECURE (recommended). Specified: *YANK,SECURE (not recommended).

Table 1-4. Software/Operational Environment Parameters (Sheet 2 of 3)

Parameter Name	Used In Job	Function
PAPER85	CML2	Keyword passed to cause MALET to be assembled for use with 212.5 mm (8.5 in) printer paper. Omitted: MALET assembled for use with standard 279.4 mm (11 in) printer paper.
OPLPAK=pack	CML2, CML3	Removable pack on which system OPL [RELIA (NOS) or PL1B (NOS/BE)] resides. Parameter not required if file is on nonremovable device. Omitted: System OPL on nonremovable device.
OPLPFN=opl	CML2, CML3	Permanent file name of system OPL [RELIA (NOS) or PL1B (NOS/BE)] if file exists. If passed, specified file is used. Parameters OPLUNID, OPLPAK, OPLDTSET, and OPLPW are used if specified. If OPLPFN is omitted, system OPL is loaded from tape and retained as permanent file. For NOS, RELIA is retained on direct access permanent file in user index. For NOS/BE, PL1B is retained on permanent file OPLIB under ident specified by CEID.
OPLPW=password	CML2, CML3	Password used for read access to OPLPFN. Parameter required only when no password is given.
OPLUNID=unid	CML2, CML3	User number (NOS) or permanent file ident (NOS/BE) of permanent file name passed in OPLPFN. Default: LIBRARY (NOS). No default for NOS/BE.
OPLVSN=vsn	CML2, CML3	The VSN of REL1A (NOS) or PL1B (NOS/BE) magnetic tape program library. Parameter not required when permanent file name passed in OPLPFN. Default: REL1A (NOS). PL1B (NOS/BE).
SYSEDIT	CML4, CML5, CML5A	Keyword specified if SYSEDIT (NOS) or EDITLIB (NOS/BE) is performed by CML4. If not specified, CML4 creates permanent file copy of compiler for use by jobs created by CML5A. If keyword not specified, CML5A uses MALET from system library after SYSEDIT (EDITLIB) performed.
OPLDEN=dn	CML2, CML3	Density of RELIA (NOS) or PL1B (NOS/BE) magnetic tape. Legal values for dn: GE (9-track GCR). PE (9-track phase). HD (9-track NRZI). HY (7-track NRZI, 800-bpi).

Table 1-4. Software/Operational Environment Parameters (Sheet 3 of 3)

Parameter Name	Used In Job	Function
OPLDTSET=Device Type =Setname	CML, CML3	The device type (NOS) or setname (NOS/BE) on which the system OPL resides. This parameter is required only if the file specified by OPLPFN resides on a private pack (NOS) or a pack other than the default set (NOS/BE). Legal values for device type: DB DG DK DQ DC DI DL DV DD DJ DM DW
SYSVER=ver	CML2, CML3, CML5, CML6	System version number. Used by CML2 and CML3 in assembly of MLD, SBP, and MALET. Used by CML6 in NOS to assemble SKEDULR. Used by CML5 to determine the need for MALET diagnostic, ELR. Default: None (this is a required parameter). Legal values for ver: NOS25 (NOS 2.5). NOS24 (NOS 2.4). NOS23 (NOS 2.3). NOS6 (NOS 2.0,2.1,2.2). NOS5 (NOS 1.4). NOS4 (NOS 1.3). NOSBE6 (NOS/BE 1.5). NOSBE5 (NOS/BE 1.4). NOSBE4 (NOS/BE 1.3).
USER	CML1, CML2, CML3, CML4, CML5, CML5A, CML6	User statement image (NOS) or first line of accounting information (NOS/BE). Parameter must be passed as a literal (enclosed in $...$) and is required in NOS unless the created jobs are manually modified to include SUI statements in the job stream. In NOS/BE, the parameter is required only if accounting information is required by the installation. Omitted: COMMENT. - NO USER NUMBER SUPPLIED -
8LPI	CML2	Keyword passed to cause MALET to be assembled to use a print density of 8 lines per inch. If not specified, means MALET assembled for print density of 6 lines per inch.

CML INSTALLATION JOBS DESCRIPTIONS (CML1 THROUGH CML6)

The following are brief descriptions of the functions performed by each of the CML installation jobs (CML1 through CML6):

CML1

- Creates a random program library (PL) on disk from the sequential PL contained on file 2 of the CML release tape.
- Loads MALET scratch file binaries and utility programs as needed to support the configuration defined.
- Loads a file of MALET diagnostic binaries for subsequent use by the jobs submitted by job CML5 (NOS only).

CML2

- Assembles/compiles MALET, HPA, REGEN, and TIO, and retains the binaries on permanent file CMLBIN.

CML3

- Assembles MALET PPU routines MLD and SBP, and retains the binaries on permanent file MLDBIN.

CML4

- Performs an optional system edit (SYSEDIT on NOS; EDITLIB on NOS/BE), incorporating the binaries generated by jobs CML2 and CML3.
- Generates a new deadstart tape (NOS only), incorporating the binaries generated by jobs CML2 and CML3.

CML5

- Routes jobs to input that will load/compile all MALET diagnostics necessary to support the defined configuration.

CML5A (One job is generated for each MALET diagnostic.)

- Loads/compiles one MALET diagnostic and retains the binary as a permanent file.
- Generates any requested hardcopy documentation for MALET diagnostics.

CML6

- Assembles/compiles SKEDULR, MLTDOC, and MTPLOT and retains the binaries on permanent files.
- On 800 series systems (except 865/875), compiles microcode utilities MCCOMP and MCDUMP and retains them as permanent files.

BINARY INSTALLATION

Portions of CML can be installed from the CML Release Tape in binary form. Binary installation is intended to be used in conjunction with the NOS-tailored release process.

Binary installation involves loading from the CML Release Tape those MALET diagnostics, MALET scratch file binaries, and any other maintenance utilities necessary to support a specific hardware configuration.

The major benefits realized through the binary installation process include:

1. A significant reduction is achieved in the time required to attain an operational concurrent maintenance capability.

2. CML capabilities will be avaliable to facilitate checkout of new system installations.

3. The most current version of CML will be included with each new operating system release.

NOTE

For users who have a signed maintenance services agreement with Control Data, those components of CML that reside on the operating system library will be included on the tailored deadstart tape. The source program libraries, installation decks, and any other files that are required to install and support the CML product are contained on the permanent file tape(s) delivered with the operating system.

The format of the CML files contained on the permanent file dump tape(s) is identical to that of the standard CML Release Tape. Therefore, binary installation can be performed from either medium. If the CML Release Tape is used (in lieu of the CML file contained on the system permanent file dump tapes), it MUST be at the same PSR level as the CML binaries contained on the tailored deadstart tape.

The NOS-tailored release process includes creation of a default username intended for use by the customer engineer for maintenance purposes, and to complete installation of the CML product. This default username/password is:

Username	Password
CDCCE	CMLCMLX

If desired, a different username/password may be created for these purposes in lieu of the default.

NOTE

If the default username is used, it is important to change the password prior to placing the system into a production environment. If another username/password is to be used for maintenance purposes, the default username should be eliminated after installation is complete. These precautions will help ensure that unauthorized access to the system does not occur.

Part of the system installation process is to load the CML file from the permanent file dump tape(s) to disk in preparation for the CML binary install process. The file will be retained as:

```
PFN = CML3
UN  = CDCCE
```

Binary installation is accomplished using the installation procedure BININST. This procedure is contained on the first file of the CML Release Tape. BININST is written in the CYBER Control Language (CCL) and is designed to be executed via an interactive terminal. Should an interactive terminal not be avaliable, BININST can be executed as a batch job or from the system console.

Installation time will vary depending on the specific hardware configuration and the installation medium used, but it should not exceed 15 minutes. When installation is complete, permanent file CML3 should be purged.

INTERACTIVE MODE

Binary installation of CML can be performed interactively by using the following procedure:

1. Login to the system through the Interactive Facility (IAF) or through the Remote Diagnostic Facility (RDF). Use either the default username/password or other username created for use by the customer engineer.

NOTE

If a username other than the default is to be used to complete CML installation, ensure that the username has been granted read permission to the CML3 permanent file.

2. Access the CML3 permanent file, or request the CML Release Tape:

```
ATTACH,CMLTAPE=CML3/UN=CDCCE.  (if using CML3 file)

                or

LABEL,CMLTAPE,VSN=CMLVSN,F=SI.  (if using CML Release Tape)
```

   The local file name must be CMLTAPE.

3. Copy the first file, which contains the installation procedures, to a local file named PROCFIL:

```
COPYBF,CMLTAPE,PROCFIL.
```

4. Call the procedure BININST into execution:

```
BEGIN,BININST?
```

   Be sure to append the question mark (?) to the procedure name, as this evokes CCL interactive prompting.

5. You will now be prompted for the equipments contained in the configuration.

   a) To indicate the presence of a device, respond with a Y, followed by a carriage return.

   b) To indicate the absence of a device, respond with a null line (carriage return only), or an N followed by a carriage return.

   c) To obtain a more detailed description of a device, enter a question mark (?) followed by a carriage return. A brief description of the device will be presented, followed by a repeat of the prompt message.

6. After all questions have been answered, the installation process will commence. Brief messages are output to the terminal during execution to keep the user informed of the installation progress.

7. When the installation process is complete, all MALET diagnostic binaries, MALET scratch file binaries, and any other maintenance utilities required to support the defined configuration will be retained as permanent files in the user's catalog.

BATCH MODE

To perform installation in batch mode, create a job of the following format and route it to the input queue using the most convenient means.

```
CMLJOB.
USER,username,password,familyname.
CHARGE,chargenumber,projectnumber.  (if required)
ATTACH,CMLTAPE=CML3/UN=CDCCE.  (if using CML3 file)
      or
LABEL,CMLTAPE,VSN=CMLVSN,F=SI.  (if using CML Release Tape)
COPYBF,CMLTAPE,PROCFIL.
BEGIN,BININST,,eq1,eq2,...,eqi.
6/7/8/9 (EOI)
```

This mode of execution requires that the configuration parameters be passed in the BEGIN statement. Refer to table 1-3, Hardware Configuration Parameters, for a description of valid parameters.

FROM THE SYSTEM CONSOLE

Installation can be performed from the system console by using the batch method described earlier, or by executing the procedure directly. Either method requires use of DIS and the 026 File Editor utility. DIS and 026 are documented in section 1, Console Commands, of the NOS Version 2 Systems Programmer's Instant. The user should be familiar with these utilities before attempting installation from the system console.

To perform the installation in batch mode, use 026 to create a job, as already shown under Batch Mode, then route the job to the input queue.

To execute the procedure directly from the system console, it is best to create a second procedure, as follows, which will then call BININST:

1. Enter DIS. Then enter the desired username/password.

2. Using 026, create the following procedure and write it to file MYPROC.

```
.PROC,MYPROC.
ATTACH,CMLTAPE=CML3/UN=CDCCE.  (if using CML3 file)
          or
LABEL,CMLTAPE,VSN=CMLVSN,F=SI.  (if using CML Release Tape)
COPYBF,CMLTAPE,PROCFIL.
BEGIN,BININST,,eq1,eq2,...,eqi.†
```

3. Under DIS enter:

```
BEGIN,,MYPROC.
```

4. The procedure now runs similar to interactive mode, except that interactive prompting is not avaliable. The hardware configuration must be passed in the BEGIN statement.

   Progress of the installation can be determined by monitoring the job dayfile.

5. When installation is complete, drop DIS.

CML INSTALLATION UTILITIES

In addition to the standard CML installation procedures, a set of CML installation utilites is provided to the NOS user. These utilities are applicable to NOS 2.1, Level 580 and later versions. These utilities are not applicable to NOS/BE or NOS 1.x.

The utilities are not intended to replace the standard CML installation procedures. Full CML installation should be accomplished using the standard installation procedures. The utilities should be used in those situations where changes must be made to individual CML components, and when it is not desirable to perform a complete reinstallation of CML.

The utilities are included in file CMLINST, which is created during CML installation. CMLINST should be retained as a permanent file following installation so that access to the utilities is readily available.

---

†This statement may exceed one line image; if so, the first line must end in a separator. Table 1-3 contains valid hardware configuration parameters for eqi.

The utilities provide the following capabilites.

- Compile MALET diagnostic.
- Generate MALET diagnostic documentation.
- Compile HPA.
- Assemble MALET.
- Assemble MLD and SBP.
- Compile REGEN or TIO.
- Assemble SKEDULR.
- Compile MCCOMP, MCDUMP, MLTDOC, or MTPLOT.
- Load MALET scratch file binaries from CML release tape.
- Perform SYSEDIT of the running operating system, incorporating binaries contained on permanent files MLDBIN and CMLBIN.
- Generate new deadstart tape, incorporating binaries contained on permanent files MLDBIN and CMLBIN.

EXECUTION MODES

The installation utilities are designed for execution primarily from an interactive terminal, although they can be executed from a batch job.

INTERACTIVE EXECUTION

The utilities can be executed interactively through use of the menu-driven procedure, CMLUTIL. When CMLUTIL is executed, the user is presented with the menu shown in figure 1-4.

- After the menu is presented, general information concerning execution of the utilities is obtained by entering a question mark, followed by a carriage return.
- Information concerning any specific utility is obtained by appending a question mark to the number corresponding to the utility, followed by a carriage return.
- The desired function is selected by entering the corresponding number followed by a carriage return. Then, user is prompted for various parameters necessary to perform selected functions.
- Local modifications/corrections code are utilized during updates. Refer to Perform Local Modifications/Corrections Code for file names.

Help is available for all prompts and is obtained by entering a question mark, followed by a carriage return.

```
CML INSTALLATION UTILITIES

1.  COMPILE MALET DIAGNOSTIC
2.  GENERATE MALET DIAGNOSTC DOCUMENTATION
3.  COMPILE HPA
4.  ASSEMBLE MALET
5.  ASSEMBLE MLD AND SBP
6.  COMPILE REGEN OR TIO
7.  ASSEMBLE SKEDULR
8.  COMPILE MCCOMP, MCDUMP, MLTDOC OR MTPLOT
9.  LOAD MALET SCRATCH FILE BINARIES
10. LOAD CML PROGRAM LIBRARY
11. SYSEDIT THE OPERATING SYSTEM
12. GENERATE NEW DEADSTART TAPE

SELECT BY NUMBER OR TYPE Q TO QUIT ?
```

Figure 1-4. CMLUTIL Menu

If a user wishes to bypass the menu presented by CMLUTIL, each utility can be called directly. Refer to table 1-5 for the name of each utility and the corresponding menu selection number.

When executing the utilities from an interactive terminal, the user is asked if the execution is to be accomplished in BATCH or INTERACTIVE mode. If BATCH mode is selected, the user is prompted for user and charge numbers, which must be enclosed with dollar sign delimiters as $user,pw,fm$, and for output listing disposition. A job is then created and routed to the input queue to perform the selected function. This mode of execution frees the terminal and allows several different utilities to be in execution simultaneously. It is best suited for long, time-consuming tasks.

If the interactive mode of execution is selected, the utility is executed in the true interactive mode. Most utilities issue informative messages to the user so that progress can be monitored. In this mode, another utility cannot be started until the current one has completed. True interactive mode is best for single, quickly completed functions.

Table 1-5. Installation Utilities And Functions

CMLUTIL Selection Number	Procedure Called	Procedure Function
1	MALDIAG	Compile one MALET diagnostic and retain binary as a permanent file.
2	MALDOC	Generate documentation for one MALET diagnostic.
3	HPACOMP	Compile HPA. If execution is to be from a local file, retain binary on permanent file HP. If execution is to be from system library, include binary on file CMLBIN.
4	MALASSY	Assemble MALET compiler/executive and include binary on file CMLBIN.
5	MLDSBP	Assemble MALET PPU routines MLD and SBP and retain binaries on file MLDBIN.
6	REGTIO	Compile REGEN or TIO. Retain binary as a permanent file for execution from a local file, and also include binary on file CMLBIN.
7	SKDASSY	Assemble CML utility, SKEDULR, and retain binary on a permanent file.
8	UTLCOMP	Compile CML utility MCCOMP, MCDUMP, MLTDOC, or MTPLOT and retain binary on a permanent file.
9	BINLOAD	Load selected MALET scratch file binaries from CML release tape and retain as permanent files.
10	CMLLOAD	Load CML program library to disk and retain as permanent file CMLPL.
11	SYSEDIT	SYSEDIT running system, incorporating binaries contained on files CMLBIN and/or MLDBIN.
12	SYSTAPE	Generate new deadstart tape, incorporating binaries contained on files CMLBIN and/or MLDBIN.

BATCH JOB EXECUTION

If it is not convenient to execute the utilities via an interactive terminal, they can be called from a batch job. In this case, all required parameters must be passed to the utility by including them on the associated BEGIN statement. Table 1-6 contains a list of the utilities and associated parameters. Any output listings generated are contained on file OUTPUT. Disposition of output is implicit and depends upon the mechanism used to enter the job into the system. The ROUTE statement (see the NOS Reference Set, Volume 3) can be used to override implicit routing. Figure 1-5 shows a sample batch job which uses the utilities.

Local modifications/corrections code are utilized during updates. Refer to Peform Local Modifications/Corrections Code for file names.

All parameters passed in the BEGIN statement must be in the form as shown in figure 1-5.

```
Keyword=option.

    CMLJOB.
    USER,USERNAME,PASSWORD,FAMILYNAME.
    BEGIN,MALDIAG,CMLINST,DIAG=FMC,LISTOPT=S.
    BEGIN,MALDOC,CMLINST,UTILITY=MCCOMP,LISTING=YES.
    6/7/8/9
```

Figure 1-5. Sample Batch Job

SYSOPL (REL1A)

Three of the installation utilities require access to the system OPL (REL1A) for proper execution; MALASSY, MLDSBP, and SKDASSY. These utilities use supplemental procedures (contained on CMLINST) to gain access to this file. When one of these utilities is executed, the user must have either previously gained access to the files, or must provide the utility with the location of the file. The location of the system OPL is passed through the following parameters:

OPLPFN	OPLUNID	OPLPW	OPLPAK
OPLDTSET	OPLVSN	OPLDEN	

Legal values for these parameters are described in table 1-4.

Following is the sequence the utilities use to gain access to the file. The process continues until one of the actions is satisfied.

- When using an interactive terminal:
  - Check for a local file with the name OPL.
  - Check for a local file with the name OPLDIR; it contains directives which, when executed, access the file.
  - Prompt user for location of file. Either attach file or load it from tape, depending on response of user to prompts.

- When running from a batch job (as in figure 1-5):
  - Check to see if file OPL is local.
  - Use permanent file parameters passed in BEGIN statement to access file. (OPLPFN, OPLUNID, OPLPW, OPLPAK, OPLDTSET)
  - If no permanent file parameters were passed, use permanent file SYSOPL if it exists in index of user.
  - If SYSOPL is not in index of user, load it from tape and retain it as SYSOPL under index of user and use values passed in OPLVSN and OPLDEN. If none were passed, use a VSN of REL1A.

The following examples illustrate how a batch job can be structured to gain access to system OPL:

```
JOB...
USER...
ATTACH,OPL/UN=LIBRARY.
BEGIN,MALASSY,CMLINST,SYSVER=NOS23.
BEGIN,MLDSBP,CMLINST,SYSVER=NOS23.
.
.
    or

JOB...
USER...
BEGIN,MALASSY,CMLINST,SYSVER=NOS23,OPLPFN=OPL,
OPLUNID=LIBRARY.
BEGIN,MLDSBP,CMLINST,SYSVER=NOS23.
.
.
```

Once file OPL is local to the job, it remains assigned until job termination.

Table 1-6. Batch Job Parameters (Sheet 1 of 7)

Procedure	Parameter	Function
BINLOAD	CR1	Loads CR1SCR to disk. Legal Values: Yes - Load CR1SCR to disk. No or omitted - Do not load CR1SCR.
	LCN	Loads LCN tests to disk. Legal Values: Yes - Load LCN tests NDM, NDP, and NDT to disk. No or omitted - Do not load LCN tests.

Table 1-6. Batch Job Parameters (Sheet 2 of 7)

Procedure	Parameter	Function
	MY8BCX	Loads MY8BC and BCXBC to disk. Legal Values: Yes - Load MY8BC and BCXBC to disk. No or omitted - Do not load diagnostics.
	ODS	Loads 255X/ODS diagnostics to disk. Legal Values: Yes - Load 255X/ODS diagnostics to disk. No or omitted - Do not load diagnostics.
CMLLOAD	LISTING	Causes deck list to be generated for subsequent printing. Legal Values: Yes - Generate listing. No or omitted - Do not generate listing.
HPACOMP	HPSORT	Causes internal sort routine to be used in lieu of SORT/MERGE 5. This parameter can be used when SORT/MERGE 5 not available. Legal Values: I - Use internal sort routine. No or omitted - Use SORT/MERGE 5.
	LISTING	Causes listing of the HPA program to be generated for subsequent printing. This listing is quite large and may take several hours to print on a remote batch printer. Legal Values: Yes - Generate listing. No or omitted - Do not generate listing.
	XRUN	Causes HPA to be compiled for execution from local file. Legal Values: Yes - HPA is compiled and retained on permanent file HP. No or omitted - HPA is compiled for execution from system library. Binary is included on permanent file CMLBIN.

Table 1-6. Batch Job Parameters (Sheet 3 of 7)

Procedure	Parameter	Function
	66X	Causes Detailed Status Tables for 66X Tape Subsystem to be included in HPA. Legal Values: Yes - Include tables in HPA. No or omitted - 66X Detailed Status Tables are not included in HPA.
	67X	Causes Detailed Status Tables for 67X Tape Subsystem to be included in HPA. Legal Values: Yes - Include tables in HPA. No or omitted - 67X Detailed Status Tables not included in HPA.
	844	Causes Detailed Status Tables for 844 Disk Subsystem to be included in HPA. Legal Values: Yes - Include tables in HPA. No or omitted - 844 Detailed Status Tables not included in HPA.
	885	Causes Detailed Status Tables for 885 Disk Subsystem to be included in HPA. These tables are also used to analyze DEMA detailed status. Legal Values: Yes - Include tables in HPA. No or omitted - 885 Detailed Status Tables not included in HPA.
MALASSY	LISTING	Causes listing of the MALET compiler/executive to be generated for subsequent printing. Legal Values: Yes - Generate listing. No or omitted - Do not generate listing.
	PAPER85	Causes MALET to be assembled for use with 212.5 mm (8.5 in) paper. Legal Values: Yes - Assemble MALET for use with 212.5 mm (8.5 in) paper. No or omitted - Assemble MALET for use with standard 279.4 mm (11 in) paper.

Table 1-6. Batch Job Parameters (Sheet 4 of 7)

Procedure	Parameter	Function
	SYSVER	Specifies level of NOS for which MALET to be assembled. Legal Values: NOS25 - NOS 2.5 NOS24 - NOS 2.4 NOS23 - NOS 2.3 NOS6 - NOS 2.2 or NOS 2.1
	8LPI	Causes MALET to use print density of 8 lines per inch. Legal Values: Yes - MALET output printed at density of 8 lines per inch. No or omitted - MALET output printed at density of 6 lines per inch.
	OPLPFN OPLUNID OPLPW OPLPAK OPLDTSET OPLVSN OPLDEN	See notes on SYSOPL elsewhere in this section.
MALDIAG	DIAG	Specifies which MALET diagnostic to be compiled. Legal Values: Any three-character MALET diagnostic contained on CML release tape. This is required parameter.
	LISTOPT	MALET list options. Legal Values: S - Causes source code to be carried with module binaries. L - Causes source and binary listing to be written to list file. I - Causes all input source lines to be written to list file. F - Turns on all options (S, L, and I). (The list options can be entered in continuous string [e.g., SL, SI, etc.].)

Table 1-6. Batch Job Parameters (Sheet 5 of 7)

Procedure	Parameter	Function
	XMAP	Causes global cross-reference map of all modules compiled to be included in output listing. L list option must also be selected if global cross-reference map is desired. Legal Values: Yes - Include global cross-reference map on list file. No or omitted - Do not produce global cross-reference.
MALDOC	DIAG	Specifies the MALET diagnostic for which documentation to be generated. Legal Values: Any three-character MALET diagnostic contained on CML release tape. This is required parameter.
	DOCTYPE	Specifies type of documentation desired. Legal Values: EXT - Specifies external documentation to be produced. INT - Specifies internal documentation to be produced. BOTH - Specifies both external and internal documentation to be produced.
MLDSBP	LISTING	Causes listing of MLD and SBP to be produced for subsequent printing. Legal Values: Yes - Generate listing. No or omitted - Do not generate listing.
	SYSVER	Specifies level of operating system for which MLD and SBP to be assembled. Legal Values: NOS25 - NOS 2.5 NOS24 - NOS 2.4 NOS23 - NOS 2.3 NOS6 - NOS 2.2 or NOS 2.1

Table 1-6. Batch Job Parameters (Sheet 6 of 7)

Procedure	Parameter	Function
	OPLPFN OPLUNID OPLPW OPLPAK OPLDTSET OPLVSN OPLDEN	See notes on SYSOPL elsewhere in this section.
REGTIO	DIAG	Specifies which diagnostic (REGEN or TIO) to be compiled. This is required parameter. Legal Values: REGEN - Compile REGEN. TIO - Compile TIO.
	LISTING	Causes listing of selected diagnostic (REGEN or TIO) to be generated for subsequent printing. Legal Values: Yes - Generate listing. No or omitted - Do not generate listing.
SKDASSY	LISTING	Causes listing of SKEDULR to be generated for subsequent printing. Legal Values: Yes - Generate listing. No or omitted - Do not generate listing.
	MMF	Specifies that SKEDULR is to be compiled for use in an ECS-based, multimainframe environment. Legal Values: Yes - Assemble SKEDULR for use in multimainframe environment. No or omitted - Assemble SKEDULR for use in a single-mainframe environment.
	SYSVER	Specifies level of operating system for which SKEDULR is to be assembled. Legal Values: NOS25 (NOS 2.5). NOS4 (NOS 1.3). NOS24 (NOS 2.4). NOSBE6 (NOS/BE 1.5). NOS23 (NOS 2.3). NOSBE5 (NOS/BE 1.4). NOS6 (NOS 2.0, 2.1, 2.2). NOSBE4 (NOS/BE 1.3). NOS5 (NOS 1.4).

Table 1-6. Batch Job Parameters (Sheet 7 of 7)

Procedure	Parameter	Function
	OPLPFN OPLUNID OPLPW OPLPAK OPLDTSET OPLVSN OPLDEN	See notes on SYSOPL elsewhere in this section.
SYSEDIT	LISTING	Causes listing of SYSEDIT directives and actions to be generated for subsequent printing. Legal Values: Yes - Generate listing. No or omitted - Do not generate listing.
SYSTAPE	LISTOPT	Specifies desired LIBEDIT list options. Legal Values: C - List directives. E - List errors. M - List modifications. N - List records written to new file. F - Give full listing (enable all options). Omitted - Do not produce listing. (You may combine list options; for example, CM, CE, or CEM.)
UTLCOMP	LISTING	Causes listing of requested utility to be generated for subsequent printing. Legal Values: Yes - Generate listing. No or omitted - Do not generate listing.
	UTILITY	Specifies utility to be compiled. Legal Values: MCCOMP MCDUMP MLTDOC MTPLOT

MALET LANGUAGE 2

MALET is an acronym for the Maintenance Application Language for Equipment Testing. It was designed to provide a set of commands capable of driving 6000/CYBER computer system peripherals after being granted access by the operating system. MALET executes under the NOS and NOS/BE operating systems, but CYBER 170 Model 176 support is limited to I/O devices attached to the Peripheral Processor Subsystem (PPS) chassis.

The capability for writing maintenance tools in the MALET language is provided by a Central Processor Unit (CPU) executive which is installed in the operating system. Maintenance routines may then be written and stored as permanent files. These files do not become part of the operating system and may be released bypassing the normal software release mechanism as they represent application-type programs.

The MALET compiler and executive are written in the 6000 COMPASS Assembly language and form a common product which interfaces with the operating system. The operating system interface routine, CVL, is required for MALET to run. Diagnostics are written in the MALET language, compiled into executable Peripheral Processor (PP) code by the MALET compiler, loaded and linked by the MALET compiler, and executed by the PP driver and its product overlays.

MALET ELEMENTS

MALET is composed of the following nine elements.

- An executive which controls the PP driver and provides a user interface.
- A compiler which compiles high-level, low-level, and non-I/O language commands into executable PP code subroutines called modules.
- An executive/PP driver interface residing in Central Memory (CM).
- A PP driver which executes non-I/O statements.
- Product overlays for the PP driver which execute low-level and high-level I/O commands.
- Low-level commands such as CONNECT, STATUS INPUT, OUTPUT, and FUNCTION designed for channel interfaces and controllers and which are not device oriented.
- High-level language commands such as REWIND or ENDFILE for tape units and PRINT or EJECT for printers, designed for specific hardware families.
- Non-I/O commands which control module program flow and data in the PP operating registers and buffers. Non-I/O commands may apply to both the high-level and low-level languages.
- Seven MALET local files: INPUT, OUTPUT, LIST, RANDOM(RFILE), PUNCH, BINARY, and SCRATCH.

A more detailed description of these nine elements follows.

EXECUTIVE

The executive is brought to a control point in the operating system by the MALET call. It then resides in a designated CPU area with the MALET compiler.

Input to the executive is in one of two forms: directives which are executed by the executive or source code which is saved for the compiler. The executive performs the following tasks:

- Acts upon directives.
- Saves source code for the compiler.
- Inputs and outputs compiled modules to or from the random file.
- Drives the system console display.
- Controls the PP driver and product overlays.
- Accepts PP requests for assistance during the execution of MALET modules.

Data input to the executive which is not recognized as a directive is assumed to be source code and is saved in a source code buffer. Any data which does not begin with either an alphabetic character, or a plus or minus sign is considered to be source data. The MALET executive provides the following capabilities for the input of directives and source code.

- Input may be from the system K/L display, the input file, or a local file. A module taken from any of these sources may be read, compiled, and executed by the executive.
- Input may be from a local file with modifications from the system K/L display or input file.
- Precompiled modules may be loaded from a local file and executed.
- A remote batch job may be submitted which contains executive directives and source code to be compiled and/or executed with the output being returned to the remote station.
- Remote batch jobs may be submitted which obtain source code or precompiled modules from a local file for execution.

Typically, a number of modules are required to form a total diagnostic. The executive allows the user to run specific modules in a chosen sequence with the option to loop on a single module or a series of modules.

The executive reads a complete module into the module buffer and source code, if present, is moved to the source code buffer. The source code may be displayed, modified, and/or recompiled. If the module is executed, PP code is read from the binary code block (figure 2-1) to the PP instruction stack by the PP driver. If there are any DATA statements in the module, this data is read to the PP output buffer for access by the user.

The output-of-results mechanism within the MALET language allows the user to issue messages to the system B display, the dayfile, and/or the output file. Additional capabilities allow the user to accumulate error messages in the PP display buffer and control the point in time at which a picture of the display is sent to the active output file.

COMPILER

The compiler compiles source code in the form of MALET language commands into PP executable binary code. It can also produce a listing of the source code and binary PP code on the output file.

The interface between the executive and the compiler consists of a source code buffer which contains a maximum of 120 decimal source code lines and a module buffer of 1100 CM words in which the compiler stores the binary code it produces from source code.

The module buffer also holds MALET modules retrieved from the random file for execution by the PP driver. The format for the module buffer is shown in figure 2-1.

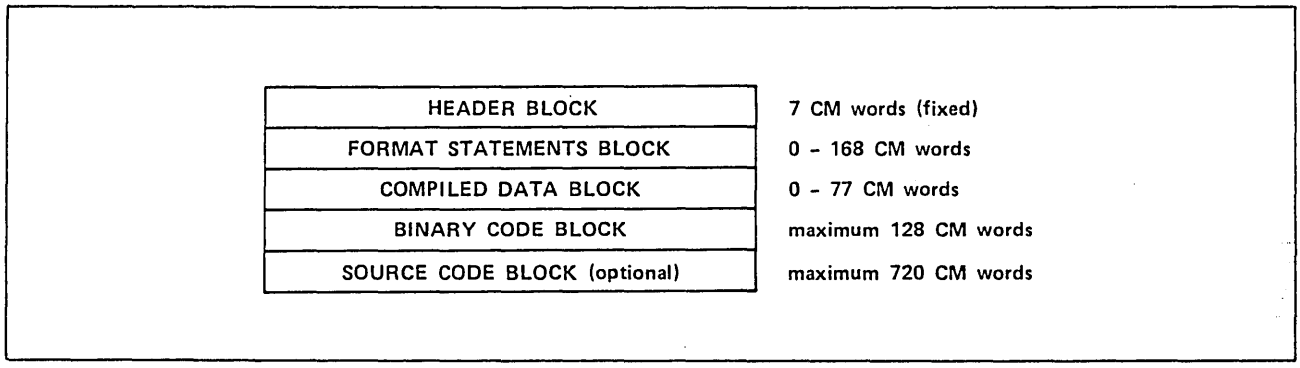

Figure 2-1. Module Buffer Memory Format

The segments of the module buffer serve the following purposes:

- The seven word header block contains data pertinent to loading and executing the module.
- The format statements block contains messages defined for use during module execution.
- The compiled data block contains data which is sent to the PP output buffer (prior to module execution) for use by the module during execution.
- The binary code block contains the executable code generated by the MALET compiler. This code is transferred to the PP instruction stack when the module is executed.
- The source code block contains the MALET language statements which were used by the compiler to form the binary code block. This block may be viewed on the K/L display when the module is executed. When the source code is present, the user has the option of changing the code, recompiling, and reexecuting the module. The source code is only saved with the module when the source option is specified on the COMPILE directive.

EXECUTIVE/PP DRIVER INTERFACE

The executive/PP driver interface is a block of consecutive CM locations subdivided to perform the monitoring activities described in table 2-1.

Table 2-1. Executive/PP Driver Interface Memory Distribution

Description	CM Word Length
Security Word	1
Input register	1
Output register	1
Message buffer	5
PP display buffer	123
PP plot buffer	40
PP checkpoint buffer	599
Scratch file working buffer	64
Module buffer	1100
Product overlay error message buffer	120
Standby PP communications	2

Executive commands to the PP driver are processed through the input register and message buffer. PP requests to the executive are processed through the output register and message buffer.

PP DRIVER

The PP driver portion of MALET which resides in the PPS with the product overlays is a Maintenance Language Driver (MLD). It executes the instructions in the binary code block of the module buffer. The PP driver is called into operation by the executive RUN directive. It remains in operation until one of the following conditions occur:

- A STOP directive is processed by the executive and passed to the driver. The PP stops when a logical breakpoint (such as a RES or REL sequence) is reached.
- An END statement is executed by the PP driver. The PP drops the control point and flags the executive.

- A run time limit is detected by the executive. This limit may be set using the TL option on the executive PARAM directive. The time limit condition is only encountered when operating in the terminal mode.

- The PP driver detects that more than 2 seconds have elapsed since the last release of the assigned I/O channel to the operating system.

MALET modules execute from the instruction stack in the PP driver. The executive loads PP code into the instruction stack from the binary code block of the module buffer and transfers control to the PP driver. The PP driver executes return jumps from the instruction stack to PP driver subroutines for assistance in executing some commands.

Communications between the PP driver and the product overlays are accomplished by return jumps from the instruction stack to command subroutines in the product overlays. Parameters are passed to the subroutines through the PP direct cells. Results generated by the product overlays are stored into areas defined by the parameters, usually the product overlay input data buffer, output data buffer, or status buffer.

PRODUCT OVERLAYS TO PP DRIVER INTERFACE

The PP product overlay is a specialized overlay to the basic PP driver whose purpose is to drive specific I/O devices. All I/O related commands executed out of the instruction stack make return jump calls to the product overlay to accomplish the task.

Command subroutines within the product overlay consist of fixed and relocatable areas. Jumps from the instruction stack proceed to the fixed areas where other jump instructions direct execution to the appropriate subroutine. This method of first jumping to a fixed area offers two advantages:

- Entry points to the command subroutines may be defined in the compiler allowing it to generate its own return jumps.

- Product overlays may be significantly modified without the necessity of changing the entry points defined in the compiler.

Once the task is complete, the product overlay returns control to the instruction stack where the module continues execution.

I/O channel failures, I/O sequence errors and range errors on data buffers are processed in the product overlay as follows:

- The error code is set in the Error Code (EC) register.

- The current line number is set in the Error Address (EA) register.

- The error message index is set in the Error Message (EM) register.

The overlay completes error processing by jumping to the PP driver error exit sequence.

LOW-LEVEL I/O LANGUAGE COMMANDS

The low-level I/O language commands apply to all 3000 and 6000 computer system equipments as well as buffer controller devices on 6000/CYBER channels.

This level of language is primarily used to test I/O channel interfaces and controllers. PP code generated by the compiler passes parameters to direct cells in the PP and return jumps to a subroutine in the PP overlay to accomplish the command task. The compiler generates PP code for performing direct I/O channel operations only when inputting the real-time clock for use as an operand. Normal I/O is always through the product overlays.

HIGH-LEVEL I/O LANGUAGE COMMANDS

The high-level I/O language commands in MALET are designed and fitted for specific hardware families. The compiler generates PP code which passes parameters to the PP direct cells and return jumps to the PP product overlay to accomplish the task. This level of language is primarily used to test peripheral units.

NON-I/O LANGUAGE COMMANDS

The non-I/O language commands supplement both levels of I/O commands. The primary purpose of the non-I/O commands is to control program flow within a module and to supervise data within the PP operating registers and buffers.

The MALET compiler creates executable PP code from the non-I/O commands in one of the following formats:

- Code which executes with no assistance.
- Code which requires assistance from basic PP driver subroutines.
- Code which executes entirely through PPU driver subroutines.

The compiled non-I/O commands never perform direct Central Memory (CM) reads or writes. If a command requires communication with CM or the CPU executive, the communication is performed by PP driver subroutines.

MALET LOCAL FILES

The seven MALET local files are the INPUT, OUTPUT, LIST, RFILE, PUNCH, BINARY, and SCRATCH files. Briefly stated, their purposes are:

- INPUT contains source code and executive directives.
- LIST and OUTPUT contain module listings and pictured data.
- RFILE contains compiled modules for execution.
- PUNCH contains source buffer data placed there by an executive PUNCH directive.
- BINARY is a sequential file containing compiled modules which are available to RFILE.
- SCRATCH is a local scratch file used for read/write data, system commands, and controlware.

STRUCTURE AND ORGANIZATION

In addition to the nine elements of the MALET language, there are details of structure and organization which are essential for a more complete understanding of MALET. Following are descriptions of MALET modules, registers and buffers, memory requirements, and data organization.

MODULE

The smallest executable MALET diagnostic is a module. It consists of a maximum of 120 (decimal) lines of source code separated into four divisions.

- The identification division names the module and describes the device type and specific device codes supported.
- The data division is optional and may contain the following:
  - Data statements for use within the module.
  - Format statements which define messages for use within the module. A maximum of eight microsubstitutions may be made within each format statement with a maximum of 24 format statements allowed per module.
  - Equate statements which equate a user specified name to an octal or decimal value. A maximum of 20 equates are allowed within the module. The equated value may be used for any variable which allows an octal or decimal value.
  - Base statements which define the base (octal or decimal) of numeric values used within the module. The default base is octal, but any value may be followed by a B (octal) or D (decimal) to provide a specific base for that number.
  - Comment statements which begin with either a slash (/) or an asterisk (*) document the purpose and use of the module. These statements may appear anywhere between the identification and termination divisions of the module.
- The executable code division contains the commands which are converted to PP code for the PP driver. High- and low-level language commands may not be used together in a module.
- The termination division ends the module and specifies the module entry point.

Identification of source code lines is performed by MALET which assigns each line a number between 0 and 167 (octal). Additionally, user assigned line numbers from 0 to 77 (octal) or 0 to 63 (decimal) are permitted. Because there are 120 (decimal) source lines available and only 63 (decimal) user defined lines, user defined lines should be limited to statements which require them.

REGISTERS AND BUFFERS

A number of registers and buffers are contained within MALET which are used as variables by the MALET language. Registers and buffers identified as being user-programmable may have their values altered by the user during module execution. Registers which are identified as read-only may not be altered once module execution begins but may be referenced as operands from the module. Read-only registers are loaded using executive directives.

The following user-programmable registers exist in the PP driver.

Register Mnemonic	Description
B0 through B15	Index registers.
WC	Word count register.
BA	Beginning address register of buffer first word address (fwa).

The following read-only registers exist in the PP driver.

Register Mnemonic	Description
P0 through P9	Parameter registers. These registers are set using the executive PARAM directive.
WT	Words transmitted register. Contains number of words transmitted during the last I/O operation.
LF	Last function register. Contains the last function code issued to I/O.
EC	Error code register. Contains the last posted error code.
EA	Error address register. Contains the module source code line where the last posted error occurred.
EM	Error message index register. Contains the index which refers to an error message in the module format block.
ES	Executive switch register. Contains bit switches referenced by modules executing in a PPU and driving a device directly connected to the I/O channel.
DS	Diagnostic switch register. Contains bit switches referenced by modules executing remotely in devices such as the Loosely Coupled Network (LCN) and the Federal Standard Channel (FSC).
DC	Device code register. Contains the device code of the assigned device.
AL	Access level register. Contains the access level of the assigned device.
P	Program address register. Contains the current module source line number being executed.
RT	Real-time clock register. Contains the value of the real-time clock. This register may be referenced as an operand only in a few specific cases.

The following buffers exist in each PP driver product overlay.

Register Mnemonic	Description
OB	Output buffer. Contains 500 (octal) words of data from the PP to be written to a device.
IB	Input buffer. Contains 500 (octal) words of data from a device to be read by the PP.
SB	Status buffer. A read-only buffer containing 100 (octal) words of status input from the assigned device.

Buffers may be directly referenced with octal or decimal subscripts. Buffers may be indirectly referenced using equated values or registers and optional octal or decimal values.

Examples:

IB(20)	Input buffer directly referenced by an octal subscript.
OB(8)	Output buffer directly referenced by a decimal subscript.
SB(GENST1)	Status buffer indirectly referenced by an equated value.
OB(B1)	Output buffer indirectly referenced by an index register.
SB(B1+5)	Status buffer indirectly referenced by an index register summed with an octal value.

The default base for numeric fields is octal. The occurrence of an 8 or 9 in a numeric field indicates the field is decimal and causes it to be converted to an octal value internally. If a postradix of B or D is present, the number is interpreted according to the postradix.

Examples:

LINE=78	78 is interpreted as 116_8.
LINE=72D	72 is interpreted as 110_8.
ASSIGN,AL=8,CH=1	AL is interpreted as 10_8.
LINE=71B	71B is interpreted as 71_8.

EXECUTIVE AND DIAGNOSTIC SWITCH REGISTERS

The MALET compiler supports the following keywords as equated values allowing easy reference to specific bits of the ES and DS registers. These values may be used anywhere a constant or equated value is legal.

ES Register		DS Register	
RM	Repeat module.	RM	Repeat module.
RC	Repeat condition.	RC	Repeat condition.
MS	Module stop (repeat module or stop on error).	MS	Module stop (repeat module or stop on error).
CS	Condition stop (repeat condition or stop on error).	CS	Condition stop (repeat condition or stop on error).
DL	Dayfile log.	BM	Beginning of module stop.
PD	Print log or dayfile log.	CE	Condition end.
SE	Stop on error.	S1	Programmable switch 1.
SL	Scoping loop (repeat module or condition, do not stop on error).	SE	Stop on error.
		CM	Repeat condition or module.
CM	Repeat condition or module.	BC	Beginning of condition stop.
TM	Terminal mode.	BT	Beginning of test stop.
PL	Print log.	TE	Test end stop.
KL	Assign K/L display.	ME	Module end stop.
		S2	Programmable switch 2.

MEMORY REQUIREMENTS

The MALET language requires memory space in both the CPU and PPs. The memory requirement in the CPU is 21K (octal) as shown in figure 2-2. This memory contains the MALET executive and compiler. The memory requirement for a PP is 10K (octal) as shown in figure 2-3. This memory is subdivided into specific areas such as the PP driver; instruction stack; input, output, and status buffers.

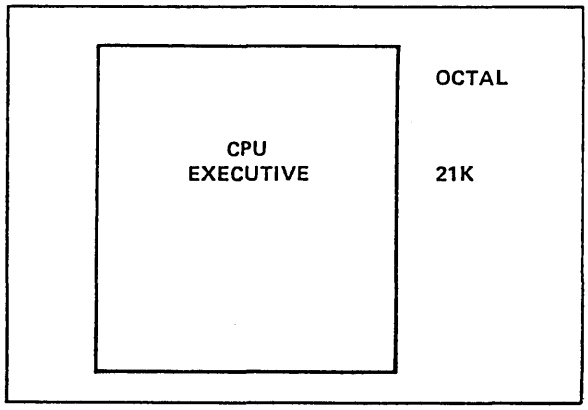

Figure 2-2. MALET CPU Memory Requirements

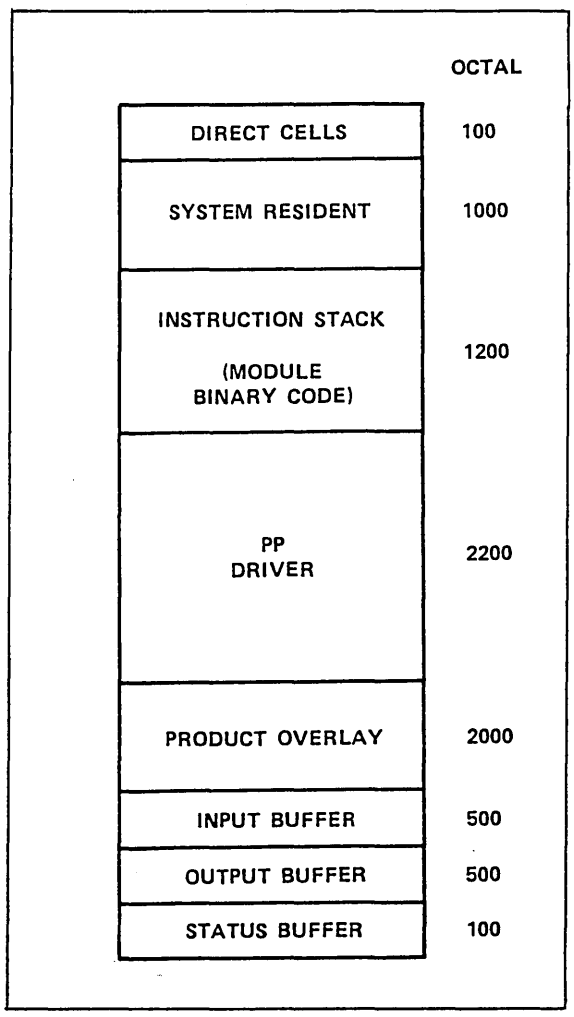

Figure 2-3. MALET PP Memory Requirements

DATA ORGANIZATION

Executive directives and source code lines are assembled for input using a 60 character maximum line format. Directives are decoded and acted upon immediately. Source code data is stored in the source code buffer using a fixed format of 60 characters (six CM words) per line. A maximum of 120 (decimal) source code lines (one module) are permitted as the source code buffer has a fixed length of 720 decimal CM words.

MALET PROGRAM RELATIONSHIPS

The MALET excutive calls CVL during the ASSIGN directive to receive validation of system devices for maintenance. MALET also calls CVL to return the device during the exit sequence.

The MALET executive requests the operating system to load MLD along with the correct product overlay into a pool PP for I/O.

If SETSW, SB directive is used, the MALET executive requests the operating system to load PP program SBP into a pool PP. SBP monitors the assigned channel for hang conditions and automatically disconnects the ASSIGN channel. Refer to figure 2-4.

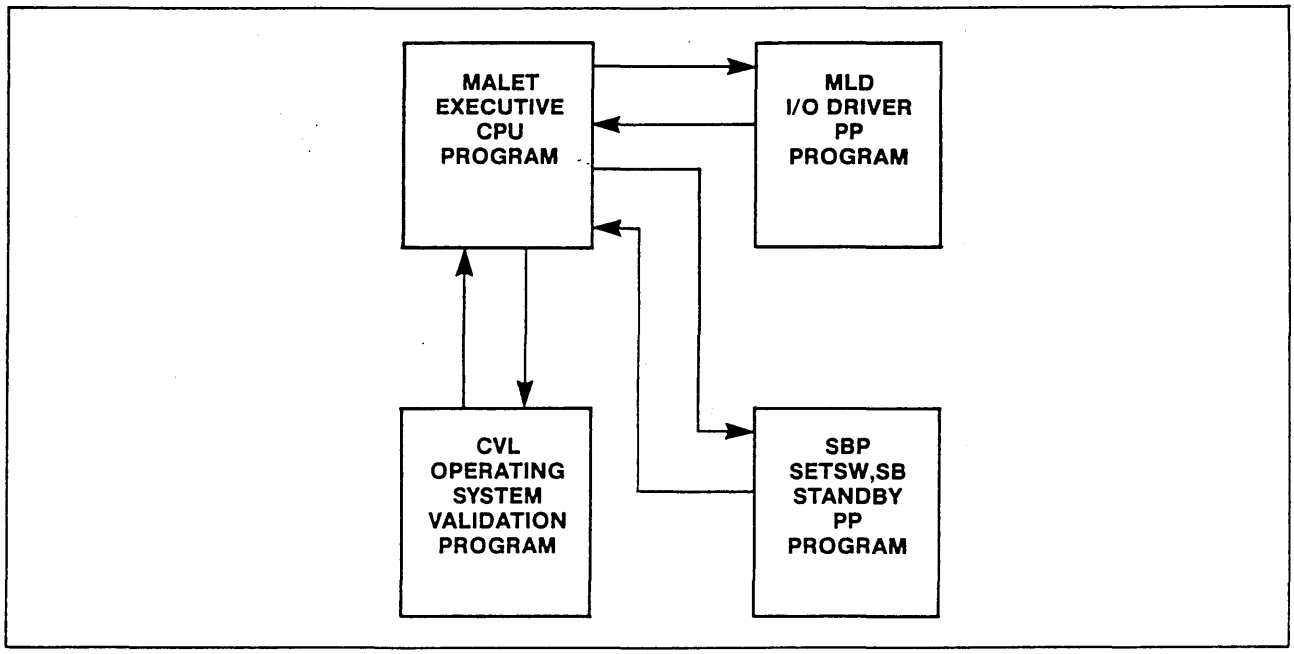

Figure 2-4. MALET Program Relationships

MALET LANGUAGE FORMAT

Certain formatting rules exist in the MALET language to simplify interpretation of the three basic command groups (non-I/O, high- and low-level I/O) by the compiler. These rules are stated as follows.

1. STATEMENT NUMBERING. User-defined statement numbers 0 through 77 (octal) or 0 through 63 (decimal) are permitted for labeling MALET statements. Statement numbers must appear in positions 1 and/or 2. Single-digit statement numbers may appear in either column 1 or 2 with leading zero optional.

2. NUMERIC BASE FOR STATEMENT NUMBERING. If decimal values 0 through 63 are used for the user-defined statement numbers, the command BASE DECIMAL must appear in the module. Otherwise, octal values are assumed.

3. POSITION 3 BLANK. Position 3 must be blank on all noncomment statements, with the exception of DATA continuation where a comma (,) in column 3 indicates the continuation. Continuation is allowed only on data statements. Comment statements have no restrictions concerning position 3.

4. COMMAND KEYNAMES AND SUCCESSIVE BLANKS. Command keynames begin in either position 4 or 5. The compiler interprets two successive blanks as the end of the command. Each command begins with a keyname followed by a sequence of parameters.

5. SEPARATORS. Commas (,), periods (.) or single blanks may be used to separate keynames and parameters. Two successive commas or periods cause a zero to be stored as a parameter. Each successive comma or period after the first two stores an additional zero parameter. Single blanks, single commas, and single periods may be used anywhere in MALET syntax for clarity or readability.

   Examples:

`DATA (OB(10),CON) 1,2,,3`	Data items = 1, 2, 0, and 3.
`IB(10) = B1 + 1234`	Two succesive blanks following the digit 4 terminate the command. Single blanks are ignored.
`IB(10) = B1.AND.0707`	Single periods are ignored.
`REWIND,ABT 10`	Single commas are ignored.

6. OPERATORS. Operation symbols such as plus (+), minus (-), left and right parentheses ((and)), equal (=), dollar ($), asterisk (*), and slash (/) must be used if they appear in the format of the specific command. These symbols are not optional.

7. COMMAND TERMINATION AND DOCUMENTATION. Once a command has been properly terminated with two successive blanks, any character positions remaining in the 60 character line length may be used for documentation purposes.

8. COMMENT STATEMENTS. An asterisk placed in position 1 indicates that the line is a comment. Comments may be printed by the executive but are not used by the compiler.

OPERATIONAL PROCEDURE

After the physical installation of MALET, users must be defined along with file names and execution modes before MALET becomes available. Documentation of these procedures is included with the CML/MALET software release material.

When MALET is available, it may be called and executed using the sequence of steps outlined in table 2-2.

Table 2-2. MALET Test Execution Sequence

Step	Directive	Action
1	GET or ATTACH	Assign a scratch file (if required).
2	MALET	Call MALET (console, terminal or batch).
3	ASSIGN	Assign a device for testing.
4	PARAM	Set parameter registers.
5	SETSW	Set switches in the ES register.
6	RUN	Execute the test module(s).
7	GO	Resume execution upon failure.
8	DROP	Terminate use of MALET.

The MALET executive is called and executed by the MALET control statement formatted as follows:

```
MALET,p1=m,p2=n,...,pn=q

    p1,p2,...,pn        Specific keynames
    m,n,...,q           User supplied values
```

Specific keynames may be used to define local files, override executive default conditions, set terminal/batch and keyboard conditions, and specify control list options.

The following keynames allow the user to supply a local file name in place of a defaulted file name. All files must be disk resident for use by MALET.

I or INPUT

File containing input source code and MALET executive directives. If the file is named something other than INPUT, it is an alternate input file read until EOF/EOI is encountered. At that point, the file named INPUT is read (Default: I=INPUT).

O or OUTPUT
L or LIST

File which is to receive output data from MALET. If MALET is run using a terminal, the file name determines where a copy of OUTPUT is destined (Default: O=OUTPUT or L=OUTPUT).

If you do not want the terminal to receive a copy of output when compiling a large number of modules, use the L or O option to select an alternative output file and ensure that the input file name is COMPILE. With these conditions, the alternative output file receives a copy of data not sent to the terminal.

B or BINARY

File where MALET reads or writes precompiled modules for possible execution (Default: B=BINARY).

P or PUNCH

File where source data is written in response to a PUNCH directive.

R or RANDOM

File used for compiled modules (Default: R=RFILE).

S or SCRATCH

File used as a local scratch file for reading and writing during SYS commands (Default: S=SCRATCH).

The following keynames allow the user to override default conditions in the executive.

T=nn or TERM=nn

Terminal/Batch mode; informs MALET that execution is in terminal mode. The nn denotes the screen line length in decimal. The parameter is needed only if split compiled module listing over lines. Terminal mode is obtained from the operating system (Default=Omitted).

KL

Console Keyboard On/Off; informs MALET not to drop when the input file is exhausted. The K/L may be turned on via the SETSW KL bit in the ES register, but the ES register does not turn on the display (Default: KL=Off).

LO=x

Control list options (Default=Off). The following options are available. Refer to samples in appendix D.

x	Description
S	Source; turns on the source options in the compiler. This option causes the source code to be carried with the compiled module binary code when it is saved in the random file. The source code is carried regardless of errors encountered during module compilation. Note that this option significantly increases mass storage requirements and module load time.
L	List; turns on the list options in the compiler. This option causes the source and binary listings of the compiled module to be written to the output file.
I	Input; lists all input source lines, including update sequence numbers.
F	Full; turns on all options.

MAP

Causes assembly of MALET modules with listing to form a global cross-reference map at end of file on input. Refer to sample in appendix D.

CMSE

Assembles MALET modules for off-line execution. Any modules assembled this way cannot be executed on-line (Default=On-line). Following is a CMSE sample job:

```
Job/Account Cards
REQUEST,CML,VSN=CML,NT,PE,F=SI,LB=KU.
UPDATE,Q,P=CML.
MALET,I=COMPILE,CMSE,B=MSLBIN.
REQUEST,TAPE,F=SI,LB=KU,NT,PE,VSN=TAPE.
REWIND,MSLBIN,TAPE.
COPYBF,MSLBIN,TAPE.
7/8/9
*ID,CMSE.
*DF,CMSE17x.
*COMPILE,DIAG.†                    (where DIAG is test to be copied)
7/8/9
DUMP
6/7/8/9
```

A binary file is created on tape which may be added to the MSL disk library with the MSL TDX utility and executed under DEMOT.

BATCH MODE EXAMPLE

All default options are used.

- INPUT is from card reader Input file.
- OUTPUT is to the standard Output file.
- BINARY modules are on Binary or do not exist.
- SCRATCH is on Scratch or does not exist.
- PUNCH is on Punch or does not exist.
- RANDOM is on Rfile or does not exist.

The run is in batch mode, no input is to be read from the system console, and no list options are forced.

SYSTEM CONSOLE MODE EXAMPLE

NOS	NOS/BE
ATTACH,T6X.	ATTACH, T6X, ID=T6X.
MALET (R=T6X,KL)	MALET (R=T6X, KL)

T6X contains compiled modules.

The K/L parameter allows input from the system console keyboard.

†Refer to appendix B in this manual for use with DEMA diagnostics.

TERMINAL MODE EXAMPLE

NOS	NOS/BE
ATTACH, T7X.	CONNECT,INPUT,OUTPUT
MALET (R=T7X, T=80	ATTACH,T7X, ID=T7X.
	MALET (R=T7X,T=80)

The T=80 parameter selects terminal mode with a screen width of 80 columns. If the terminal has the input and output files connected, the terminal becomes the MALET input and output device.

MALET EXECUTIVE DIRECTIVES

The following directives control the MALET executive. All the directives are valid when entering the full mnemonic and some are valid when only the first character is entered. Commas and single blanks may be used as separators in the directives.

ASSIGN or A Directive

Assign a device to MALET for testing.

```
ASSIGN EST=aaa,CH=bb,EQ=cc,UN=dd,DC=ee,SN=ffffff,AL=gg,SHARE,ESM=NO,MSA=h,
       MSAID=ii,C=j,CSU=k,CSUID=ll,MST=m,MSTID=nn,MSTPH=p,MNE=q,TCU=rr,
       LTA=sss,AC=ttttH.

     A EST=aaa,CH=bb,...,LTA=sss,AC=ttttH.
```

aaa	Equipment Status Table (EST) ordinal; default=0. If EST=NO, the device is not listed in the EST.
bb	Channel number; default=0.
cc	Equipment number; default=0.
dd	Unit number; default=0.
ee	Device code; default=0. Before module execution, MALET checks the module header to verify that the module contains instructions for the assigned device code. Each module may support 10 unique devices.
ffffff	Media serial number; default=0.
gg	Access level; default=0 for non-RMS, =1 for RMS (refer to appendix C).

SHARE — Share channel with another mainframe; no default value. The SHARE parameter causes the RES (reserve) command to release the I/O channel even if the operating system does not wish to use it. A delay of 1 millisecond for non-RMS devices (100 milliseconds for RMS devices) occurs before the I/O channel is requested for further testing. The SHARE parameter is not necessary if there is only one access to the equipment under test.

ESM=NO — Disables assignment of ESM space for scratch file when device code (DC=ee) is 14, indicating a DEMA device.

h — Mass Storage Adapter (MSA) number; default=0.

ii — MSA identification number; default=0.

j — Mounting cartridge; default=NONE.

NONE = No cartridge mounted
7777B = Mounted manually
xx/yy = Cartridge mount from location xx_8 and yy_8.

k — Cartridge Storage Unit (CSU) number; default=0.

ll — CSU identification number; default=0.

m — Mass Storage Transport (MST) number; default=0.

nn — MST identification number; default=0.

p — MST physical address; default=0.

q — Two character mnemonic identifying assigned tape device; no default value.

MT = 7-track tape
NT = 9-track tape

rr — Trunk Control Unit (TCU) enables for LCN NAD; no default value. Multiple enables may be used if DC=301B.

bits 2^0 TCU 3
bits 2^1 TCU 2
bits 2^2 TCU 1
bits 2^3 TCU 0

sss — Logical Trunk Address (LTA) of the assigned NAD; no default value. This entry is required if DC=301B.

ttttH — Access Code (AC) of the assigned NAD; no default value. This hexadecimal entry is required if the NAD AC hardware switches specify a nonzero value and DC=301B.

Description:

The ASSIGN directive contains the parameters which MALET uses to request a device for testing. The parameters are passed to the operating system interface routine, CVL, where they are validated and access to the device is granted.

Method of Execution:

The ASSIGN directive passes all parameters to CVL for validation. When validation is complete the channel, equipment and unit numbers, as well as the EST ordinal and device code are saved for reference during module execution.

If validation is denied and the K/L display switch is OFF, MALET drops the control point. If the K/L display switch is ON, an error mesage appears and MALET retains the control point.

Restrictions:

- If tape preassignment is used (NOS/BE level 538 or higher; NOS level 531 or higher) only MNE=q is permitted in the ASSIGN directive for the device. The other parameters are obtained directly from the operating system. The unit assigned is the lowest numbered unit whose type matches the MNE=q specification.
- If the default access level is insufficient, a new access level (AL=gg) must be provided (refer to appendix C).
- If the desired device is listed in the equipment status table, EST=aaa is sufficient to request the device. For multichannel devices, MALET uses only the primary I/O channel.
- If the EST parameter is not provided, the channel, equipment and unit numbers (CH=bb, EQ=cc, UN=dd) are necessary to request the device. For LCN, MST, or CSU equipments, the following special restrictions apply:
  - For LCN, the logical trunk address (LTA=sss) must be provided instead of the unit number.
  - For MST, the MST physical address (MSTPH=p) or MST field must be provided instead of the unit number.
  - For CSU, the CSU number (CSU=k) must be provided instead of the unit number.

The following additional restrictions apply to the ASSIGN directive for 844 disk drives.

- If references outside the preallocated disk area are to be made, the disk pack serial number (SN=ffffff) is necessary.
- If the device is listed in the equipment status table and the EST parameter is not provided, the disk pack serial number (SN=ffffff) is necessary. The operating system provides the EST parameter and verifies the disk pack serial number.
- If the device is not listed in the equipment status table and the EST parameter is not provided, the device code (DC=ee) is necessary. The disk pack serial number (if included) is not verified and a *GO* command is required to begin using the disk pack.

AUTO Directive

Resume reading from input if previously suspended by STOP.

```
AUTO
```

Description:

The AUTO directive resumes a data input operation which was previously suspended by a STOP directive.

If executing a MALET diagnostic from a procedure file with the K/L display selected (two sources of control), a STOP from the console suspends reading the input file to give control to the user.

If executing a MALET diagnostic from a procedure file initiated from a terminal with I=alternate file name selected (two sources of control), a terminal interrupt (CTRL T) from the terminal suspends reading the input file to give control to the user.

CLRDS Directive

Disable options in the DS register.

```
CLRDS SE,RM,RC,BC,BM,BT,TE,CE,ME,S1,RT,S2

        SE      Stop on error; default=ON.

        RM      Repeat module; default=OFF.

        RC      Repeat condition; default=OFF.

        BC      Beginning of condition stop; default=OFF.

        BM      Beginning of module stop; default=OFF.

        BT      Beginning of test stop; default=OFF.

        TE      End of test stop; default=OFF.

        CE      Condition end stop; default=OFF.

        ME      Module end stop; default=OFF.

        S1      Programmable switch 1; default=OFF.

        RT      Repeat test; default=OFF.

        S2      Programmable switch 2; default=OFF.
```

Description:

The CLRDS directive sets specific switch options in the DS register to OFF. DS register switches are referenced by modules executing remotely in devices such as LCN and FSC.

CLRSW Directive

Disable options in the ES register.

```
CLRSW RC, SE, RM, PL, DL, RT, KL, SB

        RC      Repeat condition; default=OFF.

        SE      Stop on error; default=ON.

        RM      Repeat module; default=OFF.

        PL      Print log; default=ON.

        DL      Dayfile log; default=ON.

        RT      Repeat test; default=OFF.

        KL      System K/L display; default=OFF.

        SB      Standby PP option; default=OFF.  In this mode the PP
                driver is not assisted in escaping an I/O channel hang.
```

Description:

The CLRSW directive sets specific switch options in the ES register to OFF. The ES register switches are referenced by modules executing in a PP.

COMPILE or C Directive

Compile a MALET module.

```
COMPILE LIST, SOURCE

      C LIST, SOURCE

        LIST        Write source and binary listings of compiled module to
                    output file.

        SOURCE      Save source code.
```

Description:

The COMPILE directive calls the MALET compiler to compile instructions residing in the source code buffer.

Adding the LIST option causes the source and binary listings of the compiled module to be written to the output file.

Adding the SOURCE option causes the source code to be carried with the compiled module binary code when it is saved in the random file. The source code is carried regardless of errors encountered during module compilation.

Restrictions:

The SOURCE option defaults to OFF unless LO=S appears in the MALET call statement.

DEVICES Directive

Display a list of supported device codes.

```
DEVICES
```

Description:

The DEVICES directive displays a listing of valid device codes for use in the ASSIGN directive (refer to figure 2-4).

DROP Directive

Stop the PP driver and drop MALET.

```
DROP
```

Description:

The DROP directive halts any activity in the PP driver and exits from MALET

Description:

The DEVICES directive displays a listing of valid device codes for use in the ASSIGN directive (refer to figure 2-4).

```
MALET - MAINTENANCE APPLICATION LANGUAGE FOR EQUIP. TESTING

                    CODES FOR DEVICES

01=841 DISK                          02=7054/844-2 DISK
03=7054/844-4 DISK                   04=7154/844-2 DISK
05=7154/7155/844-4 DISK              06=819 DISK
07=FMD-885-1X DISK                   10=FSC-100 MBYTE
11=FSC-200 MBYTE                     12=FSC-317 MBYTE
13=RESERVED                          14=7155/855-42 DISK (DEMA)
15=7155/885-42 DISK (PFMDW)          16=7155/885-1X DISK (LSFMD)
17=RESERVED                          20=405 CARD READER
21=415 CARD PUNCH                    22=512 LINE PRINTER
23=580-12 LINE PRINTER               24=580-16 LINE PRINTER
25=580-20 LINE PRINTER               26=580-12 PFC LINE PRINTER
27=580-16 PFC LINE PRINTER           30=580-20 PFC LINE PRINTER
31-37=RESERVED                       40=60X 7 TRK TAPE
41=65X 7 TRK TAPE                    42=66X 7 TRK TAPE
43=RESERVED                          44=67X 7 TRK TAPE
45=FSC 7 TRK TAPE                    46=RESERVED
47=RESERVED                          50=60X 9 TRK TAPE
51=65X 9 TRK TAPE                    52=66X 9 TRK TAPE
```

```
                    CODES FOR DEVICES

53=RESERVED                          54=67X 9 TRK TAPE
55=67X GCR 9 TRK TAPE                56=FSC 9 TRK TAPE
57=639 9 TRK TAPE                    60=6671 MUX
61=6676 MUX                          62=2550-100 EMULATING 6671
63=2550-100 EMULATING 6676           64=2550
65=7077-1 LCC                        66=6673 DATA SET
67=6683 COUPLER                      70=DDP
71=ECS COUPLER                       72=RESERVED
73-77=RESERVED                       100=MSS-MST
101=MSS-CSU                          102-103=RESERVED
104=CCC/MASSTOR SUBSYSTEM            105-107=RESERVED
110=7255/834 DISK(FSDI)              111-7255/836 DISK (FSDII)
115=CCC/895 DISK                     116-177=RESERVED
200=CYBERPLUS RING PORT              201-277=RESERVED
300=LCN-NAD LOCAL                    301=LCN NAD REMOTE
302-377=RESERVED                     400=CIU OUTPUT CONTRL
401=CIU INPUT CONTRL                 402-7677=RESERVED
500=CYBER/UNIBUS INTERFACE           501-7677=RESERVED
7700-7777=QSE


LINE=000        PPU STATUS=STOPPED          MODULE=TST01
```

Figure 2-5. Sample K/L Displays for DEVICES Directive

DUMP Directive

Copy random file to a sequential binary file.

```
DUMP
```

Description:

The DUMP directive dumps the contents of the random file to the sequential binary file specified by the B option on the MALET call statement.

GO or G Directive

Resume execution of the PP driver.

```
GO
 G
```

Description:

The GO or G directive resumes PP driver execution after it has performed a normal halt.

LINE or L Directive

Choose line number to receive next source code.

```
LINE nnn
   L nnn

          nnn        Source code buffer line number; values 0 through
                     167B are valid.
```

Description:

The LINE or L directive sets the data input pointer of the source code buffer to the line number specified by nnn. The next source code input is placed on this line.

Restrictions:

A valid line number must appear in the directive, there is no default.

LIST Directive

List n lines beginning with current line number.

LIST n

n	Number of lines to be listed; values 0 through 167_8 are valid.

Description:

The LIST directive displays n consecutive module source code lines starting with the number in the data input pointer. If n exceeds the number of lines remaining in the module, the directive displays only those lines available.

LOAD Directive

Load specified module from the random file.

LOAD MNEnnn

MNE	Three letter mnemonic for the module test series.
nnn	Module number.

Description:

The LOAD directive transfers a module from the random file to the module buffer and displays (if available) the first page of source code.

MALET Directive

Display MALET executive directives.

```
MALET
```

Description:

The MALET directive displays a list of all the valid executive directives (refer to figure 2-5).

MODULES Directive

Display a listing of modules in the random file.

```
MODULES
```

Description:

The MODULES directive displays a list of modules currently existing in the random file (refer to figure 2-6). Modules which are preceded by an asterisk are identified as containing assembly errors.

```
        MAINTENANCE APPLICATION LANGUAGE FOR EQUIP. TESTING

DIRECTIVE                COMMENTS

AUTO       RESUME READING FROM INPUT FILE
CLRDS      CLEAR DS REGISTER SWITCHES *CLRDS SE,RT,RM,RC*
CLRSW      CLEAR SWITCHES *CLRSW SE,RT,DL,RM,PL,SB,KL,RC*
COMPILE    COMPILE SOURCE BUFFER *COMPILE,LIST,SOURCE*
DEVICES    DISPLAY DEVICE CODES *DEVICES*
DROP       DROP MALET PPU AND CPU
DUMP       DUMP RANDOM FILE TO BINARY FILE *DUMP*
GO         RESUME EXECUTION OF MODULE THAT WAS STOPPED *GO*
LINE       SET LINE FOR NEXT ENTRY *LINE=NNN*
LIST       LIST N LINES ON OUTPUT FILE. *LIST NN*
LOAD       LOAD MODULE *LOAD MNENN*
MALET      DISPLAY MALET EXECUTIVE DIRECTIVES *MALET*
MODULES    DISPLAY MODULES CURRENTLY ON RANDOM FILE *MODULES*
MOVE       *MOVE AAA UP BBB LINES*
MOVE       *MOVE AAA DOWN BBB LINES*

DIRECTIVE                COMMENTS
PARAM      CHANGES PARAMETERS *PARAM P0=1234,P1=1234,P9=1234,
           TL=1234,LL=1234*   TL=PPU TIME LIMIT LL=LINE LIMIT
PICTURE    PRINT K/L DISPLAY ON OUTPUT FILE *PICTURE=TYPE*
PPU        DISPLAY PPU MESSAGES/PLOTS *PPU*
PUNCH      PUNCH SOURCE BUFFER  *PUNCH*
PURGE X    PURGE X FROM RANDOM FILE
RUN        RUN MODULE(S) *RUN MNE (01,02,03,04,10,07)*
SCRATCH    CHANGE SCRATCH FILE NAME *SCRATCH=LFN*
SEND       SEND MESSAGE TO B DISPLAY *SEND,CONTENT OF MSG*
SETDS      SET DS REGISTER SWITCHES *SETDS,SE,RM,RT,RC*
SETSW      SET SWITCHES *SETSW,PL,DL,SE,RM,RT,SB,KL,RC*
SOURCE     DISPLAY SOURCE FOR MODULE *SOURCE*
STOP       STOP PPU EXECUTION USED ONLY ON K/L DISPLAY
+          ADVANCE PAGE ON K/L DISPLAY
-          DECREMENT PAGE ON K/L DISPLAY


 LINE=000          PPU STATUS = STOPPED     MODULE = TST01
```

Figure 2-6. Sample K/L Display for MALET Directive

```
MALET - MAINTENANCE APPLICATION LANGUAGE FOR EQUIP. TESTING

                    MODULES ON RANDOM FILE

 TST00      TST02      TST03      TST04      TST05
 TST06      TST07      TST08      TST09B     TST10
 TST11      TST12      TST13     *TST14      TST15
 TST16     *TST17      TST18A     TST19      TST20
 TST21      TST22     *TST23      TST24      TST25
 TST26      TST27      TST28      TST290     TST291
 TST31      TST32      TST33      TST34      TST35




 LINE=000        PPU STATUS = STOPPED        MODULE = TST00
```

Figure 2-7. Sample K/L Display for MODULES Directive

MOVE or M Directive

Move source code lines.

```
MOVE aa UP bb

MOVE aa - bb

   M aa UP bb

   M aa - bb

MOVE aa DOWN bb

MOVE directive format continued.

MOVE aa + bb

   M aa DOWN bb

   M aa + bb

        aa      First line number to be shifted.

        bb      Number of line positions shifted.
```

Description:

The MOVE or M directive moves source lines up or down within the source code buffer. It is normally used to make space for modifications within a module or to eliminate unwanted lines. A move up overwrites data currently existing on lines to which the move is made. A move down eliminates those lines which shift past the end of the module buffer. The characters (-) and (+) may be substituted for UP and DOWN.

PARAM or P Directive

Alter and display parameters passes to the PP registers.

```
PARAM P0=aa,P1=bb,P2=cc,P3=dd,P4=ee,P5=ff,P6=gg,P7=hh,P8=ii,P9=jj,LL=rr,
      TL=ss

    P P0=aa,P1=bb,...,LL=rr,TL=ss
```

aa,bb,...,jj	Value to be placed in the specified parameter register.
rr	Number of executed lines allowed before MALET displays the line limit GO/DROP message; no default.
ss	Number of seconds the PP driver is allowed to execute before being forced to stop; default=5 seconds. The TL parameter is used only if operating in the terminal mode. A STOP is sent to the module every TL seconds to allow the user to issue a TERMINAL INTERRUPT (CONTROL T under NOS) and read an input. If no terminal interrupt is pending, a resume execution is automatically issued for the module by the executive.

Description:

The PARAM directive changes specified parameters and/or lists current parameters to the display. Supplying no parameters in the command format causes only the display to be listed.

Method of Execution:

Program parameters are passed to the PP driver as a module is brought into execution with a RUN directive. The K/L display changes to the parameter display when the PARAM directive is processed (refer to figure 2-7).

Restrictions:

The upper 4 bits of hexadecimal entries may only be accessed by product overlays which support 16-bit parameter entries such as LCN.

```
MALET - MAINT. APPLICATION LANGUAGE FOR EQUIP. TESTING

                         PARAMETERS

   P0 = 0000            P1 = 0000            P2 = 0000
   P3 = 0000            P4 = 0000            P5 = 0000
   P6 = 0000            P7 = 0000            P8 = 0000
   P9 = 0000            TL = 0005            LL = 1750

                         SWITCH STATUS

 ON - (SE) STOP ON ERROR          OFF - (RM) REPEAT MODULE
OFF - (RT) REPEAT TEST            OFF - (SB) STANDBY PPU FOR I/O
 ON - (PL) PRINT LOG               ON - (DL) DAYFILE LOG
OFF - (KL) K/L DISPLAY            OFF - (RC) REPEAT CONDITION

                         ASSIGN VALUES

DC=0000     EST=0000    CH=0000     EQ=0000     UN=0000    AL=0000

                    ADDITIONAL ASSIGN VALUES
              SN=000000  SHARE=NO   SCRATCH=SCRATCH
             LTA=0000   TCU=0000   AC=0000   BFZ=0000

                    DS REGISTER SWITCH STATUS

 ON - (SE) STOP ON ERROR          OFF - (RT) REPEAT TEST
OFF - (RM) REPEAT MODULE          OFF - (RC) REPEAT CONDITION
OFF - (BC) BEGINNING OF COND      OFF - (BM) BEGINNING OF MODULE
OFF - (BT) BEGINNING OF TEST      OFF - (TE) TEST END STOP
OFF - (CE) CONDITION END STOP     OFF - (ME) MODULE END STOP
OFF - (S1) PROGRAMMABLE SW 1      OFF - (S2) PROGRAMMABLE SW 2

              ADDITIONAL ASSIGN VALUES FOR MSS ONLY

               MSA=0     MSAID=00        C=00/00
               CSU=0     CSUID=00
               MST=0     MSTID=00
                         MSTPH=0

      LINE=000        PPU STATUS = STOPPED        MODULE =
```

Figure 2-8. Sample K/L Display for PARAM Directive

PICTURE Directive

Send a picture of the K/L display to the output file.

```
PICTURE SOURCE,PARAM,DEVICES,MALET,PPU,MODULES

        SOURCE      Print a listing of SOURCE data (refer to figure 2-8 later in
                    this section).

        PARAM       Print a picture of the PARAM display (refer to figure 2-7).

        DEVICES     Print a picture of the DEVICES display (refer to figure 2-4).

        MALET       Print a picture of the MALET directives display (refer to
                    figure 2-5).

        PPU         Print a picture of the PPU display.

        MODULES     Print a picture of the MODULES display (refer to figure 2-6).
```

Description:

The PICTURE directive sends a copy of the specified display(s) to the output file. If no parameter is specified, the current K/L display is sent to the output file.

NOTE

When running MALET from a remote terminal, check the PP via PICTURE PPU for possible additional error information.

PPU Directive

Display PP messages/plots.

```
PPU
```

Description:

The PPU directive displays PP information on the K/L display.

PUNCH Directive

Copy current source code to the random file.

```
PUNCH
```

Description:

The PUNCH directive copies information from the source code buffer into the PUNCH file.

PURGE Directive

Delete a module from the random file.

```
PURGE MNEnnn

        MNE         Three letter mnemonic for module test series.
        nnn         Module number.
```

Description:

The PURGE directive purges the specified module from the random file.

Restrictions:

If no module number (nnn) is specified, all modules which begin with MNE are deleted.

The PURGE directive only deletes the module name from the index. To remove the binary it is necessary to dump the modules to the binary file, return to RFILE, and reload the modules.

RUN or R Directive

Execute a MALET module in the PP driver.

```
RUN MNE
RUN MNE(aaa,bbb,...,nnn)
  R MNE
  R MNE(aaa,bbb,...,nnn)
```

MNE	Three letter mnemonic for the module test series.
aaa,bbb,...,nnn	Module numbers.

Description:

The RUN or R directive initiates module execution. The format of the directive may specify any combination of the following:

- A test series.
- Particular modules within a test series.
- Order in which modules are to be run.

Six examples demonstrate features of the directive.

Example 1: Run modules 00 through 34 in specified order regardless of END/EXIT statements.

```
RUN MNE(00,01,02,03,04,05,06,07,...,29,30,31,
32,33,34)
```

NOTE

The right parenthesis is not present on the first line of the entry. The directive automatically continues reading module numbers from the second line until the right parenthesis is encountered.

Example 2: Run modules 00 and 01 twice regardless of END/EXIT statements.

```
RUN MNE(00,01,00,01)
```

Example 3: Run test series MNE beginning with the lowest numbered module.

```
RUN MNE
```

Example 4: Run test series MNE beginning with module 20.

```
RUN MNE20
```

Example 5: Run only test module 20.

```
RUN MNE(20)
```

Example 6: Run lowest numbered module of currently loaded test.

```
RUN
```

Restrictions:

The following restrictions apply to the RUN or R directive.

- When only the test series is specified (MNE), the test modules execute in numerical order. Testing stops following each module in which an END command is encountered and may be resumed with a GO directive.
- Leading zeros are necessary when specifying module numbers.
- If more than 60 characters are required to specify a module test sequence, a right parenthesis should only appear following the last module of the sequence.
- A maximum of 100 modules may be specified on one RUN or R directive.

SCRATCH Directive

Change the name of the SCRATCH file.

```
SCRATCH=name

        name        File designated as the scratch file.
```

Description:

The SCRATCH directive designates the indicated file as the scratch file to be accessed by SYSRD/SYSWR/SYSREW commands.

SEND Directive

Send a message to the B display and output file.

```
SEND message

        message     Character string to be delivered to B display and
                    output file.
```

Description:

The SEND directive copies the accompanying message to both the B display and the output file.

Restrictions:

On NOS at PSR level 528 or higher, and on all NOS/BE operating systems, the message flashes on the B display awaiting an operator x.GO command where x is the control point. The message does not flash on NOS operating systems below PSR level 528 but still requires the x.GO command. After the x.GO command is issued, an OPERATOR GAVE GO message appears on the B display and is entered in the print file which appears at the remote terminal.

SETDS Directive

Enable options in the DS register.

```
SETDS SE,RM,RC,BC,BM,BT,TE,CE,ME,S1,RT,S2

        SE      Stop on error; default=ON.
        RM      Repeat module; default=OFF.
        RC      Repeat condition; default=OFF.
        BC      Beginning of condition stop; default=OFF.
        BM      Beginning of module stop; default=OFF.
        BT      Beginning of test stop; default=OFF.
        TE      End of test stop; default=OFF.
        CE      Condition end stop; default=OFF.
        ME      Module end stop; default=OFF.
        S1      Programmable switch 1; default=OFF.
        RT      Repeat test; default=OFF.
        S2      Programmable switch 2; default=OFF.
```

Description:

The SETDS directive sets specific switch options in the DS register to ON. The DS register switches are referenced by modules executing remotely in devices such as LCN and FSC.

SETSW Directive

Enable switch options in the ES register.

```
CLRSW RC,SE,RM,PL,DL,RT,KL,SB

        RC      Repeat condition; default=OFF.

        SE      Stop on error; default=ON.

        RM      Repeat module; default=OFF.

        PL      Print log; default=ON.

        DL      Dayfile log; default=ON.

        RT      Repeat test; default=OFF.

        KL      System K/L display; default=OFF.

        SB      Standby PP option; default=OFF.  In this mode the PP
                driver is not assisted in escaping an I/O channel hang.
```

Description:

The CLRSW directive sets specific switch options in the ES register to ON. The ES register switches are referenced by modules executing in a PP.

SOURCE Directive

Display a page of the source code buffer.

```
SOURCE
```

Description:

The SOURCE directive displays a page of the source code buffer (refer to figure 2-8). The (+) and (-) directives may be used to page through the source code buffer.

```
MALET - MAINTENANCE APPLICATION LANGUAGE FOR EQUIP. TESTING


000=    MODULE TST01,3000(20,21,22)
001=01  FORMAT SAMPLE MODULE TO DISPLAY MSGS ON LINE=*OCT
002=02  B2=0
003=03  B1=0
004=04  MSG 1(B1) to LINE B1
005=    GOTO 4 WHILE (B1+1.NE.51B)
006=    GOTO 3 WHILE (B2+1.NE.100B)
007=    END 2
010=
011=
012=
013=
014=
015=
016=
017=


 .
 .
 .


045=
046=
047=


LINE=010      PPU STATUS = STOPPED        MODULE = TST01
```

Figure 2-9. Sample K/L Display for the SOURCE Directive

STOP or S Directive

Stop execution of a MALET module.

```
STOP
```

Description:

The STOP directive halts execution of a MALET module in the PP driver.

Method of Execution:

The STOP directive sets the PP stop switch and sends it to the PP driver. The PP checkpoints the currently executing module at the next logical breakpoint and drops.

+ Directive

Increment display one page.

```
+
```

Description:

The + directive increments the existing K/L display one page (23 or 40 lines, depending on operating system version).

- Directive

Decrement display one page.

```
-
```

Description:

The - directive decrements the existing K/L display one page (23 or 40 lines, depending on operating system version).

INPUT/OUTPUT CONFIDENCE LEVEL TESTS 4

TIO

TIO is a confidence level test designed to provide an on-line check of I/O operations and to assess the performance of the mass storage devices, tapes, and disks on CYBER 70, CYBER 170, CYBER 180, and 6000 Computer Systems. Testing is performed through the use of FORTRAN Extended statements BUFFERIN, BUFFEROUT, REWIND, ENDFIL, and BACKSPACE. FORTRAN I/O subprograms used are UNIT and LENGTH. The test is composed of two sections. The first is the sequential section in which the device is tested by sequentially writing and reading patterns of preset and random lengths. The second is the random section in which the device is tested by performing random operations with random data of random lengths.

OPERATING PROCEDURES

TIO is a control card callable program requiring parameters selecting the operation and functions to be performed.

PARAMETERS

SEQ=N

Defines the decimal number of sequential records to be written on the test file(s) during the sequential section. If SEQ=0, the sequential section is skipped. Default is 200. The maximum value for N is 9999999.

RAN=N

Defines the decimal number of random operations to perform during the random section. If RAN=0, the random section is skipped. Default is 1000. The maximum value for N is 9999999.

FILES=N
FILES=LFN1/LFN2/LFN3/LFN4/LFN5

FILE=N where N is the value 1 through 5 used to turn on N test file(s) with default logical file name(s): TAPE1, TAPE2, TAPE3, TAPE4, and TAPE5. The alternate form of this parameter is used if the user wishes to turn on the test file(s) and change the logical file name(s) from the default. Default of FILE is 1 and FILES is TAPE1.

OUTPUT=LFN
O=LFN

Used to change the logical file name of the output file. Default of OUTPUT is OUTPUT and O is OUTPUT.

PASSES=N

Defines the decimal number of passes that the sequential and random sections execute. One pass includes both binary and coded modes if the mode parameter is not used. When the mode parameter is used, one pass consists of the selected mode. Default is 3. The maximum value for N is 9999999.

LIST=N

The value for N is the decimal number of errors to be written to the output file before terminating. Default is 100. The maximum value for N is 9999999.

MAXREC=N

N equals the maximum record length to be written on the test file(s) in 60-bit central memory words. Default is 2000. The maximum value for N is 2000.

MINREC=N

N equals the minimum record length to be written on the test file(s) in 60-bit central memory words. Default is 1. The minimum value for N is 1.

PATTERN=NN/AAAA/BBBB/CCCC/DDDD/EEEE
PAT=NN/AAAA/BBBB/CCCC/DDDD/EEEE

The pattern parameter is used if a certain pattern is desired for the data written on the test file(s). The value for NN can be any number from 1 through 59 decimal and defines the number of bits to be used in the pattern. The values for AAAA through EEEE must be four octal digits, right-justified, representing the pattern desired. For example:

- PAT=12/3210 stores 321032103210 throughout the write buffer.
- PAT=6/0012 stores 121212 throughout the write buffer.
- PAT=30/0012/3456/7077 stores 123456707712345 throughout the write buffer.
- PAT=8/0376 stores 775773767757737 throughout the write buffer.

To force random data for the sequential section, use PAT=RAN, PAT=RANDOM, PATTERN=RAN, or PATTERN=RANDOM. Default is PRESET/RANDOM.

When operating in the coded (BCD) mode, all 00B characters are replaced by 33B; if byte 4 of a 60-bit word is 6362B, it is replaced by 3333B.

HISTORY=N

If present on the call statement, TIO lists up to the last decimal N operation performed on the test file(s) preceding an error. If N is greater than 50, all of the operations listed are listed on the output file. Default is 0.

WRITE

Used to run the sequential section in a write-only mode to create test file(s). The test file(s) can be used in another run of TIO through the use of the READ parameter. The random section is disabled if this parameter is used. The mode parameter must be used if coded test file(s) is desired. Default is OFF.

READ

Used to run the sequential section in a read-only mode on test file(s) prewritten through the use of the WRITE parameter. The random section is disabled if this parameter is used. The mode parameter must be used if the test file(s) is in coded format. Default is OFF.

MODE=BCD
MODE=BIN

Used to select coded or binary mode where BCD is coded and BIN is binary. Default is both used.

SAMPLE JOBS

Example 1: This job runs TIO default parameters.

```
JOB/USER/ACCOUNT CARDS.
REQUEST,TAPE1.
TIO.
6/7/8/9
```

Example 2: This job runs 10 passes of the sequential and random sections with record lengths of 10 words and lists all errors on a file called SCRATCH.

```
JOB/USER/ACCOUNT CARDS.
REQUEST,TAPE1.
TIO(O=SCRATCH,PASSES=10,MINREC=10,MAXREC=10)
6/7/8/9
```

Example 3: This job runs the sequential and random sections on a TAPE22, in binary mode and terminates if 10 errors are test file, encountered or end of test is reached.

```
JOB/USER/ACCOUNT CARDS.
REQUEST,TAPE22.
TIO(FILES=TAPE22,LIST=10,MODE=BIN)
6/7/8/9
```

Example 4: This job runs the sequential and random sections of five system scratch test files and lists all of the operations on the output file. The output provided by this job is large.

```
JOB/USER/ACCOUNT CARDS.
TIO(FILES=5,HISTORY=ALL)
6/7/8/9
```

Example 5: This job performs a write operation only, with a pattern of all ones in binary mode and loops for the default number of passes.

```
JOB/USER/ACCOUNT CARDS.
REQUEST,TAPE1.
TIO(PAT=12/777,WRITE)
6/7/8/9
```

Example 6. This job reads TAPE1 created in job 5 and reports any data errors encountered with the ones pattern.

```
JOB/USER/ACCOUNT CARDS.
REQUEST,TAPE1.
TIO(PAT=12/7777,READ)
6/7/8/9
```

Example 7: This job runs the sequential and random sections writing and reading a pattern of 00200400100 octal on default system scratch file TAPE1, and reports the last five operations preceding any error encountered.

```
JOB/USER/ACCOUNT CARDS.
TIO(PAT=8/0020,HISTORY=5)
6/7/8/9
```

PARAMETER ERROR MESSAGES

UNRECOGNIZED PARAMETER...(parameter)

Unrecognized parameter on call statement.

UNRECOGNIZED PARAMETER...(PARAMETER=parameter)

Unrecognized parameter on call statement.

DECIMAL CONVERT ERROR PARAMETER...(PATTERN=A/1234)

Value for A must be decimal.

VALUE TOO HIGH PARAMETER...(FILES=6)

Value is above maximum allowed for parameter.

OCTAL CONVERT ERROR PARAMETER...(PATTERN=60/ALPHA)

Alpha value must be octal.

MAXREC=1999 MINREC=2000...MAX LESS THAN MIN

Maximum cannot be less than minimum.

...PROGRAM ABORT...BAD PARAMETERS...

One or more parameter errors on control card (refer to output).

VALUE TOO LOW PARAMETER...(FILES=0)

Value is below minimum allowed for parameter.

NO SECTION SELECTED (SEQ=0, RAN=0)

User must select at least one section.

FILE NAME ABC=ABC DUPLICATE NAMES

Two file names are the same on either the FILES or OUTPUT parameters.

SEQ AND RAN ERROR MESSAGES

In the following error messages, the alphabetic codes used are:

R	Record number.
L	Record length.
W	Word number.
E	Expected data.
A	Actual data.
D	Difference.

SEQ PASS=0001 TAPE1 BIN PARITY ERR R=0001 L=2000

SEQ PASS=0001 TAPE2 BCD DATA ERR R=0001 W=1999 E=X D=X

X Twenty octal digits.

SEQ PASS=0002 TAPE1 BIN EOF EXPECTED R=0002 L=0

SEQ PASS=0003 TAPE2 BCD EOF NOT EXPECTED R=0002 L=2000

RAN PASS=0001 TAPE5 BIN LENGTH ERR R=0001 E=2000 A=1999

RAN PASS=0002 TAPE1 BCD GT 10 DATA ERRS R=0002

RAN PASS=0002 TAPE2 BIN M/R CMP ERR R=0002 M=X R=Y

X	Twenty octal digits of the model word.
Model word	12/FWA of write, 12/record length, 36/record number.
Y	Twenty octal digits of the record word.
Record word	12/FWA of write, 12/record length, 36/record number.

```
ERRORS EXCEED LIST = 000010 RUN ENDED

SEQ PASS=0002 TAPE1 BIN REC-NO ERR E=0002 A=0003 CONT. BY
SETTING E=A
```

Record number read does not match internal model so TIO attempted recovery by setting expected record to actual.

```
SEQ PASS=0002 TAPE1 BIN REC-NO ERR E=0003 A=0001 CAN NOT
CONTINUE
```

Record number read does not match expected value and correction would not work. Program calls EXIT. The following message is entered in the dayfile.

```
REC NUMBER ERR UNRECVD..CALL EXIT
```

INSTALLATION DECK

Refer to CML Installation in section 1.

TEST FILE RECORD FORMAT

The first 60 bits of data form a control word containing the first word address of the output from the write buffer, record length, and record number as follows:

```
         5         4         3         2         1

9876543210987654321098765432109876543210987654321098765432109876543210

*--FWA------**---LENGTH--**-----------RECORD NUMBER--------*
```

Data is checked by comparing the data in the read buffer with the data in the write buffer. If a record has more than 10 errors, further data checking on that record is ended with the message:

```
GT 10 DATA ERRORS
```

REGEN

REGEN is an on-line confidence test program written in the FORTRAN Extended language and designed to run on the CYBER 70, CYBER 170/180, and 6000 Series Computer Systems. Its two major functions are to provide validation of system data paths by passing checksummed data across selected data paths and verifying that data does not deteriorate due to hardware or software errors, and to verify that the central processor and central memory units can execute diagnostic tests FST and ALX, producing answers that agree with a master data base of answers. Any answers detected are entered into an error log assigned as TAPE99.

Although REGEN is normally run as a user job concurrently with other user jobs, it can be used more effectively in a dedicated system where several copies of REGEN are the only jobs. This allows validation of the total system under voltage of clock margins or while shocktesting.

REGEN can operate in any one of six basic modes selected by data cards in an input file. The modes are:

Generate mode	Generate a data base on a disk or tape file.
Copy mode	Copy a data base from one file to another.
Regenerate mode	Regenerate a data base and compare it with a previous data base file.
Return mode	Return a file (close/unload).
Runpass mode	Run a specific pass of diagnostic tests FST and ALX and print the results.
Isolate mode	Analyze an error log from a previous REGEN run that had errors.

In selecting a specific mode, a distributor routine reads the input card deck and makes the selection. The selected mode, using FORTRAN Extended, calls and supplies parameters to six COMPASS subroutines used to accomplish the tasks of the various modes of operation. Refer to figure 4-1.

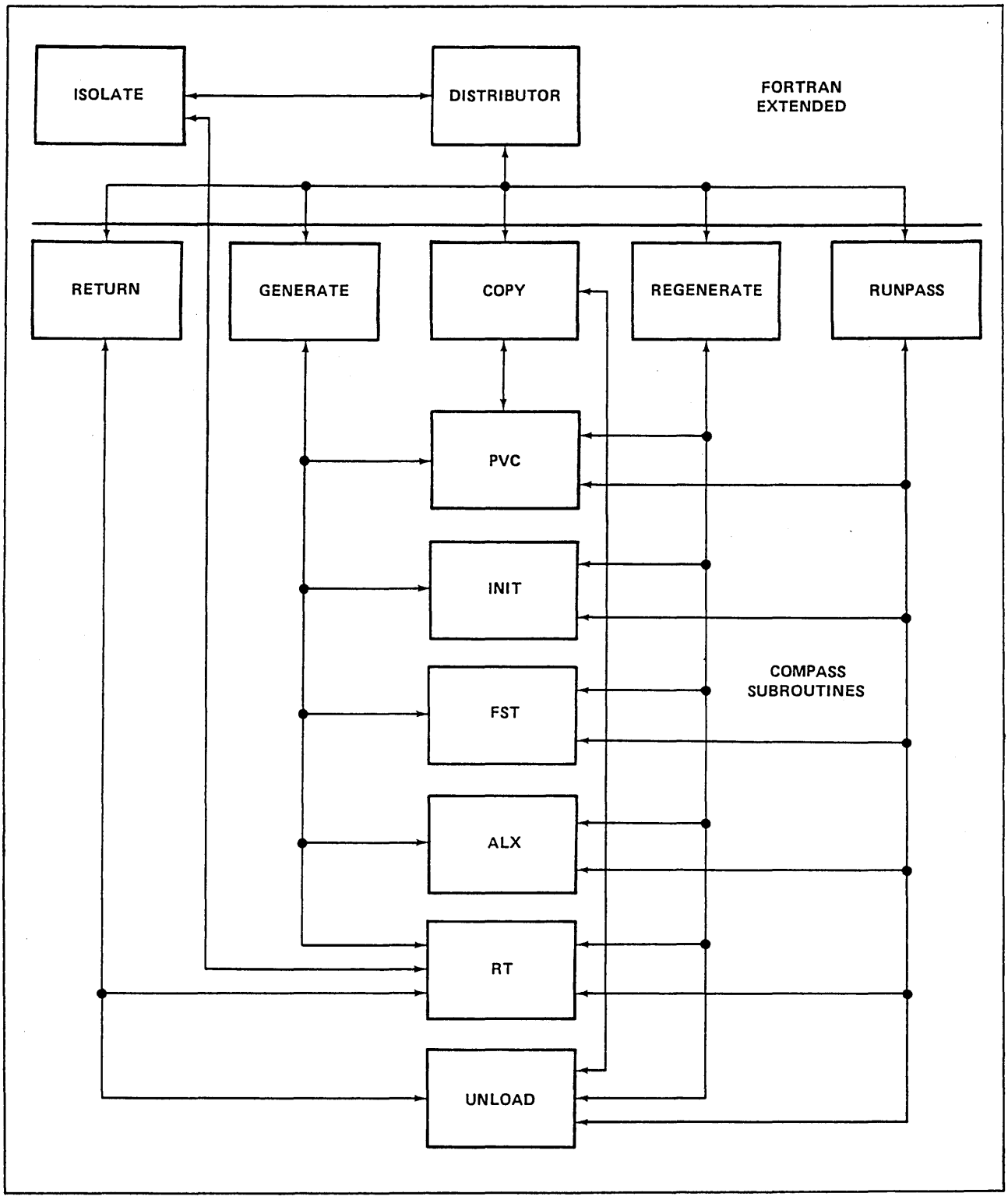

Figure 4-1. REGEN Modes

Subroutine PVC (parity vector checksumming) is a high-speed checksum program that generates the checksum of data blocks of any length. It generates a three-word checksum for each 60 words or less of input data: straight-line, left-diagonal, and right-diagonal logical sums (figure 4-2). This technique allows isolation of a single bit failure to the word and bit that failed. The code is ultrafast, providing checksumming capability on the model 74 at about 1.5 microseconds per checksummed word.

Subroutine INIT is used to reestablish the starting base numbers for a specific pass of subroutines FST and ALX. This allows the re-creation of specific passes that are identical to previously run passes.

Subroutine FST is designed to run random instructions back-to-back in a conflicting mode and then spaced apart in a nonconflicting mode. FST then determines if any variations in results occurred (figure 4-3).

Subroutine ALX is designed to run at a maximum instruction issue rate by first running in the instruction stack and then at a slower out-of-stack mode. ALX then determines if any variations in answers occurred (figure 4-4).

Subroutine RT is used to calculate wall-clock timing for each operation and the total run.

Subroutine UNLOAD is used to rewind and unload a file. This releases the unit or disk space for other operations after a data path has been validated.

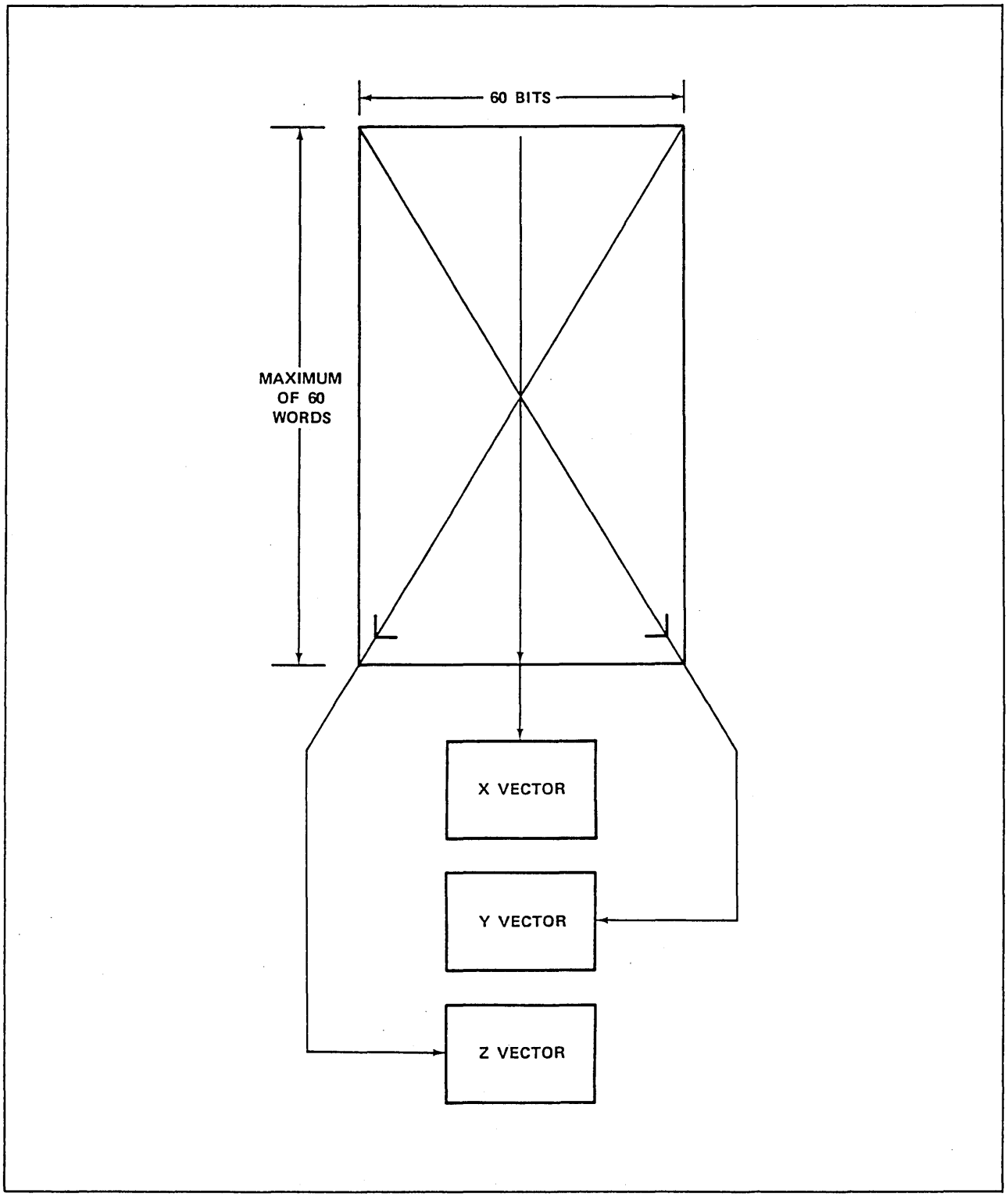

Figure 4-2. PVC Checksumming

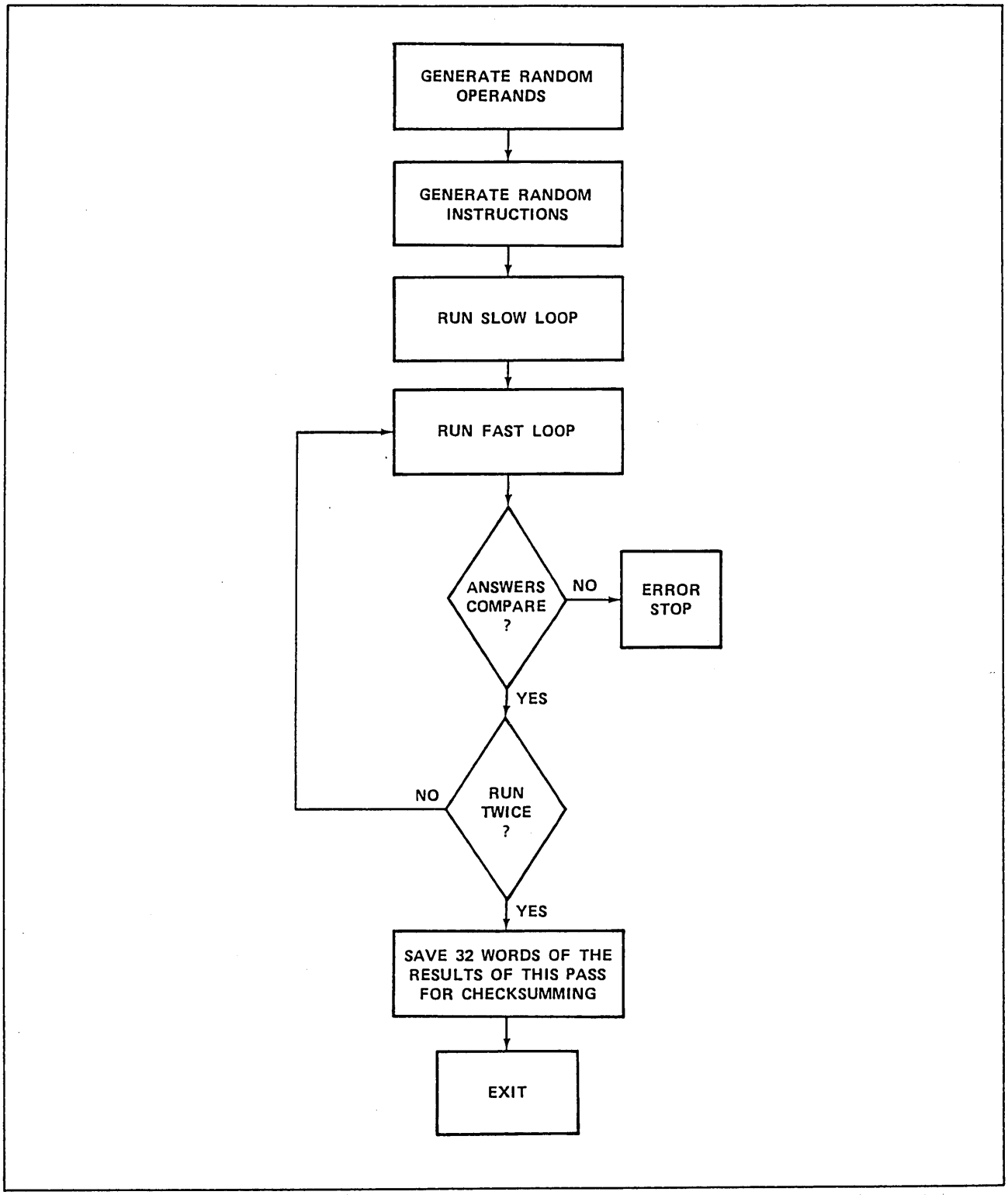

Figure 4-3. FST Flowchart

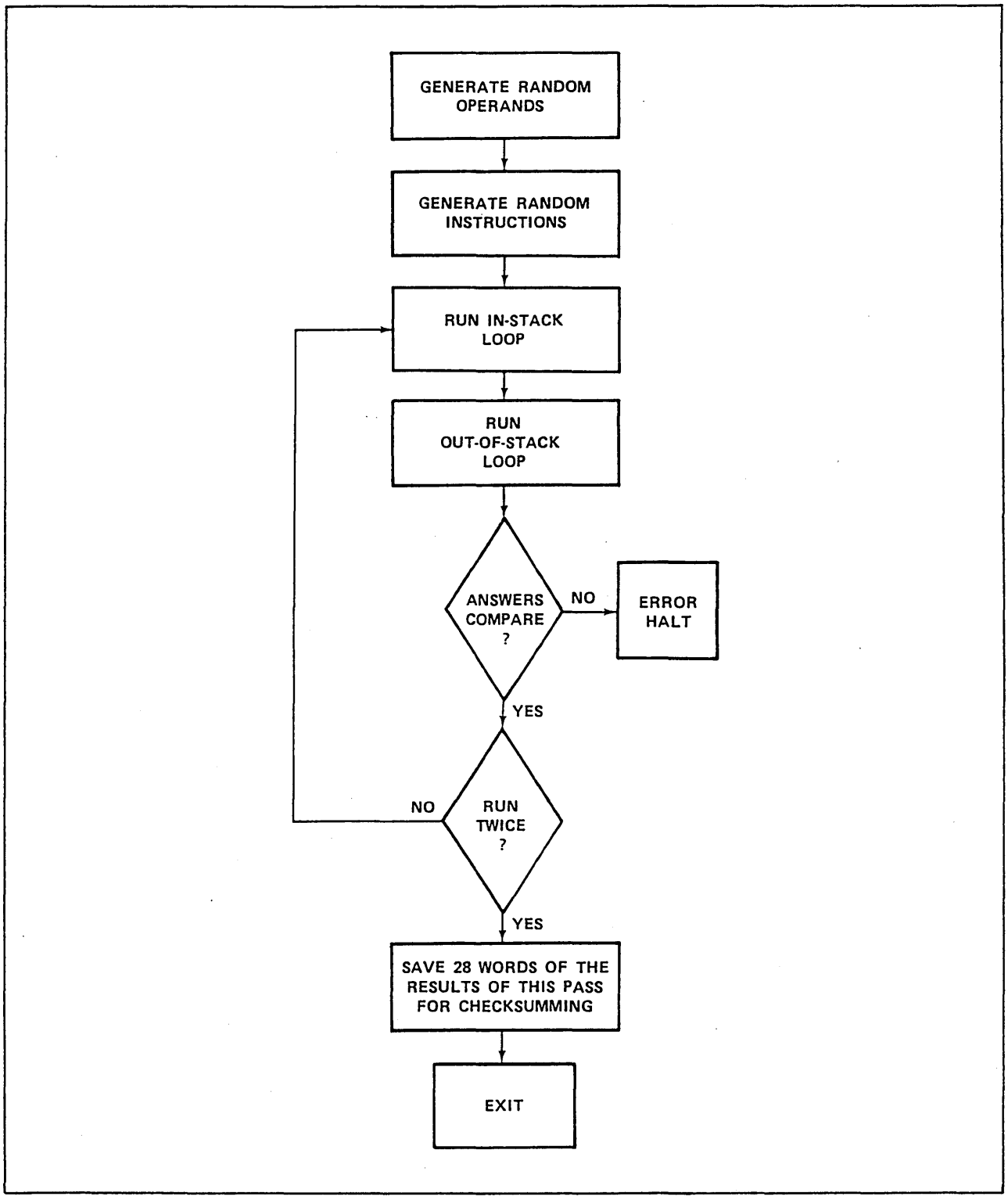

Figure 4-4. ALX Flowchart

MODES OF OPERATION

Following are the modes of operation for the REGEN program.

Generate Mode

The generate mode, shown in figure 4-5, is provided to build an initial master data base. To check compatibility between sites in a network, only one site should run the initial generate. Copies of the master data base should then be made using the copy mode to build data base copies for all other sites in the network.

In building the master data base, one pass each of FST and ALX is run. This tests the central processor and central memory and provides 60 words of resultant data as follows:

Words 1 through 16	FST fast loop answers (exchange package format).
Words 17 through 32	FST fast loop instructions.
Words 33 through 48	ALX fast loop answers (exchange package format).
Words 49 through 53	ALX fast loop instructions.
Words 54 through 59	ALX random operands.
Word 60	ALX pass count +1.

Subroutine PVC is then called to generate a three-word checksum (straight-line, left-diagonal, and right-diagonal) of these 60 words to represent this pass. This procedure is repeated 20 times until 60 words of checksums have been generated. Subroutine PVC is then called again to generate a three-word checksum of the checksums for a total of 63 words. A 64th word, called the position data word, is added which identifies the job name, file number, record number, and block number for this particular block. This word is constructed as follows:

Bits 0 through 2	Block number (1 through 7).
Bits 3 through 14	Record number (1 through 7246B).
Bits 15 through 23	File number (1 through 32) that represents tape 1 through 32.
Bits 24 through 59	Unique job name identifier supplied by the user on the first input card.

The total process is repeated eight times to generate a block of 512 words; 511 of these words are then written to tape x as a record. The last position data word is not written.

The entire process is repeated the number of times requested on the first data card to REGEN. A maximum of 3750 records fills a 2400-foot reel of tape at 800 bpi.

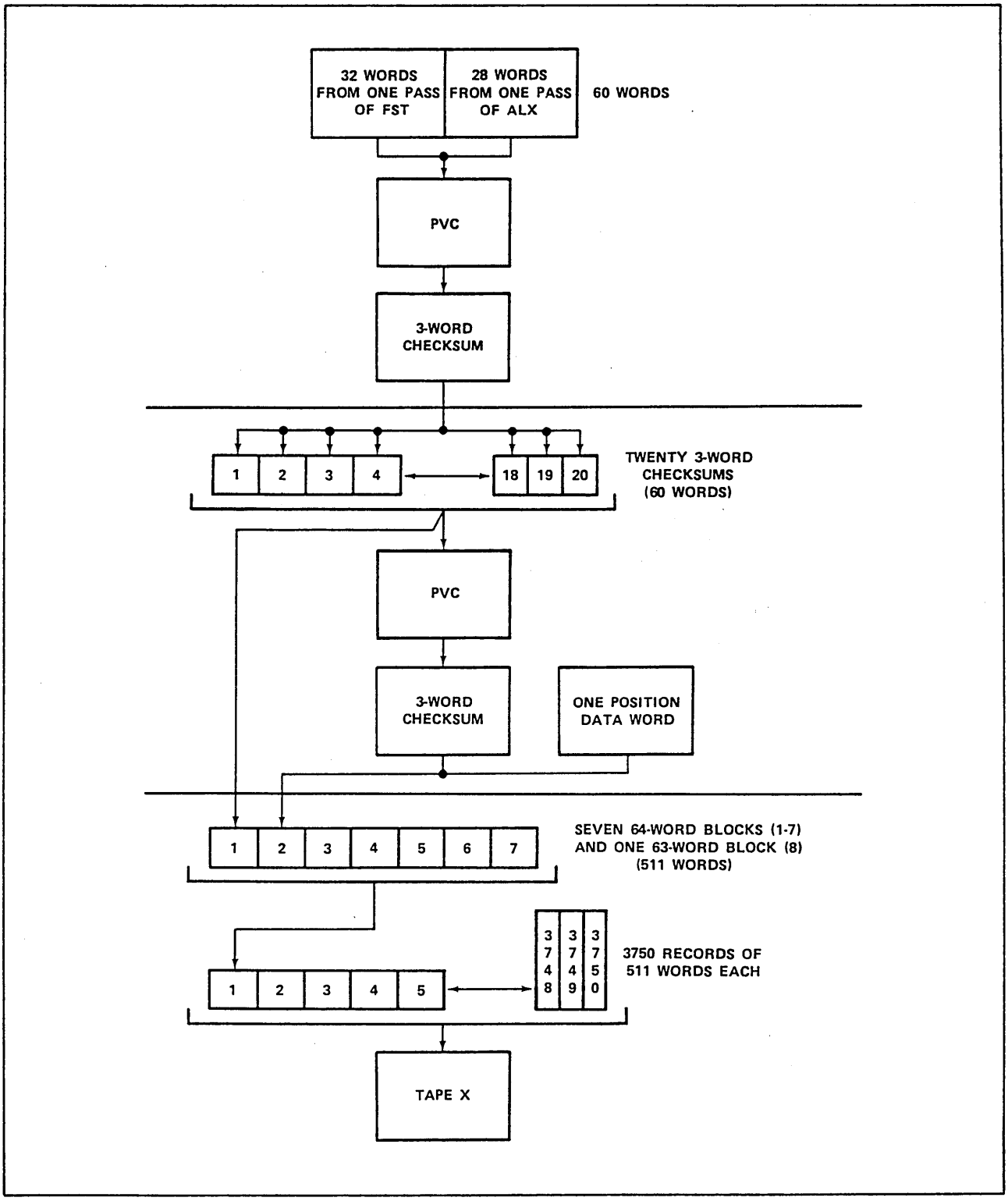

Figure 4-5. Generate Mode Flowchart

Copy Mode

The copy mode, shown in figure 4-6, verifies an input data path, and creates a copy of the master data base on the same or another device in the system. Normally, the copy created in this mode is verified on a subsequent copy or when in the regenerate mode.

A 511-word record is read from tape n, specified on an input data card to REGEN, as the one containing the master data base and copied to another tape nn. Prior to writing this record, a number of validation checks are made to ensure that the data has not been deteriorated by the hardware or software. On all reads of a data base, the following checks are made.

Parity error status check

If the system returns control to REGEN after detecting a parity error, a backspace reread is performed once. If the error persists, the job is aborted.

End-of-file status check

If an end-of-file status is detected, it indicates that the data base does not contain the expected number of records requested by the first input data card to REGEN. In this case, the job is aborted.

Record length check

The record length is checked. If it does not contain 511 words, a backspace-reread is performed. If the error persists, the job is aborted.

Data check

Data is checked to ensure that it belongs to the current job, is in the correct record-number sequence, and that the blocks (or sectors, in the case of mass storage) are also in the correct sequence. This check is performed by validating the seven position data words within the record. Up to six retries can be made to correct a record that fails these tests.

An in-core retry of the test is performed to ensure that is is not a transient CPU error.

A backspace-reread to the same buffer is performed to determine if it is a transient I/O network error.

A backspace-reread to a second buffer in central memory is performed to determine if the error is central memory address sensitive. Buffers 1 and 2 first word addresses are staged one word off from each other so that a central memory bank failure brings the detected failure to a new address when switching from buffer 1 to buffer 2.

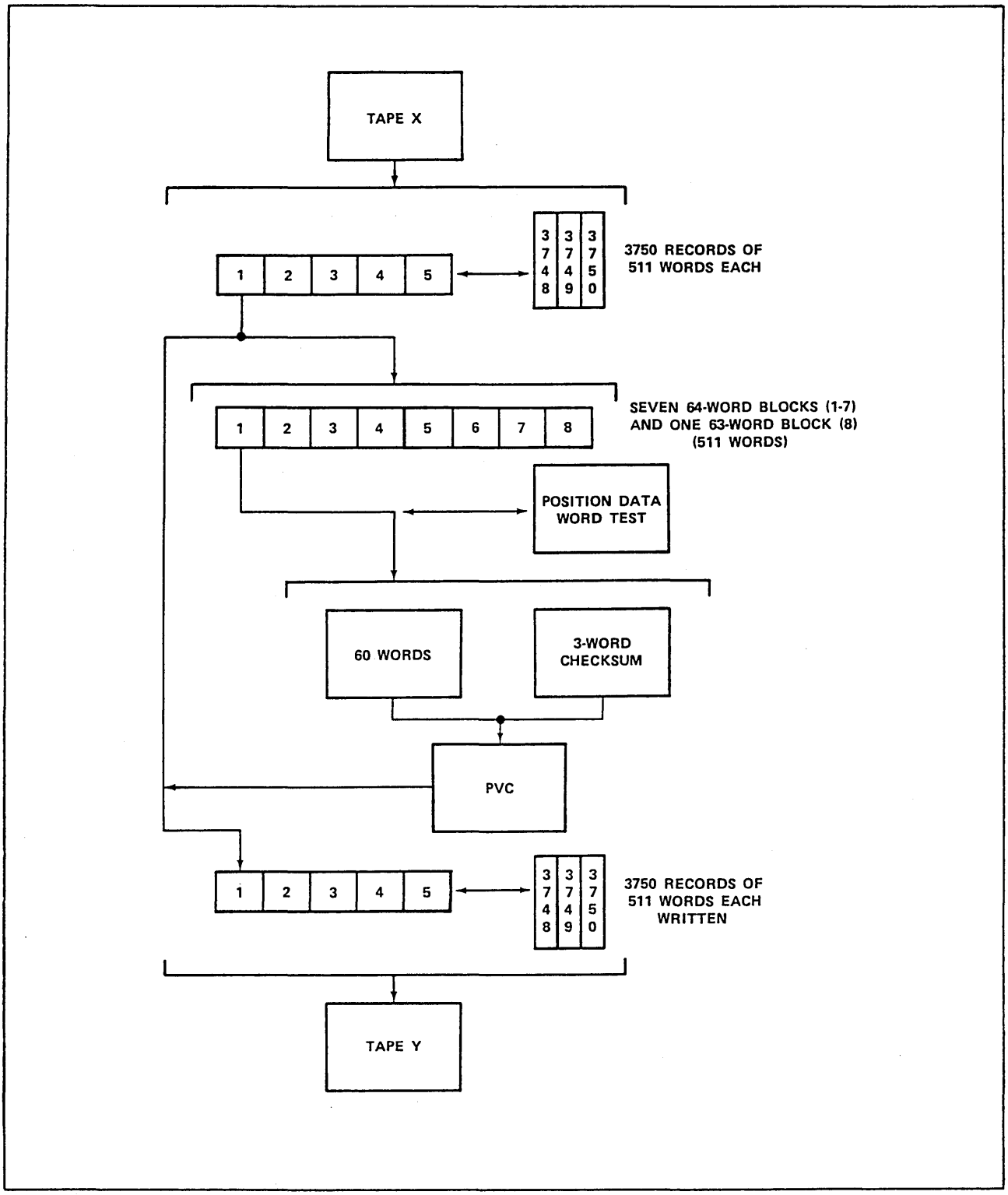

Figure 4-6. Copy Mode Flowchart

NOTE

The previous process could correct a position data error since it is only one word in length. However, a checksum data error could follow since it is possible to move data into the bad bank addresses. Once a switch is made from buffer 1 to buffer 2 on a record due to an error, the program stays in that buffer until the record is processed. The next record is processed out of buffer 1. Buffer 1 is always used in the generate mode. Buffer use in the copy or regenerate modes is dictated by error situations.

An in-core retry of buffer 2.

A backspace/reread to buffer 2.

Retries are attempted only until a recovery has been successfully completed or until all options have been tried without success, in which case the job is aborted. On each failure, the data in question is printed for customer engineering analysis, and an error log is written to a file called tape 99 for later and more detailed analysis during a run of the isolate mode. Data is also printed after a recovery to provide the customer engineer with correct data to compare with bad data previously written. If it is not possible to effect a recovery and all errors are identical, it can be assumed that the data being read was written incorrectly.

The checksums of all eight blocks are validated by subroutine PVC. Six retires are possible in case of error and are identical to those used on position data errors. In all cases, if a reread is performed to recover a specific block error, the reread could destory a previously validated block. Therefore, all position data and checksum words must be revalidated after recovery if a reread has been performed.

If all these checks are successful, new position data is inserted into the data read from tape x and the 511 words are written to tape xx. New position words must be added to ensure that the job name identifier supplied for this run on the first input data card to

REGEN is carried with the file. The process of reading, validating, and writing is repeated for the number of records specified on the first input data card to REGEN (number of records to process).

Regenerate Mode

The regenerate mode, shown in figure 4-7, accomplishes two tasks.

- It verifies an input data path in the same manner as in the copy mode (parity, length, position, and checksums). The file number is specified on the input data cards to REGEN.

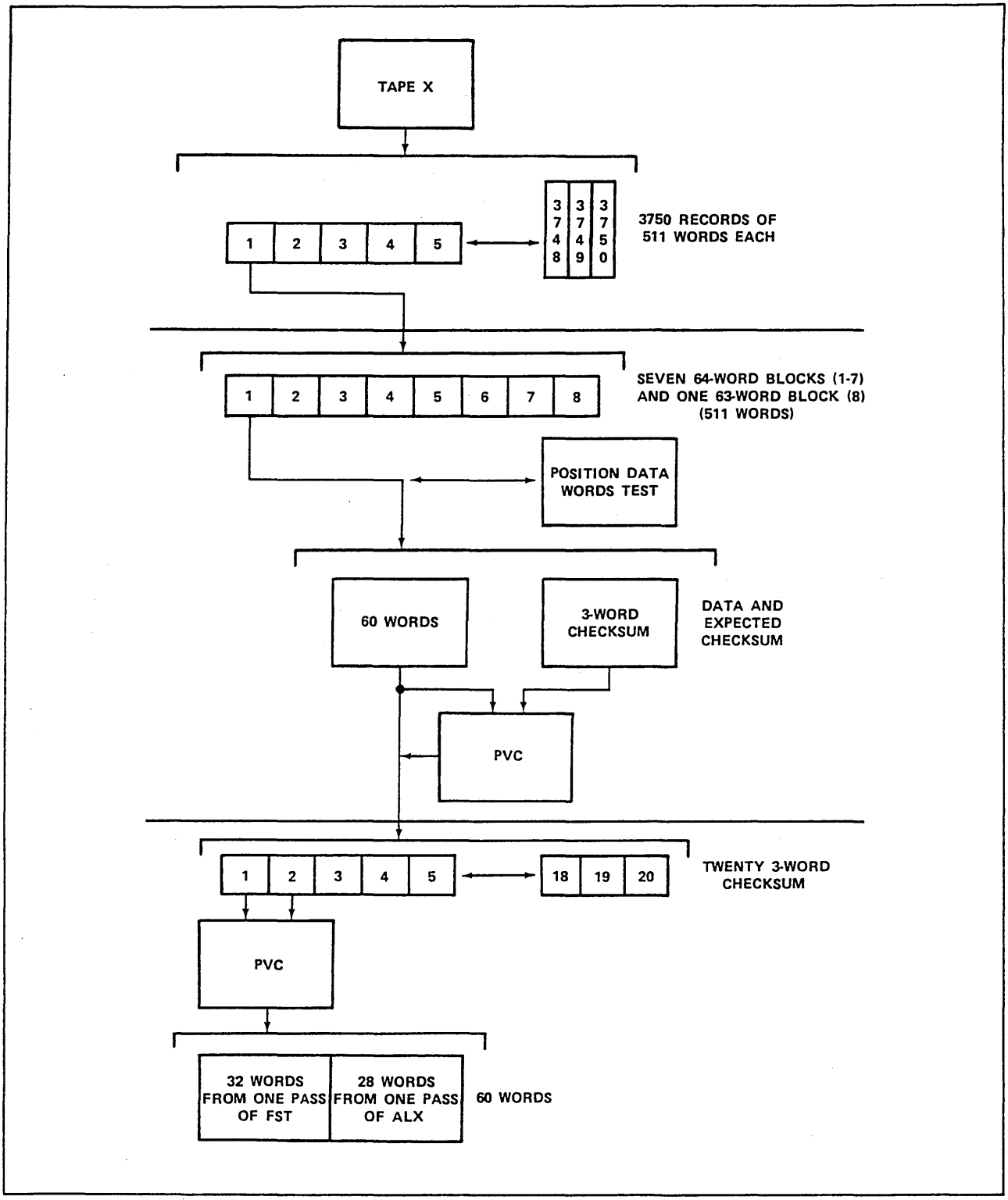

Figure 4-7. Regenerate Mode Flowchart

- It tests the central processor and central memory units by running FST and ALX, calling subroutine PVC to generate checksums of answers of each pass and comparing them to the checksums read across the input data paths.

After a record of 511 words has been read and validated, each block of this record is regarded as 20 three-word checksums that represent the checksums of 20 passes of FST/ALX. Therefore, 20 passes of FST/ALX are run before a block is used up. This process is repeated eight times to use up a total record representing 160 passes of FST/ALX per record. The process of reading a record, validating a record, and running FST/ALX/PVC to test the central processor and central memory is repeated the number of times specified by the number of records to process given on the first input data card to REGEN.

Return Mode

The RETURN command rewinds tape x and calls a COMPASS subroutine UNLOAD to perform a close/unload operation. This releases a physical tape unit for use by another job in the system or releases disk space for use by either REGEN or another job in the system.

Runpass Mode

Runpass is not a test. It is a maintenance aid designed to aid a customer engineer in determining more about a failure on a previous REGEN job where FST/ALX failed to re-create the expected checksums of a particular pass in the regenerate mode. When this type of error occurs, all that is known is the current data generated by FST/ALX, the current checksums of that data, and the expected checksums. There is no way of knowing the correct data that FST/ALX should have generated on this pass for checksumming.

RUNPASS provides this data by initializing FST/ALX to the pass requested on the input data card to REGEN. Following the pass, PVC is called to generate a three-word checksum of this data. The results are then printed.

CAUTION

If the same system is used to run this task, it could fail in the same way as the previous run of REGEN. If this happens, SMM off-line diagnostics should be used to verify system operation.

Isolate Mode

Isolate is another maintenance aid within REGEN that aids the customer engineer in determining more about the characteristics of a failure in a previous REGEN run. Its function is to process the tape 99 error log generated by a previous REGEN job in which errors were detected.

Isolate requires two tape files specified on an input data card. One file must contain a good copy of the master data base, and the other file must contain the error log from a previous REGEN run.

Isolate rewinds both tapes and searches the master data base for records that are on the error log. Once found, the two records are compared, and any noncompare words are printed. The raw error log, generated by a REGEN job, can have records out of sequential order relative to the master data base. Therefore, it is advisable to run the raw error log through a sort/merge phase before running the isolate mode. If SORTMERGE is not available on site, Isolate detects that the records are out of sequence. It then rewinds the master data base file and restarts the search. If there are many errors in the error log, this action could be time-consuming.

OPERATING PROCEDURES AND EXECUTION DESCRIPTION

Program REGEN is control card callable; the first card of the input deck must contain a unique job identifier consisting of alphabetic characters in columns 1 through 6. This card must also contain the number of records (in decimal) to be processed right-justified in columns 8 through 11.

The various REGEN modes of operation are selected by input data cards as shown in table 4-1. All cards begin in column 1.

Table 4-1. Input Control Cards (Sheet 1 of 2)

Control Card	Description
GENERATE xx	Select generate mode and generate a master data base on tape xx. xx is any decimal number representing tapes 1 through 32, xx begins in column 12, right justified.
COPY xx yy	Select copy mode and copy tape xx to tape yy. xx is the input file number (1 through 32) representing tapes 1 through 32, and yy is the output file number (1 through 32). xx begins in column 12 and yy begins in column 15, both right justified.
REGENERATE xx	Select regenerate mode and use the data base on tape xx. xx is the input file number representing tapes 1 through 32. xx begins in column 12, right justified.

Table 4-1. Input Control Cards (Sheet 2 of 2)

Control Card		Description
RETURN	xx	Select return mode to rewind, close, and unload tape xx. xx begins in column 12, right justified.
RUNPASS	xxxxxxx	Select runpass mode and run FST/ALX for pass requested by xxxxxxx. xxxxxxx is an octal pass number between 0 and 2223677. xxxxxxx begins in column 12, right justified.
ISOLATE	xx yy	Select isolate mode to obtain a good copy of the master data base at file number xx. xx is a number between 1 and 32. yy is the tape number between 1 and 32 containing the error log of a previous REGEN run. xx begins in column 12, and yy begins in column 15; both are right justified.

SENSE SWITCHES

REGEN uses three sense switches to control job execution and error reporting.

Sense switch 1, when set, causes REGEN to terminate at the end of the current mode regardless of the modes remaining in the input file.

Sense switch 2, when set, causes REGEN to pause after each of the following error situations.

Pause 1	Read parity error.
Pause 2	Read length error.
Pause 3	Position data error.
Pause 4	Checksum data error on input data.
Pause 5	First try checksum error on FST/ALX data during regenerate mode.
Pause 6	The in-core retry of the checksum operation on the FST/ALX data.

The pause message number and dayfile message issued prior to the pause allows the user to determine the nature of the pause. An operator GO command is required to continue processing.

Sense switch 3 is used to control the amount of data that is printed on each failure analyzed during the isolate mode. If this switch is off, only the first 10 words in error within each record are printed. If the switch is on, all noncompare words are printed and could result in over 511 lines per record in error.

SAMPLE JOBS

The following sample jobs illustrate the use of program REGEN and how the various modes are called.

Example 1: This job generates a master data base of 3750 records to be used as the site master copy. Use the job name of MASTER.

```
JOBCARD
ACCOUNT and RESOURC cards as required
REQUEST,TAPE1.  Magnetic tape request card for
MODE,1.         data base.
REGEN.
EXIT.
DMP,10000.      In case of an FST/ALX error.
7/8/9
MASTER 3750
GENERATE   01
6/7/8/9
```

Example 2: This job copies tape 1 to tape 2 to tape 3 to tape 4 and regenerates data on tape 4. Process 500 records with job name of JOB2. The master data base is on tape 1; therefore, a REQUEST card must be be included that properly identifies it. The files for tapes 2, 3, and 4 can reside anywhere on tape or disk as the user desires.

```
JOBCARD
ACCOUNT, and RESOURC cards as required.
REQUEST cards to specify the path to tapes 1
        through 4.
REQUEST,TAPE99.  Magnetic tape request card for
                 error log.
MODE,1.
REGEN.
EXIT.
DMP,10000.       In case of an FST/ALX error.
7/8/9
JOB2   0500
COPY        01 02
RETURN      01
COPY        02 03
COPY        03 04
REGENERATE 04
6/7/8/9
```

Example 3: This job executes a runpass of FST/ALX to pass number 0000123.

```
JOBCARD
ACCOUNT and RESOURC cards as required.
MODE,1.
REGEN.
EXIT.
DMP,10000.        In case of an FST/ALX error.
7/8/9
JOB3
RUNPASS     0000123
6/7/8/9
```

Example 4: This job executes an isolate mode against an error log from a previous REGEN job. Sort/Merge is in the system, so sort the error log from tape 99 to tape 2 before presenting it to REGEN. The master data base is on tape 1.

```
JOBCARD
ACCOUNT and RESOURC cards as required.
REQUEST,TAPE99,VSN=XXX. REGEN  error log.
REWIND,TAPE2.
File cards as required.  Maximum character length
  equals 5180.
SORTMRG.
REQUEST,TAPE1,VSN=YYY.  Master data base.

REGEN.
7/8/9
SORT
BYTESIZE,60
FILE,INPUT=TAPE99(CU),OUTPUT=TAPE2(R)
FIELD,RECORD(1,1),BLOCK(2,1)
KEY,RECORD(A,INTBCD),BLOCK(A,INTBCD)
OPTIONS,RETAIN
END
7/8/9
JOB4
ISOLATE     01 02
6/7/8/9
```

Example 5: This job executes an isolate mode against an error log from a previous REGEN job. Sort/Merge is not in the system. The error log is on tape 99, and the master data base is on tape 1.

```
JOBCARD
ACCOUNT and RESOURC cards as required.
REQUEST,TAPE99,VSN=XX. REGEN error log.
REQUEST,TAPE1,VSN=YY. REGEN master data base.
REGEN.
7/8/9
JOB5
ISOLATE     01 99
6/7/8/9
```

MESSAGES

Two types of messages are issued by REGEN, dayfile messages and output file messages. Dayfile messages provide the user with an identification of the current operation in progress. Output file messages provide a crosscheck of the dayfile messages, and all are preceded by the date and time of day.

Dayfile Messages

GENERATE TAPE XX

A master data base is being generated on tape xx.

COPY TAPE XX TO YY

The data on tape xx is being copied on tape yy.

REGENERATE TAPE XX

The master data base on tape xx is being regenerated and revalidated.

RETURN TAPE XX

A close/unload operation is being performed on tape xx.

RUNPASS XXXXXXX

A pass of FST/ALX designated by xxxxxxx is being run.

RUNNING ISOLATE MODE

Program REGEN is currently being run in the isolate mode.

Output File Messages

START nnnnnn NUMBER OF RECORDS
SELECTED = YYYY

The job named nnnnnn has been started and the number of records selected is yyyy.

START GENERATION OF TAPE XX

The generation of a master data base on tape xx has begun.

WALL CLOCK GENERATE TIME = HH.MM.SS
(YYYY.YY CPU SECONDS USED)

The total time taken to run the generate mode is given in hours, minutes, and seconds. The total CPU time in seconds is also given.

START COPY OF TAPE XX TO TAPE YY

Copying the data from tape xx to tape yy has begun.

WALL CLOCK COPY TIME = HH.MM.SS
(YYYY.YY CPU SECONDS USED)

The total time taken to run the copy mode is given in hours, minutes, and seconds. The total CPU time in seconds is also given.

START REGENERATION OF TAPE XX

The regeneration of the master data base on tape xx has begun.

WALL CLOCK REGENERATE TIME = HH.MM.SS
(YYYY.YY CPU SECONDS USED)

The total time taken to run the regenerate mode is given in hours, minutes, and seconds. The total CPU time in seconds is also given.

RETURNING TAPE XX

The COMPASS subroutine UNLOAD has been called and is performing a close/unload operation on tape xx.

WALL CLOCK RETURN TIME = HH.MM.SS
(YYYY.YY CPU SECONDS USED

The total time taken to run the return mode is given in hours, minutes, and seconds. The total CPU time in seconds is also given.

START ISOLATE MODE

The processing of data from the error log generated on tape 99 has begun.

WALL CLOCK ISOLATE TIME = HH.MM.SS
YY.YY CPU SECONDS USED)

The total time taken to run the isolate mode is given in hours, minutes, and seconds. The total CPU time in seconds is also given.

END OF RUN

This message signifies the end of the current run of program REGEN. The asterisks have no significance.

```
TOTAL WALL CLOCK RUN TIME = HH.MM.SS
(YYYYYY.YY CPU SECONDS USED
```

The total time taken for a complete run of REGEN is given in hours, minutes, and seconds. The total CPU time in seconds is also given.

```
***TOTAL BITS FOR RUN = XXXXXXXXXXXX
```

This is a count of the total number of bits, ones, and zeros generated and used by all modes during the previous run of REGEN.

```
TOTAL ERRORS SAVED ON TAPE99=XXXXX
```

The total number of errors entered into the error log is generated on tape 99.

The following output file messages are generated during the isolate mode of a REGEN run.

```
ANALYZING ERROR ON RECORD A BLOCK B
TAPE C DUE TO D ERROR

(BFRFWA=EEEEE)     DATE=F      TIME=G
WORD          EXPECTED     ACTUAL     DIFFERENCE

(Words that failed to compare)
```

A record containing errors and entered into the error log on tape 99 is being compared to a corresponding record in the master data base.

A	Record number of 1 through 3750, decimal.
B	Block number of 1 through 8.
C	Tape number of 1 through 32.
D	Error type (parity, length, position, checksum).
E	First word address of the buffer that failed.
F	Date error occurred.
G	Time error occurred.
WORD	Word number of 000 through 776 octal in the record that failed to compare with the master data base. To find the absolute memory address of a failure, add WORD+E+RA.
EXPECTED	Data from the master data base.
ACTUAL	Data from the error log.
DIFFERENCE	Logical difference between expected and actual.

RECORD XXXX HAD YYYY NON-COMPARE WORDS

XXXX 1 through 3750 decimal.

YYYY 0 through 511 decimal.

NORMAL TERMINATION OF ISOLATE DUE TO EOF ON ERROR-LOG

An end-of-file mark was detected on tape 99 (the error log) and the isolate mode was terminated.

ERROR CONDITIONS

If FST detects a failure, as with ALX it can generate a mode 1, mode 2, or mode 4 error. Since FST must exit to address 377776 on a mode 1 error only, a control card must be in the job deck that selects mode 1. Exit and dump control cards should be used after the call to REGEN to provide a memory dump if FST or ALX exits on error detection.

If ALX detects an error, the same procedure as in FST is followed except that ALX jumps to address 377777.

A checksum error can occur on data read from tape x while running in the regenerate mode. Subroutine PVC reports the error and has the ability to isolate a single-bit failure to the word and bit that failed. Multiple-bit failures are detected but cannot be isolated closer than the 63-word block in which the failures occurred and are dumped for analysis.

The following information is printed when an error occurs.

- File name.
- Buffer in use.
- Try/retry condition.
- Record number.
- Block number.
- Address containing the bit that failed.
- X, Y, and Z checksum vectors generated.
- 60 words of data in the block.
- Three-word checksums from tape for this block.
- Position data word for this block.

The format of address and bit data in 6 is:

xxxxxxxxyyzzzzzzzzzz

x...x Address containing the failing bit. If the failure is a multiple-bit failure, x...x is the fwa of the block containing the failures.

yy Bit that failed. If yy is equal to 00, a multiple-bit failure occurred.

z...z Equal to 0 if the failure is a single-bit failure. It is equal to -0 if the failure is a multiple-bit failure.

I/O errors are reported and retries are executed on read length and read parity errors. The following errors are fatal and the job is aborted.

- Write parity error on buffer out.
- EOF status on buffer out.
- EOF status on buffer in.
- Uncorrectable parity error on buffer in.
- Uncorrectable length error on buffer in.

Position data errors can occur on data read from tape x. The following data is reported to the output file.

- File name.
- Buffer in use.
- Try/retry condition.
- Record number.
- Block number.
- Expected and actual data.

ERROR MESSAGES

All error messages written to the output file are preceded by the date and time. The messages are self-explanatory.

Table 4-2 lists each error situation, the type of message issued, the contents of each message, and the action taken by REGEN after the message is posted. The numbers shown in columns 2, 3, and 4 are translated in table 4-3.

Table 4-2. Error Messages (Sheet 1 of 2)

Problem	D File Message	O/P Message	Error Log	Action
Read parity error	1	2	8	Backspace/reread once.
Read parity error recovery		2		Continue.
Uncorrectable read parity error		2		Abort job.
Read length error	1	2		Backspace/reread once.
Read length recovery		3		Continue.
Uncorrectable read length error		3		Abort job.
EOF on buffer in		2		Abort job.
Write parity error		2		Abort job.
EOF on write		2		Abort job.
Position data error	1	4	8	Up to six retries.
Position data recovery		4		Continue (may re-check positions).
Uncorrectable position error		4		Abort job.
Data checksum error	1	5	8	Up to six retries.
Data checksum recovery		5		Continue (may recheck positions and all checksums).
Uncorrectable data checksum error		5		Abort job.
FST/ALX checksum error		6		Retry once; rerun FST/ALX once.
FST/ALX checksum recovery		6		Continue.

Table 4-2. Error Messages (Sheet 2 of 2)

Problem	D File Message	O/P Message	Error Log	Action
Uncorrectable FST/ALX checksum error		6		Abort job.
FST failure				Mode 1 at 377776.
ALX failure				Mode 1 at 377777.
Illegal runpass number		7		Abort runpass.

Table 4-3. Key to Error Messages

Message Number	Message Contents
1	Time, retry count.
2	Date, time, record number, file name.
3	Date, time, record number, file name, length 1, length 2.
4	Date, time, file name, buffer, try condition, record number, block number, expected data, actual data.
5	Date, time, file name, buffer, try condition, record number, block number, address data, XYZ vectors, 60 words of data, expected three-word checksum, position data (blocks 1 through 7 only).
6	Date, time, pass count, address data, XYZ vectors, 60 words of data, expected checksum.
7	Date, time, pass count.
8	Record number, block number, file name, failure code, date, time, FWA of buffer in use, 511 words of data.

ERROR LOG

In addition to the error messages and data printed via the output file, program REGEN writes a record to tape 99 on any of the following errors.

- Read parity error (if REGEN obtains control from the system).
- Read length error.
- Position data error.
- Checksum error on input data.

The information recorded on tape 99 for each error consists of:

- Record number that failed (1 through 3750 decimal).
- Block number (1 through 8 if isolated, 0 if not).
- File number (1 through 32 representing tapes 1 through 32).
- Error code (1 is parity, 2 is length, 3 is position, 4 is checksum).
- Date of failure.
- Time of failure.

MICROCODE VERIFICATION PROCESS 5

The verification of microcode is a process that compares a suspected bad image of control store to a known good image. This is achieved by comparing the control store dump record on an Express Deadstart Dump (EDD) tape to the binary microde load module [contained on the CYBER Initialization Package (CIP) tape], used during system installation.

When an undefined system problem occurs, a system EDD is normally performed by the operator. A dump of control store (microcode memory) is contained on the EDD tape. Program MCDUMP reads the control store dump from the EDD tape and converts it to listable (readable) hexadecimal. MCDUMP is then used to convert the microcode load module from binary to hexadecimal. Both list files are then used as input to program MCCOMP. If differences are detected, the good and bad control store addresses and data are written to the output file.

Another source of a good control store image would be an EDD tape created shortly after system initialization.

CML installation automatically installs MCCOMP and MCDUMP.

MCDUMP UTILITY

The program call format is:

```
MCDUMP(p1,p2,p3)
```

Where p1, p2, or p3 can be any of the following in any order:

I=lfn1 From one to seven alphanumeric characters (first character must be an alpha) naming the local file to be dumped (default is TAPE1).

Input data can be from:

1. CTI/HIVS microcode record.
2. EDD tape file:

MCDUMP determines which one it is.

L=lfn2 From one to seven alphanumeric characters (first character must be an alpha) naming the local file to which the output is written (default is OUTPUT). If lfn2 is not a local file, MCDUMP creates it. It does not rewind lfn2 either before or after the dump.

NR Do not rewind lfn1 before the dump (default is to rewind lfn1).

The MCDUMP utility formats and lists CYBER 170/180 Models 810, 815, 825, 830, 835, 845, and 855 microcode from an input file. The input file can exist in one of two formats.

MCDUMP FORMATS

1. The CIP tape contains the microcode. The following examples generate hardcopy listings of the microcode contained on the CIP tape.

   EXAMPLE: NOS CIP microcode dump

```
REQUEST(CIP,NT,PE,F=SI,PO=R,LB=KU,VSN=CIP)O=R,LB=KU,VSN=CIP)
COPYBR,CIP,DISCARD,xx.   (where xx is number of decimal records to skip)
COPYBR,CIP,MCODE.
ATTACH,MCDUMP.
MCDUMP(I=MCODE,L=OUT)
ROUTE(OUT,DC=PR)
```

   EXAMPLE: NOS/BE CIP microcode dump

```
REQUEST(CIP,NT,PE,VSN=CIP)
COPYBR,CIP,DISCARD,xx.   (where xx is number of decimal records to skip)
COPYBR,CIP,MCCODE.
ATTACH,MCDUMP,ID=CDCCE.
MCDUMP(I=MCCODE,L=OUT)
ROUTE(OUT,DC=PR)
```

2. Express Deadstart Dump (EDD) format.

   When the microcode you want to dump is on an EDD file, it is not necessary to extract the control store dump from the rest of the data. MCDUMP does this for you. The file is searched and the control store data (PCSx) is formatted and dumped automatically.

   EXAMPLE: NOS EDD microcode dump

```
REQUEST(EDDTAPE,NT,PE,F=S,PO=R,LB=KU,VSN=EDD)
ATTACH,MCDUMP.
MCDUMP(I=EDDTAPE,L=OUT)
ROUTE(OUT,DC=PR)
```

   EXAMPLE: NOS/BE EDD microcode dump

```
REQUEST(EDDTAPE,NT,PE,S,VSN=EDD)
ATTACH,MCDUMP,IDS=CDCCE.
MCDUMP(I=EDDTAPE,L=OUT)
ROUTE(OUT,DC=PR)
```

MCCOMP UTILITY

The MCCOMP utility provides one specific function. It compares two list files that were previously generated by program MCDUMP. MCDUMP is run twice; once to convert a known good microcode file to hexadecimal, and a second time to convert a suspected microcode file to hexadecimal. These two output files are then compared by MCCOMP and all miscompares are written to a list file.

The program call format is:

```
MCCOMP,p1,p2,p3,p4.
```

Where p1, p2, p3, or p4 can be any of the following in any order:

I1=lfn1	From one to seven alphanumeric characters (first character must be an alpha) naming either one of the list files created by MCDUMP (default is TAPE1).
I2=lfn2	From one to seven alphanumeric characters (first character must be an alpha) naming the other list file created by MCDUMP (default is TAPE2).
L=LFN3	From one to seven alphanumeric characters (first character must be an alpha) naming the local file to which the output is written (default is OUTPUT). This file shows differences between LFN1 and LFN2. If LFN3 is not a local file, MCCOMP creates it. The file is not rewound before or after the compare.
NR	Do not rewind files LFN1 and LFN2 before compare (default is to rewind both LFN1 and LFN2).

MCCOMP EXAMPLES

A typical sequence for NOS is:

```
REQUEST,TAPE,NT,PE,PO=R,F=S,LB=KU,VSN=GOOD.   Assign good EDD tape.
ATTACH,MCDUMP.                                Attach MCDUMP program.
MCDUMP,I=TAPE,L=GOODMC.                       Create good MC list file.
RETURN,TAPE.
REQUEST,TAPE,NT,PE,PO=R,F=S,LB=KU,VSN=BAD.    Assign problem EDD tape.
MCDUMP,I=TAPE,L=BADMC.                        Create suspect MC list file.
ATTACH,MCCOMP.                                Attach MCCOMP program.
MCCOMPP,I1=GOODMC,I2=BADMC,L=OUT.             Compare/Write to out.
ROUTE,OUT,DC=LP.                              List miscompares.
```

A typical sequence for NOS/BE is:

```
REQUEST,TAPE,NT,PE,S,VSN=GOOD.                Assign good EDD tape.
ATTACH,MCDUMP,ID=CDCCE.                       Attach MCDUMP program.
MCDUMP,I=TAPE,L=GOODMC.                       Create good MC list file.
RETURN,TAPE.
REQUEST,TAPE,NT,PE,S,VSN=BAD.                 Assign problem EDD tape.
MCDUMP,I=TAPE,L=BADMC.                        Create suspect MC list file.
ATTACH,MCCOMP,ID=CDCCE.                       Attach MCCOMP program.
MCCOMP,I1=GOODMC,I2=BADMC,L=OUT.              Compare/Write to out.
ROUTE,OUT,DC=LP.                              List miscompare.
```

VERIFICATION PROCESS

A typical verification process follows:

1. Convert good control store binary.

   NOS CIP tape

   ```
   REQUEST,CIP,NT,PE,F=SI,PO=R,VSN=CIP,LB=KU.
   COPYBR,CIP,DISCARD,xx.   (where xx is number of decimal records to skip)
   COPYBR,CIP,GOOD.
   RETURN,CIP.
   ATTACH,MCDUMP.
   MCDUMP,I=GOOD,L=GOODMC.
   ```

   NOS/BE CIP tape

   ```
   REQUEST,CIP,NT,PE,VSN=CIP.
   COPYBR,CIP,DISCARD,xx.   (where xx is number of decimal records to skip)
   COPYBR,CIP,GOOD.
   RETURN,CIP.
   ATTACH,MCDUMP,ID=CDCCE.
   MCDUMP,I=GOOD,L=GOODMC.
   ```

2. Convert suspected bad control store dump.

   NOS EDD dump

   ```
   REQUEST, EDDBAD,NT,PE,F=S,PO=R,LB=KU,VSN=EDDBAD.
   ATTACH,MCDUMP.
   MCDUMP,I=EDDBAD,L=BADMC.
   ```

   NOS/BE EDD dump

   ```
   REQUEST,EDDBAD,NT,PE,S,VSN=EDDBAD.
   ATTACH,MCDUMP,ID=CDCCE.
   MCDUMP,I=EDDBAD,L=BADMC.
   ```

3. Compare the two converted files and list all the differences.

   NOS compare list

   ```
   ATTACH,MCCOMP.
   MCCOMP,I1=GOODMC,I2=BADMC,L=OUT.
   ROUTE,OUT,DC=PR.
   ```

   NOS/BE compare list

   ```
   ATTACH,MCCOMP.
   MCCOMP,I1=GOODMC,I2=BADMC,L=OUT.
   ROUTE,OUT,DC=PR.
   ```

   NOTE

   If there are no miscompares, MCCOMP does not generate an output file. The message, COMPARE GOOD, is written to the dayfile.

SKEDULR 6

SKEDULR is an application program that assists in the implementation of concurrent maintenance, remote technical assistance, and the on-call maintenance strategy for CYBER systems.

SKEDULR automatically submits NOS and NOS/BE jobs to the system input queue. This permits maintenance jobs (MALET, HPA, REGEN, TIO, CEAIDS) to be initiated without the need for manual intervention. SKEDULR allows maintenance jobs to be executed at times convenient to the customer, and the customer engineer need not be present. This concept is helpful at installations maintained from a service center in an on-call maintenance environment. Proper use of SKEDULR also supports the remote technical assistance center concept.

Three elements exist in the scheduling function as follows:

- The scheduler (SKEDULR).
- The jobs to be executed.
- The schedule (list of jobs to be scheduled).

Normally, all three elements reside on system permanent files. SKEDULR can be executed as a job from a user file and does not have to be installed into the customer's system or reside on the deadstart tape.

The next task is to generate the jobs to be executed. The jobs vary with each installation but typically include daily runs of HPA and REGEN, weekly runs of diagnostics on each peripheral, weekly runs of HPA for tape media reports, CAMS runs, and so on. Any job currently executed manually is a candidate for this list. These jobs must be saved as permanent files for later execution (indirect access permanent files for NOS).

Once the jobs are created, the schedule of when the jobs are to be run must be generated. This is accomplished by creating a file using the operating system EDIT or EDITOR feature, or keypunching the file and saving it as a permanent file (indirect access file for NOS). A typical entry in this file is as follows:

```
RUN D44020A MONDAY AT 2300
```

This entry directs SKEDULR to submit file D44020A to the system input queue every Monday night at 2300 hours (11 p.m.). The syntax for the RUN directives provides the flexibility to submit any number of jobs at a variety of intervals, from several times a day to once a year.

INSTALLING SKEDULR

(Refer to section 1, under CML Installation, for installation procedures.)

CREATING JOBS TO EXECUTE

EDIT or EDITOR can be used to create the jobs to be executed, or the jobs can be keypunched. In either case, the jobs must be saved as permanent files for later execution (indirect access files under NOS). The content of each job depends upon the task that must be accomplished. There are two basic formats for jobs that are to be submitted.

- If the first line of the file contains /JOB in columns one through four, SKEDULR creates a new file to be submitted. The /JOB statement is eliminated and any lines with /EOR or /EOF in columns one through four are replaced by the end-of-record or end-of-file, respectively. This allows the user to create a multirecord file with editors that are not capable of identifying 7/8/9 cards (EOR) or 6/7/8/9 cards (EOF). A typical example is as follows (NOS format shown):

```
/JOB
D440020A,P4,T200.
User/Account cards as required.
ATTACH,RFILE=D44.
Request engineering mode and EST20 access.
MALET.
REPLACE,OUTPUT=044020A.
EXIT.
REPLACE,OUTPUT=044020A.
/EOR
ASSIGN,EST=20,CH=31,AL=3
CLRSW SE
RUN D44(01,02,03,04,05,06,07,08,09,10,11,12,13,14,15,16)
ASSIGN,EST=20,CH=30,AL=3
RUN D44(01,02,03,04,05,06,07,08,09,10,11,12,13,14,15,16)
/EOF
```

- If the first line of the file does not contain /JOB, the file is submitted without any changes. A typical example of this format is as follows:

```
D440020A,P4,T200.
User/Account cards as required.
ATTACH,RFILE=D44.
Request engineering mode and EST20 access.
MALET.
REPLACE,OUTPUT=044020A.
EXIT.
REPLACE,OUTPUT=044020A.
7/8/9    (EOR)
ASSIGN,EST=20,CH=31,AL=3
CLRSW SE
RUN D44(01,02,03,04,05,06,07,08,09,10,11,12,13,14,15,16)
ASSIGN,EST=20,CH=30,AL=3
RUN D44(01,02,03,04,05,06,07,08,09,10,11,12,13,14,15,16)
6/7/8/9      (EOF)
```

In the examples, the diagnostic output is saved as a permanent file in addition to being printed at the local site (REPLACE,OUTPUT=044020A). This technique provides a hardcopy for on-site customer engineers and the permanent file copy can be accessed from a remote service center or technical assistance center.

CREATING THE JOB SCHEDULE

Once the job decks are created, a schedule must be generated on permanent file JOBS (JOBSxx if NOS multimainframe). This file must be an indirect access file for NOS.

Jobs to be executed are defined with RUN directives that are in 80-column format, starting in column one as follows:

RUN lfn when AT time,time,time,...

lfn — Permanent file name of the file to be submitted (one to seven characters). This must be an indirect access file for NOS.

when — Day the job is to be submitted. The following keywords are allowed:

DAILY	Seven days a week
WEEKDAYS	Monday through Friday (five days)
WEEKENDS	Saturday and Sunday
SUNDAY through SATURDAY	On one specific day
MONTHLY (x)	Monthly on day x (1 through 31)
MONTYLY (y z)	Monthly on day y z y FIRST through FIFTH z SUNDAY through SATURDAY (for example, MONTHLY (SECOND MONDAY))
MONTHLY (LAST DAY)	Monthly on last day
JANUARY through DECEMBER (x)	Specific month, day x (1 through 31)
JANUARY through DECEMBER (y z)	Specific month, day y z y FIRST through FIFTH z SUNDAY through SATURDAY {for example, JULY (FOURTH FRIDAY)}

time — Time of day the job is to be submitted (0000 through 2359). This is decimal military time defined in four digits (leading zeros required) with no separators. If the job is to be submitted more than once per day, multiple times can be specified.

Single blanks and/or commas can be used as delimiters between each word or time entry; however, two blanks terminate the RUN directive. Remaining columns, if any, may be used for documentation.

Examples:

```
RUN HPASUM DAILY AT 0830
RUN REGEN01 DAILY AT 0800,1000,1200,1400,1600,1800,2000
RUN REGEN02 WEEKDAYS AT 2300
RUN REGEN03 WEEKENDS AT 1900
RUN D44020A MONDAY AT 0400      *RUN D44 ON EST 20
RUN HPAMON MONTHLY (15) AT 1500
RUN FTP015A MONTHLY (FIRST MONDAY) AT 1200
RUN HPATERM MONTHLY (LAST DAY) AT 2300
RUN CAMSRUN JULY (15) AT 0800
RUN CAMS JULY (FIRST WEDNESDAY) AT 0900
```

EXECUTING SKEDULR — NOS

To start the scheduling function under the NOS operating system, the following job must be executed:

```
SKED,P4,TO.
User and charge cards as required.
GET,SKEDULR.
SKEDULR.
EXIT.
REQUEST,HELP.   SKEDULR ABORTED.
6/7/8/9     (EOF)
```

SKEDULR puts a date and time stamp on the time file (TFILE), rolls out for 30 minutes, and determines if the time stamp has been changed on TFILE. If it has, another copy of SKEDULR is running and this copy terminates. If not, a fresh copy of the JOBS file is obtained and processed. Jobs that need to be scheduled are submitted to the input queue and SKEDULR rolls out for 15 minutes or until it is time to schedule the next job, whichever is less.

The process of obtaining a fresh copy of the JOBS file at least every 15 minutes allows the schedule to be changed if it is necessary (by modifying JOBS or JOBSxx) and the new schedule is put into effect within 15 minutes of the changes.

Due to the protection feature (ensuring two or more copies of SKEDULR are never executing at the same time), the scheduling function job can be executed after every system deadstart, if desired. It must be executed again if SKEDULR aborts, after the cause of the abort is corrected. A message is issued to the operator when this occurs. SKEDULR aborts if it submits a job to the input queue that contains an invalid user card.

When SKEDULR submits jobs to the input queue, the job being submitted is given an origin ID that is the same as SKEDULR. If SKEDULR was initiated using the scheduling function job, all jobs submitted would have an origin of batch, with no special privileges. However, if SKEDULR is initiated from the system console, using DIS, it would have system origin privileges. This is a very powerful capability that must not be misused, but it can be useful since this mode allows the running of NORM/GETLOG/HPA jobs without the system being in debug mode.

EXECUTING SKEDULR — NOS/BE

SKEDULR is executed as normal job under the control of the automatic program recall (APR) feature of NOS/BE so that it is executed every 15 minutes. This ensures that jobs are scheduled within 15 minutes of the requested time and a new or modified JOBS file can be put into effect within 15 minutes of being created. The following job should be executed:

```
SKED,TO.  (accounting if necessary)
ACCOUNT,--------(if necessary).
APR(1,001702)
ATTACH,JOBS,ID=(user ID).
ATTACH,SKEDULR,ID=(user ID).
SKEDULR.
EXIT.
REQUEST,HELP.  SKEDULR ABORTED.
6/7/8/9    (EOF)
```

In the APR call, the parameters (1,001702) request APR to put this job into the APR table for recall in 15 minutes. The last two digits of the request indicate the job number APR assigns to this job (02).

SKEDULR uses the time file (TFILE) to determine when it was last executed. This allows it to schedule jobs that fall into the current time frame and when SKEDULR was last run. The entire JOBS file is processed, SKEDULR exits and is placed into the APR table, and executes again 15 minutes later.

EXECUTIVE RUNNING AND ERROR MESSAGES A

The following are the running and error messages issued by MALET.

CVT ERROR VALUE.

Error in the process of converting value to binary.

DAYFILE LIMIT REACHED DAYFILE LOG TURNED OFF

One-hundred (decimal) dayfile messages have been logged. No more are allowed until the dayfile log switch is turned on.

DROPPED BY USER.

DROP directive encountered.

END OF TEST SERIES.

Test series being run has encountered an END statement.

EOF/EOI INPUT END MALET.

End-of-file or end-of-information encountered on input and MALET K/L display is not turned on.

ERROR EF = xx P = yyyy.

Error occurred where xx = error flag and yyyy = P address. The error flag values are as follows:

NOS Version 1†

EF = 01	Arithmetic error.
= 02	CPU encountered program stop.
= 03	PP abort.
= 04	CPU abort.
= 05	PP call error.
= 06	Time limit.
= 07	File limit.
= 10B	Track limit.
= 11B	SRU limit.
= 12B	Forced error.
= 13B	Operator dropped job.
= 14B	Special error flag or job rerun.
= 15B	Operator killed job.
= 16B	Subsystem abort.
= 17B	ECS parity error.
= 20B	CPU or CM parity error.
= 21B	System abort.
= 22B	Override of error condition.

†For further information about error messages, refer to NOS Version 1 Reference Manual, Volume 2.

NOS Version 2, NOS/BE†

PP PROGRAM REQUESTED ABORT
PP CALL ERROR
OPERATOR DROP
OPERATOR KILL
OPERATOR RERUN
ECS PARITY ERROR
AUTO-RECALL ERROR
HUNG IN AUTO-RECALL
PP PROGRAM NOT IN LIBRARY
TERMINAL INTERRUPT

FORMAT WAS OUT OF RANGE.

Format statement could not be found. Check module source for error.

HALT - STOPPED ON ERROR.

Module executed a HALT command. To resume execution a GO directive must be entered through K/L display, if running on console; or input, if running with a terminal.

ILLEGAL SEPARATOR RE-ENTER.

Separator was not recognized by MALET. Reenter with legal separator.

ILLEGAL WHEN MODULE NOT RUNNING.

Directive is illegal when module is not running. For example, STOP when module is not running.

ILLEGAL WHEN MODULE RUNNING.

Directive is illegal when module is running.

?

Please enter input to MALET on terminal.

K/L DISPLAY NOT ON - DROP FORCED.

K/L display flag not set. Drop is forced when assign with validation denied given.

LANGUAGE CHANGE - MODULE TERMINATED.

In response to a CALL directive, a module was called that is not written in the same language as the module that performed the call.

LINE LIMIT -- GO/DROP.

Line limit reached, user must give GO or DROP to resume with default line limit restored.

---

†For further information about error messages, refer to NOS Version 2 Reference Set, Volume 2, Guide to System Usage, or NOS/BE Version 1 Reference Manual.

LINE VALUE OUT OF RANGE.

Value too large. Reenter with a value that is in range.

MALET ABORTED.

MALET was aborted due to some unrecoverable error.

MODULE CHECKSUM ERR.

Checksum did not compare with checksum within binary.

MODULE DOES NOT SUPPORT DEVICE CODE.

Device codes stored in module do not include current one assigned.

MODULE FUNCTION OUT OF RANGE.

Internal error. Function being sent by PP is not recognized by the MALET executive.

MODULE INITILIZE ERROR MALET DROPPED.
MODULE INITILIZE ERROR — SEE DAYFILE.

MALET PP driver program MLD has found an error in the module binary. MALET is dropped if the K/L display is off.

MODULE LOCKED USE NEW NAME.

Module lock parameter was used when binary was made.

MODULE NAME MUST BE 5 CHARACTERS.

Module name must be five characters.

MODULE NOT CHECKPOINTED.

Module binary was not saved for restart. Must use RUN directive to resume execution.

MODULE NOT FOUND = name.

Module name not found on random file.

MODULE NUMBERS MUST BE 2 DIGITS.

Module numbers must be two digits on RUN directive.

MODULE WILL NOT RUN WITH ACCESS LEVEL .LT. 10_8.

Module is low level. Access level for low level must be greater than or equal to 10_8.

MODULE WRITTEN = name.

Module was compiled correctly and binary placed on random file.

MODULES DUMPED.

Dump directive has written modules on binary file.

MULTIPLE CALLS - MODULE TERMINATED.

A CALL command was executed while a previous CALL was still active.

NO DEVICE ASSIGNED.

Device is not assigned. Enter ASSIGN directive before trying to run module.

NO NAME FOR MODULE.

Module name on directive is missing.

GO/DROP. EQOOO-VERIFY DOWN ON ALL MAINFRAMES.

Operator must ensure independent shared device is downed on all mainframes.

GO/DROP. EQOOO-VERIFY UNLOAD ON ALL MAINFRAMES.

Operator ensures independent shared device unloaded on all mainframes.

GO/DROP ESTOOO COO UOO ALOOOO.

Permission is granted to use the device. However, operator must verify that the device is not being shared by another system. If the EST ordinal is not defined, the device was not found in the system.

GO/DROP HALT STOPPED ON ERROR.

Module executed a HALT command and job is running in batch mode. To resume execution, operator must enter a CONTROL POINT GO command where N is the control point number.

GO/DROP INDETERMINANT SITUATION.

If remote NAD rejected (403_{16}) or unable to initialize (107_{16}) for the state/clarifier, enter N.GO if remote host is not using NAD.

GO/DROP. I/O DRAWER(S) OPEN OR IN OCCUPIED.

One or more of the I/O drawer(s) is open or occupied. Correct and have operator enter N.GO.(CR), where N is the control point number.

GO/DROP. UNABLE TO CLR CARTRIDGE(S) FROM MST.

Due to a failure, the cartridge(s) cannot be returned to their proper location within the matrix. Manual intervention is required to remove the cartridge(s) and validation is granted via an entry of N.GO.(CR) from the operator.

GO/DROP. UNABLE TO READ LABEL.

Label could not be read; assignment allowed if OPERATOR GO is entered.

OPERATOR GAVE GO.

Operator has given an X.GO response to the SEND command message which was flashing on the B display.

PARAMETER ERROR .. PARAMETER

Parameter was in error. Reenter with correct parameter.

PARAMETER ERROR .. KL ILLEGAL

Selecting KL while in terminal mode is an illegal command. If this occurs on the MALET call, MALET is aborted and the message is logged in the dayfile. If the SETSW,KL directive is used the message displayed is PARAMETER ERROR .. KL ILLEGAL but MALET is not aborted.

PPU OVERLAY NAME OUT OF RANGE.

Illegal product overlay referenced by a module.

RANDOM FILE FULL = name.

Random file index table is full on module name. Only 200 modules are allowed on the random file.

REQUEST K/L DISPLAY .. MALET.

MALET requests the K/L display to be assigned to its control point.

ROLLOUT MODULE NOT CHECK POINTED.

MALET rolled out before the module was stopped. Use RUN directive to resume execution.

RMS/AL GE 10 DISABLE UNITS NOT UNDER TEST ON CHCC.

A rotating mass storage device has been assigned and validation has been granted at access level greater than or equal to 10. The user can protect the drives on channel cc by setting the drive disable switches in channel cc's controller.

RUN DIRECTIVE INCOMPLETE NEEDS).

RUN directive was entered and never terminated. Enter) to terminate RUN directive.

SOURCE BUFFER FULL.

Source buffer is full. No more source entries, including the one which initiated this message, will be accepted until an executive directive is encountered.

SYNTAX ERROR.

Syntax error on input directive. Reenter.

TOO MANY MODULES ON RUN DIRECTIVE.

RUN directive can contain only 100 module numbers.

VALIDATION DENIED. ADDRESS OUT OF RANGE.

Address is out of range on CVL call. Reassign the device after checking parameters.

VALIDATION DENIED- AL TOO HIGH.

Response code from CVL was 14_8 or access level was too high to gain validation.

VALIDATION DENIED- AL ILLEGAL.

Access level (AL) is illegal. Use legal value.

VALIDATION DENIED- AL TOO LOW.

Access level (AL) not high enough to gain validation.

VALIDATION DENIED- CAN NOT ACCESS NETWORK.

If user is not authorized to access network, access is denied.

VALIDATION DENIED. CEVAL I/O SEQUENCE ERROR.

CEVAL detected an I/O sequence error. Reassign the device after checking parameters.

VALIDATION DENIED-CH,EQ,UN OR EST NEEDED.

Device CH, EQ, UN, or EST ordinal must be provided as a minimum to gain validation to a device.

VALIDATION DENIED- CH IN ERROR.

Channel number is in error. Channel number provided is not legal on system as defined by CVL.

VALIDATION DENIED- CRITICAL INFO MAY BE DESTROYED.

Device contains permanent file information.

VALIDATION DENIED- CSU ERR WHEN LOAD CE CARTRIDGE.

CSU error occurred when loading CE cartridge.

VALIDATION DENIED- CVL RESPONSE OUT OF RANGE.

Response code given by CVL is not within the range of response codes expected by MALET.

VALIDATION DENIED- DC NEEDED EST ABSENT.

Device code is needed if the EST number is absent.

VALIDATION DENIED- DEVICE IN MAINT. STATE. 0000.

Device is in maintenance state already.

VALIDATION DENIED- DEVICE IN TEST STATE. 0000.

Device is in test state.

VALIDATION DENIED- DEVICE NOT FOUND.

CVL could not find device in system EST.

VALIDATION DENIED- DEVICE STATUS ERROR.

The response to CVL when trying to select/connect path RH was not correct, local NAD not running, PATH status could not be obtained, remote NAD busy, and so on.

VALIDATION DENIED. ENTRY FOUND IN EST.

En entry was found in the EST for the device being assigned with an EST=NO parameter. Reassign by EST or do not use EST=NO.

VALIDATION DENIED. ENTRY NOT IN EST.

An entry for the device selected was not found in the EST. Use the EST=NO parameter on the assign statement.

VALIDATION DENIED. FILE NOT FOUND FOR TAPE PREASSIGNMENT.

When using MNE=MT or NT on the MALET assign directive a file was not found that needs to be assigned to MALET's JSN. Drop MALET and assign a file to MALET's JSN and then recall MALET.

VALIDATION DENIED- ILLEGAL CE CARTRIDGE COORDINATE.

Cartridge location was not a CE location. Refer to MSC writeup for user(s) area.

VALIDATION DENIED- IN USE BY SYSTEM.

Device is on or in use by system.

VALIDATION DENIED. INCORRECT CEVAL PARAMETER COMBINATION.

Parameter combination sent to CEVAL by MALET is incorrect. Reassign the device after checking parameters.

VALIDATION DENIED. INCORRECT CEVAL REQUEST PARAMETER(S).

Parameters sent to CEVAL by MALET are incorrect. Reassign the device after checking parameters.

VALIDATION DENIED- ISD ERR WHEN LOADING CE CARTRIDGE.

IDS had an error when loading CE cartridge.

VALIDATION DENIED. LABELED TAPE WAS PREASSIGNED.

Cannot have a labeled tape for tape preassignment. Reassign an unlabeled tape.

VALIDATION DENIED- LOCAL DC BAD.

CVL passed a response code of 12_8. To be implemented at a future date.

VALIDATION DENIED. MNE/DEVICE CODE INCOMPATIBLE.

Device code selected is not the same as the one defined by the mnemonic in the EST. Reassign by EST and do not use device code. Else, the mnemonic in the EST must be changed to the correct type.

VALIDATION DENIED- MSC MOUNTED NOT CE CARTRIDGE.

MSC cartridge mounted, not a CE cartridge.

VALIDATION DENIED- MSS SUBSYSTEM NOT ACTIVE.

MSS executive was not active.

VALIDATION DENIED- NAD CONNECTED TO HOST.

Remote NAD being used by its host. CVL was able to connect path RH.

VALIDATION DENIED- NO TCU ENABLE SET.

The parameter TCU was not set properly on the ASSIGN directive. Reenter ASSIGN directive with correct TCU enables.

VALIDATION DENIED- PICKER CONTAINS CARTRIDGE.

Picker contains cartridge, must be cleared before being assigned.

VALIDATION DENIED- REMOTE DC BAD.

Device code 301_8 was not handled correctly by MALET. This message should never be posted.

VALIDATION DENIED- SN MUST BE PROVIDED.

Serial number must be provided for access level used.

VALIDATION DENIED-SN IN ERROR.

Serial number did not match disk pack serial number, or disk pack serial number not obtainable by system.

VALIDATION DENIED- UNABLE TO ACCESS EQUIPMENT.

Could not access MSS equipment. Therefore, validation is denied.

VALIDATION DENIED- XY LOCATION EMPTY.

xx and yy coordinate did not contain a cartridge, retry with corrected coordinates.

VALIDATION GRANTED- EST000 C00 U00 AL00 DC00.

Validation has been granted for the device.

VALUE TOO LARGE = nnnnnnnn.

Value was too large on parameter register entry. Value must be less than 4096 to fit into a 12-bit field.

VERSION NUMBER ERROR REASSEMBLE.

Current version of MALET does not match version number stored in module. If the user wants to run module, it must be reassembled.

MLD MESSAGES

MLD FL ERR

Address currently being referenced by MLD is outside of MALET's field length.

MLD PARAM CALL ERR

- Address used by MLD in MALET greater than 12 bits.
- Address passed to MLD by MALET does not contain correct data.
- Binary code passed to MLD is too large for buffer area.

MLD VERS ERR

- Correct data not passed to MLD for legal call.
- Version number of MLD does not match MALET. Modification missing for MLD or MALET (REL2B/PL5 or CML3.X).
- Version number of compiled module does not match MLD's version. Reassemble modules.

MLD ILLEG CPU REQ

Illegal request from MALET to MLD. Code was not a start (1) or a resume (2) request.

MLD MOD HEAD ERR

- Data for OB greater than buffer area for OB.
- Line number to start module greater than 167_8.
- Address to store binary out of range of binary buffer.

MLD ILLEG USER ACC

- ENGR. mode not on.
- MALET not installed on system. MLD was called by a program being run from a local file.
- User does not have SYSTEM privileges (NOS).

MLD CH IN USE

The channel requested by MLD is down. Rerun the job after ensuring that another control point does not have dedicated use of the channel.

SBP MESSAGES

SBP ADDRESS NOT IN PROGRAM FL

Address being tested by SBP subroutine is not within field length.

SBP INITIALIZATION ERROR

- Incorrect data passed on call to SBP.
- Error flag set at control point during initialization.

MALET AUTO DCN PERFORMED CH=XX.

- Channel XX being used by MLD was active and full for 5 seconds. SBP did a DCN on channel XX.
- Channel XX being used by MLD was active and empty for 5 seconds. SBP did a DCN on channel.

SBP MODULE VERSION ERROR

- Correct data not passed to SBP for legal call.
- Version number of SBP does not match MALET. Modification missing for SBP or MALET (REL2B/PL5 or CML3.X)

SBP MALET/DRIVER CHANNEL MISMATCH

Channel passed to SBP by MALET did not match channel passed by MLD.

SBP ILLEGAL USER ACCESS

- ENGR. mode not on.
- Program that called SBP was not being run from systems file.
- User does not have system privileges (NOS).

CVL MESSAGES

CVL CALL ERROR

Program calling CVL did not have parameter block set up correctly.

ILLEGAL USER ACCESS

The user tried to perform an operation for which he or she is not validated (for example, attempting to run MALET from nonsystem origin). Have account update for system origin privileges, or run MALET from console under DIS.

USER VALIDATION DENIED

MALET must be run from the console, or if the user has system origin privileges, set engineering mode and resubmit job.

For NOS 2.0 and above, MODVAL parameter SAV=COLD validates the user to use on-line diagnostics.

MALET UTILITY DECK STRUCTURES B

The following deck structures permit the creation of permanent files for the MALET modules and the extraction of documentation from these files for both the NOS and the NOS/BE operating systems.

MALET MODULES - NOS

To install the MALET diagnostic modules as permanent files, run the following job.

```
  MODULES,T1000.
  ACCOUNT/CHARGE CARDS.
  LABEL,OLDPL,VSN=CMLVSN,NT,PE,F=SI.
  SKIPF,OLDPL.
$ UPDATE,Q,R=C.
$ DEFINE,RFILE=T6X.          Direct access file.
$ MALET,I=COMPILE,LO=S.      Compile modules to RFILE.
$ UNLOAD,RFILE.
  7/8/9
$ *C T6X†                    Compile T6X.
$ 7/8/9
$ 7/8/9
  6/7/8/9
```

$ Repeat for each test series wanted.

†For DEMA large sector diagnostics the format: { *ID=xxx / *DEFINE,xxxx / *COMPILE,xxx } is used as follows:

Diagnostic	DEFINE	COMPILE
FMD	*DEFINE,SFMD	*COMPILE,FMD
FHD	*DEFINE,PFMD	*COMPILE,FMD
FLD	*DEFINE,LSFMD	*COMPILE,FMD
D88	*DEFINE,SFMD	*COMPILE,D88
DH8	*DEFINE,PFMD	*COMPILE,D88
DL8	*DEFINE,LSFMD	*COMPILE D88

MALET MODULES DOCUMENTATION - NOS

To acquire external and/or internal documentation from MALET diagnostic modules, run the following job.

```
      DOCIT,T500.
      ACCOUNT/CHARGE CARDS.
      LABEL,CMLVSN,VSN=CMLVSN,NT,PE,F=SI.
      SKIPF,CMLVSN,1.
      COPYBF,CMLVSN,OLDPL.
      UPDATE,Q,C=SOURCE.        Get modules.
      UPDATE,Q.                 Get MLTDOC.
      FTN,I=COMPILE,PL=999999.  Compile MLTDOC.
      LGO,SOURCE.               External to OUTPUT,
    $ REWIND,PRINT.               internal to PRINT.
    $ COPYEI,PRINT,OUTPUT.      Copy internal
      7/8/9                       documentation to OUTPUT.
      *C T6X†                   List test series wanted.
      7/8/9
      *C MLTDOC
      7/8/9
      6/7/8/9
```

$ Eliminate these cards if internal documentation is not required.

MALET MODULES - NOS/BE

To install the MALET diagnostic modules as permanent files, run the following job.

```
      MODULES,T1000,MT1.
      LABEL,CMLVSN,VSN=CMLVSN,D=PE,NORING.
      SKIPF,CMLVSN,1.
      COPYBF,CMLVSN,OLDPL.
    $ UPDATE,Q.
    $ REQUEST,RFILE,*PF.
    $ MALET,I=COMPILE,LO=LS.
    $ CATALOG,RFILE,T6X,ID=YOUR ID.
    $ UNLOAD,RFILE.
      7/8/9
    $ *C T6X†                   Compile T6X.
    $ 7/8/9
    $ 7/8/9
      6/7/8/9
```

$ Repeat for each test series wanted.

† Refer to MALET Modules - NOS earlier in this section.

MALET MODULES DOCUMENTATION - NOS/BE

To acquire external and/or internal documentation from MALET diagnostic modules, run the following job.

```
DOCIT,T500,MT1.
LABEL,CMLVSN,VSN=CMLVSN,D=PE,NORING.
SKIPF,CMLVSN,1.
COPYBF,CMLVSN,OLDPL.
UPDATE,Q,C=SOURCE.          Get modules.
UPDATE,Q.                   Get MLTDOC.
 FTN,I=COMPILE,PL=999999.   Compile MLTDOC.
 LGO,SOURCE.                External to OUTPUT,
$ REWIND,PRINT.               internal to PRINT.
$ COPYBF,PRINT,OUTPUT.
 7/8/9
 *C T6X†                    List test series wanted.
 7/8/9
 *C MLTDOC
 7/8/9
 6/7/8/9
```

$ Eliminate these cards if internal documentation is not required.

EXAMPLES OF RUNNING JOBS

The following jobs assume that the user has created permanent files of the MALET modules.

NOTE

To run any of the following jobs on NOS/BE, change the ATTACH card to:

```
ATTACH,RFILE,nnn;ID=xxx.
```

nnn Name of the MALET diagnostic, test, or utility permanent file.

xxx Your identification number.

†Refer to MALET Modules - NOS earlier in this section.

Sample Jobs	Test or Utility and Description (nnn)	
Run RMS tests at AL=3 on NOS: JOB/ACCOUNT CARDS. ATTACH,RFILE=nnn. or GET,RFILE=nnn. MALET,KL. 7/8/9 ASSIGN,EST=nn,AL=3. P,PX=yyyy. R,nnn. 6/7/8/9	D44	7X54 and 7155 controllers with 844-2X and 844-4X disk drives.
	F44	FSC disk subsystem test (100- and 200-megabyte disk).
	F88	FSC disk subsystem test (317-megabyte disk).
	FMD	Fixed module drive diagnostic.
	FHD	885-4X fixed module drive (DEMA) diagnostic.
	FLD	Fixed module drive (large sector) diagnostic.
Run RMS tests at AL=4 on NOS: JOB/ACCOUNT CARDS. ATTACH,RFILE=nnn. or GET RFILE=nnn. MALET,KL. 7/8/9 ASSIGN,EST=nn,AL=4,SN=xxxxxx. P,PX=yyyy. R,nnn. 6/7/8/9	D88	FMD subsystem confidence test.
	DH8	7155-401/885-4X DEMA disk subsystem confidence test.
	DL8	7155/885 large sector disk subsystem confidence test.
	DTC	7155 with 844-4X drive test.
Run RMS controller tests at AL=10 on NOS: JOB/ACCOUNT CARDS. ATTACH,RFILE=nnn. or GET RFILE=nnn. MALET,KL. 7/8/9 ASSIGN,EST=nn,AL=10,SN=xxxxxx. P,PX=yyyy. R,nnn. 6/7/8/9	FMC	7155-1,1X(FA211-A/B) disk controller diagnostic.
	FHC	7155-4X(FA211-C) disk controller diagnostic.
	FEP	FMD/ESM path test (DEMA).

NOTE

In the previous jobs:

PX	Any parameter register P0 through P9.
yyyy	The value to be set in that parameter register.

For the correct parameter settings and access levels, refer to the 99 module or to the external documentation of the selected diagnostic, test, or utility.

Sample Jobs	Test or Utility and Description (nnn)	

Run Intelligent Peripheral Monitors/Tests at AL=3 on NOS:

JOB/ACCOUNT CARDS. ATTACH,RFILE=nnn. or GET,RFILE=nnn. MALET,KL. 7/8/9 ASSIGN,EST=nn,AL=3. P,PX=yyyy. R,nnn. or R,nnnxx. (xx is one 6/7/8/9 module number.)	FSD	ISD subsystem monitor and utility test. AL=10 must be used for some modules.
	NIP	Nonimpact printer test. Runs at AL=10.
	DTI	ISD/895 subsystem data path integrity test.

Run Formatting Utilities at AL=4 on NOS:

NOTE

Some modules may need AL=10.

JOB/ACCOUNT CARDS. ATTACH,RFILE=nnn. or GET,RFILE=nnn. MALET,KL. 7/8/9 ASSIGN,EST=nn,AL=4,SN=xxxxxx. P,PX=yyyy. R,nnnxx. (xx is the module.) 6/7/8/9	FMU	Format utility for 885-1X, 885-4X (DEMA), 885 (large sector), and 844-4X disk drives.
	FFU	Format utility for federal standard disk subsystems.
	DFU	Format utility for 834, 836, and 895 disks.

Run Buffer Controller Tests at AL=10 on NOS:

JOB/ACCOUNT CARDS. ATTACH,RFILE=nnn. or GET,RFILE=nnn. ATTACH,MY8BC. or GET,MY8BC. MALET,KL,S=MY8BC. 7/8/9 ASSIGN,EST=nn,AL=10. P,PX=yyyy. R,nnn. 6/7/8/9	MY8	Buffer controller memory test. BUCAL code from file 3 or 4 of CML. (File 3 is MY8BC for MY8 and file 4 is BCXBC for BCX). Both files are saved when defining 66X or 7X54 using the CMLINST procedures.
	BCX	Buffer controller command test.

Run Loader/Monitors at AL=10 on NOS:

JOB/ACCOUNT CARDS. ATTACH,RFILE=nnn. or GET,RFILE=nnn. ATTACH,NDM,NDP. or GET,NDM,NDP. MALET,KL. 7/8/9 ASSIGN,EST=nn,AL=10. SCRATCH=zzz. Where zzz is NDM or NDP. P,PX=yyyy. R,nnn. 6/7/8/9	FLM	Federal standard channel loader monitor.
	CLM	CYBER Channel Coupler loader monitor.

Sample Jobs	Test or Utility and Description (nnn)	
Run Memory Test at AL=10 on NOS:		
JOB/ACCOUNT CARDS. ATTACH,RFILE=nnn. or GET,RFILE=nnn. MALET,KL. 7/8/9 ASSIGN,EST=nn,AL=10. P,PX=yyyy. R,nnn. 6/7/8/9	FSM	Federal standard channel memory test.
	CCM	CYBER Channel Coupler memory test.
Run Tape Tests on NOS:		
JOB/ACCOUNT CARDS. ATTACH,RFILE=nnn. or GET,RFILE=nnn. MALET,KL. 7/8/9 ASSIGN,EST=nn. P,PX=yyyy. R,nnn. 6/7/8/9	T7X	67X tape transport test.
	T6X	66X tape transport test.
	T5X	60X/65X tape transport test.
	F7X	Federal standard channel tape subsystem test.
Run Tape Controller Test at AL=10 on NOS:		
JOB/ACCOUNT CARDS. ATTACH,RFILE=ATC. or GET,RFILE=ATC. MALET,KL. 7/8/9 ASSIGN,EST=nn,AL=10. P,PX=yyyy. R,ATC. 6/7/8/9	ATC	ATS-67X controller test.
Run Printer Tests on NOS:		
JOB/ACCOUNT CARDS. ATTACH,RFILE=nnn. or GET,RFILE=nnn. MALET,KL. 7/8/9 ASSIGN,EST=nn,AL=10. P,PX=yyyy. R,FTP20. To load image. R,nnn. 6/7/8/9	FTP	580 fastrain printer test.
	LP1	512 printer test.
	LPE	512 exerciser.
Run Card Reader and Punch Tests on NOS:		
JOB/ACCOUNT CARDS. ATTACH,RFILE=nnn. or GET,RFILE=nnn. ATTACH,CR1SCR. or GET,CR1SCR. For CR1. MALET,KL. 7/8/9 ASSIGN,EST=nn,AL=10. SCRATCH=CR1SCR. For CR1. P,PX=yyyy. R,nnn. 6/7/8/9	CR1	405 card reader test.
	CP1	415 card punch test.
	CRE	405 exerciser.
	CPE	415 exerciser.

Sample Jobs	Test or Utility and Description (nnn)	

Run Communication Controller Tests on NOS:

```
JOB/ACCOUNT CARDS.
ATTACH,RFILE=nnn. or GET,RFILE=nnn.
MALET,KL.
7/8/9
ASSIGN,EST=nn,AL=20.
P,PX=yyyy.
R,nnn.
6/7/8/9
```

RT3	6673/6674 communication controller test.
RT5	6671 (201 or 103 mode) communication controller test.
TT3	6676 (103 mode) communication controller test.

Run 2550 Communication Controller Tests on NOS:

```
JOB/ACCOUNT CARDS.
ATTACH,RFILE=nnn. or GET,RFILE=nnn.
MALET,KL.
7/8/9
ASSIGN,EST=nn,AL=20.
P,PX=yyyy.
R,nnn.
6/7/8/9
```

TFE	2550 emulator (201 or 6671 mode) communication controller.
TFF	2550 coupler diagnostic.

Run 2550 Diagnostic Loader on NOS:

```
JOB/ACCOUNT CARDS.
ATTACH,RFILE=TFL. or GET,RFILE=TFL.
ATTACH,zzzzzz. or GET,zzzzzz.
MALET,KL.
7/8/9
SCRATCH=zzzzzz.
ASSIGN,EST=nn,AL=20.
P,PX=yyyy.
R,TFL.
6/7/8/9
```

TFL — 2550 off-line diagnostic loader.

NOTE

On the ATTACH or GET of zzzzzz, all the files can be acquired before the MALET,KL statement. zzzzzz is one of the following:

LDCHK3	Loadcheck
MPINS	Instruction test
MPMOS	MOS memory storage/addressing test
MOSMA	MOS memory parity/protect test
MPRTC	Protect test
MINEM	Micromemory read/write test
MIINS	Microinstruction test
MST041	Multiplex system test
CEL042	Cyclic encoder test
CPL040	6000 coupler test
CPLX40	Expansion coupler test
TTYA08	CRT/TTY console test
UTOPIA	Utility routines
CCP48K	CCP 1.0 diagnostics

Sample Jobs	Test or Utility and Description (nnn)	

Run Network Access Device (NAD) Tests on Local NAD on NOS:

NOTE

LCN must have controller running. LCI destroys controlware in the NAD in which it is running.

JOB/ACCOUNT CARDS. ATTACH,RFILE=nnn. or GET,RFILE=nnn. MALET,KL. 7/8/9 ASSIGN,EST=nn,DC=300,AL=x. P,PX=yyyy. R,nnn. 6/7/8/9	LCI	LCN 170 device interface test. Must run at AL=10.
	LCN	Loosely coupled network test. Can run at AL=5.
	LCM	LCN memory test. Can run at AL=5.

Run Network Access Device (NAD) Tests on Remote NAD on NOS:

NOTE

Because NAD is a hexadecimal machine, it is necessary that the access code (AC) and the logical trunk address (LTA) be entered as hexadecimal parameters using the letter H following the parameter value.

Controlware must be loaded in the local NAD (refer to the NOS LOADBC command in the NOS 2 Analysis Handbook).

JOB/ACCOUNT CARDS. ATTACH,RFILE=nnn. or GET,RFILE=nnn. MALET,KL. 7/8/9 ASSIGN,EST=nn,DC=301,AL=6,TCU=x, LTA=xH,AC=xxxxH. P,PX=yyyy. R,nnn. 6/7/8/9	LCN	Loosely coupled network test.
	LCM	LCN memory test.

Sample Jobs	Test or Utility and Description (nnn)	

Run Network Access Device (NAD) Tests NDM, NDP, and NDT in Local NAD on NOS using NAD Loader/Monitor (NLM):

```
JOB/ACCOUNT CARDS.
ATTACH,RFILE=NLM. or GET,RFILE=NLM.
ATTACH,NDM,NDP,NDT. or GET,NDM,NDP,NDT.
MALET,KL.
7/8/9
ASSIGN,EST=nn,DC=300,AL=5.
SCRATCH=zzz.   Where zzz is NDM, NDP or NDT.
P,PX=yyyy.
R,NLM.
6/7/8/9
```

NDM	NAD memory test.
NDP	NAD processor test.
NDT	NAD trunk test.

Run Network Access Device (NAD) Tests on Remote NAD on NOS:

NOTE

Controlware must be loaded in the local NAD (refer to the NOS LOADBC command in the NOS 2 Analysis Handbook).

```
JOB/ACCOUNT CARDS.
ATTACH,RFILE=NLM. or GET,RFILE=NLM.
ATTACH,NDM,NDP,NDT. or GET,NDM,NDP,NDT.
MALET,KL.
7/8/9
ASSIGN,EST=nn,DC=301,AL=6,TCU=x,LTA=xH,AC=xxxxH.
SCRATCH=zzz.   Where zzz is NDM, NDP or NDT.
P,PX=yyyy.
R,NLM.
6/7/8/9
```

Run Special Unit Tests at AL=10 on NOS:

```
JOB/ACCOUNT CARDS.
ATTACH,RFILE=nnn. or GET,RFILE=nnn.
MALET,KL.
7/8/9
ASSIGN,EST=nn,AL=10.
P,PX=yyyy.
R,nnn.
6/7/8/9
```

CRP	CYBERPLUS ring port test.
PDP	CYBER UNIBUS and PDP11 test.
CIU	PLATO communications interface test.
ITU	Intelligent small magnetic tape subsystem utility routine.
DMA	CYBERPLUS direct memory access unit test.

ACCESS LEVELS C

Table C-1 describes the access levels needed while running MALET.

Table C-1. Access Levels (Sheet 1 of 2)

Level	Description	
	RMS	Non-RMS
0	Illegal (if specified on ASSIGN).	Dedicated unit, high level I/O.
1	Read preallocated area only, high-level I/O.	Illegal.
2†	Read any normal data area only, high-level I/O.	Illegal.
3†	Read any normal data area, write preallocated area only, high-level I/O.	Illegal.
4	Read/write any normal data area, high-level I/O dedicated unit.	Illegal.
5	Illegal.	LCN NAD dedicated. Code in NAD assigned may be written.
6	Illegal.	LCN NAD local and remote dedicated. Code may be rewritten in both NADs.
7	Reserved.	Illegal.
10	For NOS: All removable devices on the dedicated controller must be GLOBALLY UNLOADED. For NOS Version 2 the device under test must be DOWNED. All SYSTEM and TEMP devices on the dedicated controller must be removed from the EST. For NOS Version 2 REDEFINE†† can be used to logically remove device channels from the EST.	Dedicated controller and all units. High- and low-level I/O. May destroy controlware.

†Access level = 4 is required for READs outside preallocated areas, if SECURE has been defined on the intallation of MALET drivers. To lower the access level required, a command *PURGE,*SECURE must be used during a diagnostic PL update (REL2B/PL5). Refer to the NOS or NOS/BE Installation Handbook for further information.

††UNLOAD and REDEFINE can be referenced in the NOS 2 Operator/Analyst Handbook.

Table C-1. Access Levels (Sheet 2 of 2)

Level	Description	
	RMS	Non-RMS
	For NOS/BE: The dedicated controller must be OFFED and the device under test must be IDLED.	
11-17	Illegal (reserved).	Illegal (reserved).
20	For NOS: All removable devices on the dedicated channel must be GLOBALLY UNLOADED. For NOS Version 2 the device under test must be DOWNED. All SYSTEM and TEMP devices on the dedicated channel must be removed from the EST. For NOS Version 2 REDEFINE † can be used to logically remove device channels from the EST. For NOS/BE: The dedicated controller must be OFFED and the device under test must be IDLED.	Dedicated channels, all units, and controllers. All devices must be off and not in use on the channel since master clears may be performed. High- and low-level I/O. Controlware may be destroyed.
21	Same as 20, plus I/O channel is left in hung state if a channel hang occurs.	

†UNLOAD and REDEFINE can be referenced in the NOS 2 Operator/Analyst Handbook.

If validation is not granted and the system K/L display switch is set to off, MALET drops from the control point. If the K/L switch is on, an error message appears but the control point is not dropped.

MALET OUTPUT LISTING SAMPLE D

MALET output listings are illustrated in figures D-1 through D-3. The content of the listing varies, as follows, depending upon the list options selected:

Listing Contents	List Options
Items 1, 2, 9, and 12	S or none
Items 1 through 9	I
Items 1, 2, 5 through 16	L
Items 1, 2, 9, 17 through 20	L and MAP

Both L and MAP must be specified if a global cross-reference map is desired.

Selecting the S option is not evident in the listing output, but is required if source code is to be carried with the module binary.

(1) Header containing the level of MALET used to compile the module binary. This level corresponds to the level of the Concurrent Maintenance Library.

(2) The date and time that the module was compiled.

(3) Imbedded documentation not carried with the module binary.

(4) Update correction set identifier or deckname, and sequence number of the line within the correction set or deck. These line identifiers appear only when the source code has been extracted from a program library.

(5) Octal line number within MALET's source buffer, numbered 000 through 167 octal.

(6) Source line statement numbers; one or two-digit octal or decimal numbers in the range of 0 through 77 octal or 0 through 63 decimal.

(7) Source code statements recognized by the compiler.

(8) Comments carried with the module binary (if S is selected).

(9) Executive directives processed by MALET, indicating the time the directive was processed.

(10) The starting PPU address where the generated instructions (11) are loaded during module execution.

(11) The executable PPU instructions which the compiler generated for the source line (6) to the left.

(12) Header for the cross-reference table for the module indicated.

(13) The MALET executive switches, registers, and buffers referenced within the module.

(14) Statement numbers referenced within the module.

(15) The source line numbers that references the executive switches, registers, buffers (13), or statement numbers (14).

(16) Informative message identifying the module which was written to the random file.

(17) Header identifying the global cross-reference map.

(18) The executive switch, register, or buffer which was referenced.

(19) The module number(s) which referenced the element (18).

(20) The source line within the module (19) where the element (18) was referenced.

```
MALET L155       MAINTENANCE APPLICATION LANGUAGE FOR EQUIPMENT TESTING.     84/01/26. 07.51.45.
 (1)                                                                                  (2)

 000    MODULE MY802,BC(02,03,04,05,42,52)  - RUN MY8                          MY8       552
    *** MY802 - RUN MY8                                                        MY8       553
    *                                                                          MY8       554
    **  MY802 - EXTERNAL DOCUMENTATION                                         MY8       555
    *                                                                          MY8       556
    *   OVERVIEW -                                                             MY8       557
 (3)*                                                                          MY8       558
    *        THIS MODULE RUNS MY8 AFTER IT HAS BEEN  DOWNLOADED               MY8       559
    *        INTO  THE  BC  BY  MODULE  MY801.  A 60 OCTAL WORD               MY8       560
    *        FUNCTION TABLE IS  USED  TO  EXCHANGE  INFORMATION        (4)    MY8       561
    *        BETWEEN  THE PPU AND THE BC.  A FUNCTION CODE OF 1               MY8       562
    *        ISSUED BY THE PPU TO THE BC CAUSES THE BC TO  SEND               MY8       563
    *        THE  FUNCTION  TABLE TO THE PPU SO THAT PASS COUNT               MY8       564
    *        AND ERROR STATUS CAN BE MONITORED. A FUNCTION CODE               MY8       565
    *        OF  0 TO THE BC CAUSES THE BC TO READ THE FUNCTION               MY8       566
    *  (6)   TABLE FROM THE PPU SO THAT THE BC CAN PERFORM  THE               MY8       567
    *  /     REQUESTED  ACTION (THE ONLY ACTION THAT IS REQUES-               MY8       568
    * /      TED BY THE PPU IS TO RUN A PASS OF MY8).                         MY8       569
  001  1 FORMAT MY802 - MY8 IS EXECUTING                                       MY8       617
  002  2 FORMAT         MY8 FUNCTION TABLE                                     MY8       618
  003  3 FORMAT  0-3 =  *H*H  *H*H  *H*H  *H*H                                 MY8       619
  004  4 FORMAT  4-7 =  *H*H  *H*H  *H*H  *H*H                                 MY8       620
  005  5 FORMAT  8-B =  *H*H  *H*H  *H*H  *H*H                                 MY8       621
  006  6 FORMAT  C-F =  *H*H  *H*H  *H*H  *H*H                                 MY8       622
  007  7 FORMAT  MY8 PASS=*DEC SECTION=*DEC                                    MY8       623
  010 10 FORMAT    ADDRESS OF ERROR (A) = *H*H                                 MY8       624
(5)011 11 FORMAT    ACTUAL DATA     (B1) = *H*H                               MY8       625
  012 12 FORMAT    EXPECTED DATA   (B2) = *H*H                                 MY8       626
  013 13 FORMAT    ERROR STOP CODE      = *H*H                                 MY8       627
  014 14 FORMAT    ERROR COUNT = *DEC                                          MY8       628
  015 16 FORMAT MY802 - ABORT ON .IN. COMMAND.                                 MY8       629
  016 17 FORMAT MY802 - ABORT ON .OUT. COMMAND.                                MY8       630
  017 20 FORMAT MY802 ERROR(S) IN MODULE = *DEC                                MY8       631
  020 21 FORMAT MY802 - NO BC RESPONSE TO A FCN IN 30 SEC                      MY8       632
  021                                                                          MY8       633
  022 30 MSG 1 TO DISPLAY,PRINT    *POST RUNNING MESSAGE                       MY8       634
  023    B1 = 0                    *CLEAR PASS COUNT                           MY8       635
  024    WC = 60  (7)                                                          MY8       636
    .
    .
    .
  110    MSG 2 TO LINE 1                                                       MY8       688
  111    GOTO 33                                                               MY8       689
  112 /                    (8)                                                 MY8       690
  113 /--------ABORT PROCESSOR-------------------------                       MY8       691
  114 /                                                                        MY8       692
  115 70 B7 = B7 + 1                *BUMP ERROR COUNTER                        MY8       693
  116    MSG 7 (B1,IB(23)) TO LINE 10  *PASS AND SECTION COUNTS               MY8       694
  117    MSG B5 TO LINE 11          *REPORT ABORT CONDITION                   MY8       695
  120    MSG EM TO LINE 12          *PRODUCT OVERLAY ERROR MSG                MY8       696
  121    PICTURE                                                              MY8       697
  122    HALT                                                                 MY8       698
  123    BLANK                                                                MY8       699
  124    GOTO 31                                                              MY8       700
  125 77 END 30                                                               MY8       701
07.51.45.DIRECTIVE=COMPILE   - MY802 -                                        MY8       702
          (9)
```

Figure D-1. MALET Source Listing: List Option I

```
MALET L155       MAINTENANCE APPLICATION LANGUAGE FOR EQUIPMENT TESTING.     84/01/26. 07.51.45.

000       MODULE MY802,BC(02,03,04,05,42,52)  - RUN MY8
001     1 FORMAT MY802 - MY8 IS EXECUTING
002     2 FORMAT         MY8 FUNCTION TABLE
003     3 FORMAT  0-3 =  *H*H  *H*H  *H*H  *H*H
004     4 FORMAT  4-7 =  *H*H  *H*H  *H*H  *H*H
005     5 FORMAT  8-B =  *H*H  *H*H  *H*H  *H*H
006     6 FORMAT  C-F =  *H*H  *H*H  *H*H  *H*H
007     7 FORMAT  MY8 PASS=*DEC SECTION=*DEC
010    10 FORMAT    ADDRESS OF ERROR (A) = *H*H
011    11 FORMAT    ACTUAL DATA     (B1) = *H*H
012    12 FORMAT    EXPECTED DATA   (B2) = *H*H
013    13 FORMAT    ERROR STOP CODE      = *H*H
014    14 FORMAT    ERROR COUNT = *DEC
015    16 FORMAT MY802 - ABORT ON .IN. COMMAND.
016    17 FORMAT MY802 - ABORT ON .OUT. COMMAND.
017    20 FORMAT MY802 ERROR(S) IN MODULE = *DEC
020    21 FORMAT MY802 - NO BC RESPONSE TO A FCN IN 30 SEC
021                                                                  (10)                          (11)
022    30 MSG 1 TO DISPLAY,PRINT    *POST RUNNING MESSAGE            1100=1401 3402 1400 3403
3401 1455 0200 2367
023       B1 = 0                    *CLEAR PASS COUNT                1110=1400 3421
024       WC = 60                                                    1112=1460 3430
025    31 MSG 2 TO LINE 1                                            1114=1402 3402 1400 3403
3401 1401 0200 2367
026       B0 = 0                    *CLEAR FUNCTION RETRY COUNT      1124=1400 3420
027    32 RES                       *GET CHANNEL                     1126=1427 3446 1400 0200 4520
030       FUNC 1, ABT 40            *REQUEST FUNCTION TABLE FROM BC 1133=3646 1401 3410 2033
5447 0200 4544
031       B0 = 0                    *CLEAR FUNCTION RETRY COUNT      1142=1400 3420
032       B5 = 16                                                    1144=1416 3425
 .
 .
 .
110       MSG 2 TO LINE 1                                            1620=1402 3402 1400 3403
3401 1401 0200 2367
111       GOTO 33                                                    1630=2023 1306 0100 2414
112    /
113    /--------ABORT PROCESSOR-------------------------
114    /
115    70 B7 = B7 + 1                  *BUMP ERROR COUNTER           1634=3027 1601 3427
116       MSG 7 (B1,IB(23)) TO LINE 10  *PASS AND SECTION COUNTS     1637=1407 3402 3021 3407
5000 6513 3410 1402 3403 1400 3401
                                                                          1410 0200 2367
117       MSG B5 TO LINE 11            *REPORT ABORT CONDITION       1655=3025 3402 1400 3403
3401 1411 0200 2367
120       MSG EM TO LINE 12            *PRODUCT OVERLAY ERROR MSG    1665=3051 3402 1400 3403
3401 1412 0200 2367
121       PICTURE                                                    1675=1400 0200 2377
122       HALT                                                       1700=1400 0200 2353
123       BLANK                                                      1703=1403 0200 2305
124       GOTO 31                                                    1706=2012 5114 0100 2414
125    77 END 30                                                     1712=0100 2330
```

Figure D-2. MALET Source Listing: List Option L (Sheet 1 of 2)

```
MALET L155       MAINTENANCE APPLICATION LANGUAGE FOR EQUIPMENT TESTING.     84/01/26. 07.51.45.

                                 (12)
CROSS-REFERENCE TABLE FOR    MY802
B0              026      031      071
B1              023      042      054      057      116
B5              032      047      051      072      117
B7              061      062      103      103      104      115      115
CM  (13)        060
EM              120
ES              060
IB              035      036      037      040      041      042      045      054      055      056      077      077
IB              100      100      101      101      102      102      116
.
.
.                                          (15)
3               003      036
30              022      125
31              025      060      124
32              027      055      056      057      071
33              046      111
34  (14)        054
35              061      063
4               004      037
40              030      067
5               005      040
50              045      077
6               006      041
7               007      042      054      116
70              033      050      052      073      115
77              063      125
MODULE WRITTEN = MY802
        (16)
```

Figure D-2. MALET Source Listing: List Option L (Sheet 2 of 2)

```
MALET L155        MAINTENANCE APPLICATION LANGUAGE FOR EQUIPMENT TESTING.     84/01/26. 07.51.45.

                  (17)
GLOBAL CROSS-REFERENCE MAP
B0        MY801          026     026     027     037
          MY802          026     031     071
          MY803          031     031     033     043
B1        MY800          020     046
          MY801          021     022     032     052     053     067     070
 (18)     MY802          023     042     054     057     116
          MY803  (19)    024     025     036     063     064     101     102
B2        MY800          061     062
          MY803          032     046     055
B3        MY800          056     061     061     062     062     062     064
B4                       057     062     063     063
          MY803  (20)    052     053     054     056     125     141     142     143     144
B5        MY800          023     025     027     032     034     036     040     060     074
          MY801          057     065     075     100     122
  .
  .
  .
WC        MY800          021     041     065
          MY801          060     063     064     073     074
          MY802          024
          MY803          072     075     076     105     106     130     131
WT        MY800          030     037     060
          MY801          025     042     043     060     063     071     072     073
          MY803          030     047     050     072     075     103     104     105     136
137
```

Figure D-3. MALET Global Cross-Reference Map: List Options L and MAP

COMMENT SHEET

MANUAL TITLE: Control Data Concurrent Maintenance Library Reference Manual
PUBLICATION NO.: 60455980 REVISION: P

NAME:_____

COMPANY:_____

STREET ADDRESS:_____

CITY:_____ STATE:_____ ZIP CODE:_____

This form is not intended to be used as an order blank. Control Data Corporation welcomes your evaluation of this manual. Please indicate any errors, suggested additions or deletions, or general comments below (please include page number references).

☐ Please Reply ☐ No Reply Necessary

CUT ALONG LINE

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

FOLD FOLD

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 8241 MINNEAPOLIS, MINN.

POSTAGE WILL BE PAID BY

CONTROL DATA CORPORATION

Publications and Graphics Division
4201 North Lexington Avenue
St. Paul, Minnesota 55126-6198

CUT ALONG LINE

FOLD FOLD