60454200

CONTROL DATA CORPORATION

NOS
ON-LINE MAINTENANCE SOFTWARE
REFERENCE MANUAL

CDC® COMPUTER SYSTEMS:
- CYBER 180
- CYBER 170
  - MODELS 172, 173, 174, 175, 176, 720, 730, 740, 750, 760, 810, 815, 825, 835, 845, 855, 865, 875
- CYBER 70
  - MODELS 71, 72, 73, 74
- 6000 SERIES

REVISION RECORD

REVISION	DESCRIPTION
A (12-07-76)	Manual released.
B (03-01-78)	Sections 1, 2, and 5 are revised. This revision obsoletes the previous edition.
C (10-02-78)	Section 4 is revised.
D (09-26-79)	Manual revised to add CPU/memory on-line tests.
E (10-15-80)	Manual revised to include the CYBER 170 models 720, 730, 740, 750, and 760. Because of extensive changes to this manual, change bars and dots are not used and all pages reflect the latest revision level. This edition obsoletes all previous editions.
F (05-15-81)	Manual revised to PSR Level 581A. Revision includes addition of HIVS testing. The KEDIAG section was deleted. Miscellaneous editorial and technical corrections were made. This edition obsoletes all previous editions.
G (10-29-82)	Manual revised to include CYBER 170 models 825, 835, 855, 865, and 875 NOS enhancements. Miscellaneous technical corrections are made. Due to extensive changes, change bars and dots are not used, and all pages reflect the latest revision. This edition obsoletes all previous editions and replaces publication 60458360.
H (01-30-84)	Manual revised. Section 2 is deleted. Miscellaneous editorial and technical corrections are made. Due to extensive changes, change bars and dots are not used, and all pages reflect the latest revision. MDD added. This edition obsoletes all previous editions.
J (10-22-84)	Manual revised; CYBER 180 information, RDF section, and OPMSG added.

Publication No.
60454200

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

Address comments concerning this manual to:

Control Data Corporation
Publications and Graphics Division
4201 North Lexington Avenue
St. Paul, Minnesota 55112

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	PAGE	REV	PAGE	REV	PAGE	REV	PAGE	REV
Front Cover	-								
Title Page	-								
2	J								
3/4	J								
5/6	J								
7	J								
8	J								
1-1	J								
1-2	H								
2-1	J								
2-2	H								
2-3	H								
2-4	H								
2-5	H								
2-6	H								
2-7	H								
2-8	H								
2-9	H								
2-10	H								
2-11	H								
2-12	H								
2-13	H								
2-14	H								
2-15	H								
2-16	H								
2-17	H								
2-18	H								
2-19	H								
2-20	H								
2-21	H								
2-22	H								
2-23	H								
2-24	J								
2-25	J								
3-1	J								
3-2	J								
3-3	J								
3-4	J								
3-5	J								
3-6	J								
3-7	J								
3-8	J								
3-9	J								
3-10	J								
3-11	J								
3-12	J								
3-13	J								
Comment Sheet	J								
Back Cover	-								

PREFACE

This manual provides the information necessary for the installation and use of concurrent maintenance software routines designed to run on the CONTROL DATA® 6000, CDC® CYBER 70, CDC CYBER 170, and CDC CYBER 180 Computer Systems. These routines are to be installed and run under the control of the NOS Operating System.

RELATED PUBLICATIONS

The NOS/BE On-Line Maintenance Software Reference Manual, publication number 60453900, contains information similar to that in this manual. However, it should not be used in lieu of this manual except as complementary reading.

NOTE

The On-Line Diagnostics and the Off-Line Diagnostics of the same name are not necessarily at the same correction level; therefore, different results may occur.

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

FIGURES

TABLES

To understand the purpose of concurrent or on-line maintenance, it is necessary to understand basic system maintenance concepts. This introduction presents those concepts, lists the goals of the 6000/CYBER 70/CYBER 170/180 maintenance software features, and shows how these goals are accomplished.

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 the necessity of occasional emergency maintenance (EM) when a failure occurs. Unfortunately, time taken for EM becomes irrecoverable user time, as is the time between system failure and failure detection (mean time to detection, or MTTD). MTTD represents a need for rerun time for the user, and must be added to EM time to determine total time lost to the user. This relationship is illustrated in figure 1-1.

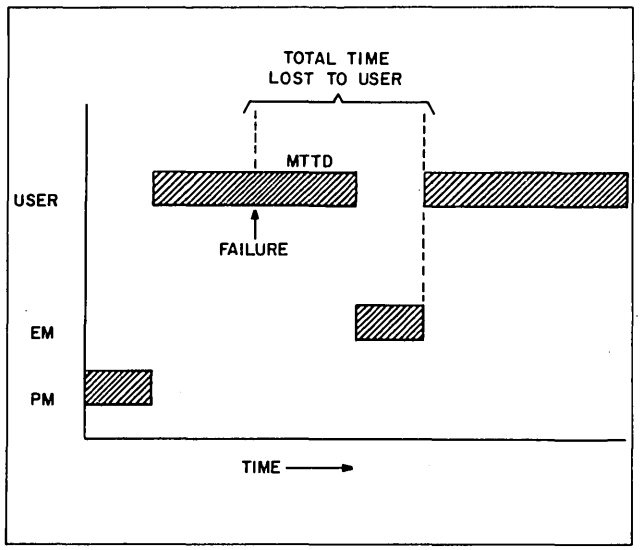

Figure 1-1. User Time Versus Maintenance Time

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

The philosophy of concurrent or on-line maintenance concerning user time and PM time differs considerably from the traditional approach. PM time is regarded as best used for the prevention of failures, rather than in their detection, and should occupy as little of the total system time as possible, thereby affording the user maximum, uninterrupted access.

To this end, operating systems have the ability to detect errors in peripheral equipment as soon as they occur and to enter the details into an engineering file. The contents of this file can be examined by the user and/or customer engineer to analyze these entries and obtain useful information with regard to the type and quantity of errors that occurred on various peripheral equipment, the date and time at which errors occurred, and how much usage the peripheral equipment has had over a period of time, so that PM time can be beneficially scheduled.

When a failure is detected, isolation and correction are attempted, using peripheral processor maintenance tool programs being run as user jobs. The information logged in the engineering file and the printout from the maintenance tool programs 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 or not the system can continue to operate effectively with the failure condition.

As the flowchart in figure 1-2 illustrates, the ultimate goal is to maintain effective system operability despite occasional hardware malfunctions.

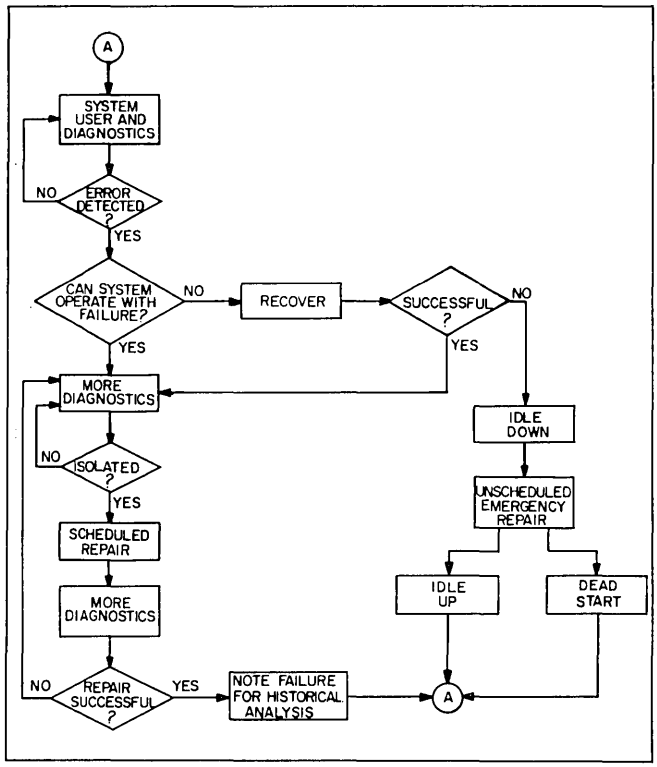

Figure 1-2. Maintenance Flowchart

PM time should not be used for EM. PM time should be used to prevent failures by performing regularly scheduled procedures, including replacement of suspect components. Concurrent 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.

Concurrent maintenance provides the capability of recognizing and curbing system degradation before collapse occurs. Minor malfunctions are detected and proper action is taken before a fatal system failure results. Degradation in a system is inevitable, but it can be recognized and controlled by analysis of the engineering file and proper application of sound concurrent maintenance.

If a system failure does occur and emergency maintenance is required, the engineering file contains information that aids system personnel in isolating and correcting the failure.

The following three objectives summarize the purpose of concurrent maintenance:

1. Increased system availability

   The ability to detect and repair faults in a real-time manner allows greater user access.

2. Increased user confidence

   The user always knows the level of system operability.

3. Improved maintainability

   Constant checking of system viability enables detection of potential failures and hidden problems, resulting in improved maintainability.

These goals are achieved because of the following interacting features of an effective concurrent maintenance system:

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

Concurrent maintenance assures confidence in the system. The user knows downtime is significantly reduced, if not eliminated, by the ability of the system to recognize and report degradation before system failure. PM time is used for failure prevention, as intended, rather than for failure detection, as in the case of off-line diagnostics. Failures, whether minor or major, are detected much sooner than they would be by the PM diagnostics. This early detection greatly reduces the necessity of job reruns, increasing availability. Because the operating system has the ability to log malfunction information in the engineering file and point out the information shortly after the failure occurs, the user is aware of the state of the system at all times. There is no need to wait until PM time to know if jobs are successful or not. Test results are more realistic and credible because the system is tested in the user environment under the operating system rather than simulated in an off-line mode.

This section describes the central processor and memory test routines that have been modified to execute concurrently with normal user jobs.

A brief description of each test precedes a list of selectable parameters, where applicable, and a list of significant locations. For a more detailed description of the tests, refer to the REL2B source listing.

The tests were selected on the basis of hardware coverage and error detection capability. Therefore, the selection of tests to be executed depends upon the CPU and memory type. As a consequence, tests are categorized as follows:

6400; CYBER 170, Models 72, 73; CPU 1 of 6700; CPU 1 of CYBER 170, Model 74-2x

ALX	EC3
CMU	FST
CT3	MRG
CU1	MY1
	RAN

6600; CYBER 70, Model 74; CPU 0 of 6700; CPU 0 of CYBER 70, Model 74-2x

ALX	IWS
CT3	MRG
CU1	MY1
EC3	RAN
FM2	

CYBER 170, Models 172, 173, 174, 720, 730

ALX	EC3
CMU	FST
CSU	MEM
CT3	MRG
CU1	RAN

CYBER 170, Models 175, 740, 750, 760

ALX	EC3
CSU	FST
CT7	MEM
CU1	RAN

CYBER 170, Model 176

CTB	MEM
LCM	

CYBER 170/180, Models 810 through 855

ALX	EC8
CTB	FS8
CT8	MRG
CU8	RA8

CYBER 170, Models 865, 875

ALX	EC8
CT6	FS8
CU8	RA8
CSU	

In order to standardize and simplify the conversion of these tests to execute concurrently, a common package (CPUMEM) was created. It also has the capability of interfacing each test with the host operating system and performing the following functions:

- Ensures that each selected test is designed to execute in a specific CPU type
- Reports all illegal parameters
- Reports all hardware failures and addresses
- Maintains a pass counter

CPUMEM resides in all CPU/memory tests from address 110 to address 1477, inclusive. The important locations are as follows:

Address	Tag Name	Description
160	C.PASNO	Pass counter

By executing the common package, CPUMEM, each test is checked to verify that it is designed to execute on the current CPU type. If a mismatch occurs, CPUMEM issues the message:

```
TEST/CPU TYPE MISMATCH
```

and terminates the test. If the match is found to be correct, CPUMEM issues the message:

```
XXX MSL VZ.Z.Z
```

XXX	The test mnemonic.
Z.Z.Z	The current MSL revision level.

Execution of the test is then initiated.

At the present time, five programs have selectable parameters: CT3, CT7, CSU, CTB, and LCM. If an illegal parameter is selected, CPUMEM issues the message:

```
xxx ENDED - ILLEGAL PARAMETER
```

xxx	The test mnemonic.

and the test is terminated.

If a test program detects an error, the following informational error message is issued to the dayfile and error log:

CPU FAILURE, xxx P=aaaaaa PC=bbbbbbbb

xxx	The test mnemonic.
aaaaaa	The failing P address of the test.
bbbbbbbb	The pass count.

The test then pauses and the following message is displayed:

CPU FAILURE - GO TO CONTINUE TESTING

If a GO command is given, the test continues execution, looking for more errors.

OPERATING PROCEDURES

The central processor and memory test programs described in this section can be called by any one of three methods:

1. By console input

   a. By using the command MAINTENANCE, all tests appropriate to the current CPU are called and executed. The categories of tests and corresponding CPU types are listed in the introduction to this section.

   b. Individual tests can be called and executed by using the command X.zzz, where zzz is the test mnemonic. This type of call enters a MODE(0) control card before the test is called, so that termination on error detection is avoided. The operator must ensure that the test called is compatible with the current CPU type.

2. By control card input

   To call tests by this method, an input card deck is required, as shown in figure 2-1.

   On the program call card, XXX. is the test mnemonic. The operator must ensure that the test called is compatible with the current CPU type.

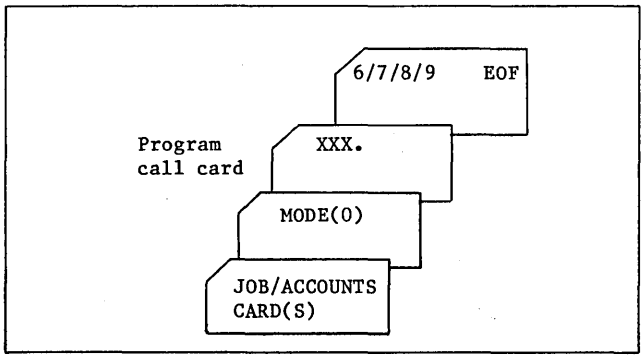

Figure 2-1. Control Card Input Deck

3. By interactive terminal

   To call test programs by this method, follow the routines for normal user jobs. The operator must ensure that the test called is compatible with the current CPU type.

None of these tests has selectable parameter entries.

ALX

ALX is a random instruction-generating program whose primary function is to test the stack registers' ability to handle instructions issued at a rate faster than that possible when not issuing from the stack.

Although the purpose and description of the test have not been changed, certain parameters and important memory locations have been changed, allowing the test to be run in an on-line mode. A detailed description of the test can be obtained from REL2B source listing.

SIGNIFICANT MEMORY LOCATIONS

The following are the significant memory locations for ALX:

Address	Tag Name	Description
440	E.SAVE	Contents of register in exchange package format (fmt) upon entering subroutine E.ERROR.
1730 through 1734	INST→INST+4	Instruction sequence.
1742 through 1761	INREG→INREG+17	Initial register contents.
1762 through 2001	DFANS→DFANS+17	Difference of fast/slow loop answers.
2002 through 2021	FLANS→FLANS+17	Fast loop answers.
2022 through 2041	SLANS→SLANS+17	Slow loop answers.

CMU

CMU is used to test the compare move unit operations that use random descriptors and operands.

SIGNIFICANT MEMORY LOCATIONS

The following are the significant memory locations for CMU:

Address	Tag Name	Description
405	E.SAVE	Contents of register in exchange package format (fmt) upon entering subroutine E.ERROR.
1604	INST1	First CMU instruction.
1606	INST2	Second CMU instruction.
1644 through 1656	HISTAB	History table of CMU instructions executed.
1660	RNOA	Current random number.

CSU

CSU writes into each location a value equal to the location address, from address 1272 up to the field length. Each location is then read and compared with the current test address five times. The test is then repeated using the complement of each address.

Although the purpose and description of the test have not been changed, certain parameters and memory locations have been changed, allowing the test to be run in an on-line mode. A detailed description of this test can be obtained from the REL2B source listing.

SIGNIFICANT MEMORY LOCATIONS

The following are the significant memory locations for CSU:

Address	Tag Name	Description
405	E.SAVE	Contents of register in exchange package format (fmt) upon entering subroutine E.ERROR.
1500	CPM.P05	Bit 0=1 Loop on failure. Bit 4=1 Repeat section. Bit 7=1 Repeat condition. Bit 9=1 Abort error checking. Default is 0.
1501	CPM.P06	Bit 0=1 Section 0. Bit 1=1 Section 1. Bit 2=1 Section 2. Bit 3=1 Section 3. Bit 4=1 Section 4. Default is 37_8.
1502	CPM.P07	If set to zero, use random number supplied by program. If set to nonzero, use this as random number.
1503	CPM.P15	Number of write operations to be performed. Default is 1.
1504	CPM.P16	Number of read operations to be performed. Default is 1.
1505	CPM.P17	xzzz where zzz is delay quantity from 000 to 777_8. If x=1, delay = zzz milliseconds. If x=2, delay = zzz seconds. If x=4, delay = zzz minutes. Default is 1002B.
1511	CMP.ULR	Field length - 1.
1512	CMP.ADR	Failing address.
1513	CPM.RCD	Received data.
1514	CPM.EXD	Expected data.

CTB

CTB is a CPU test that exercises the CPU control and functional units. It generates and executes random instructions and operands, and checks the results against a second set of answers obtained from a simulator using other functional units. CTB requires a field length of $10\ 000_8$ locations.

CONTROL CARD AND CONSOLE CALLS

To call CTB in MSL mode, the following job deck should be used:

```
JOBCARD
MODE,0.
CTB.
6/7/8/9   EOF
```

To call CTB in under DIS, the following commands should be used:

```
X.MODE,0.
X.CTB.
```

SIGNIFICANT MEMORY LOCATIONS

The following significant locations are selectable with program calls under DIS or by assembly only:

Address	Tag Name		Description
405	E.SAVE		Contents of register in exchange package format (fmt) upon entering subroutine E.ERROR.
1500	STOP	0	Ignore stop.
		1	Stop on error (assumed if not looping on error).
		2	Loop on error.
			Default is 0.
1501	AUTO		Vary parameters automatically in accordance with one of three variable parameter tables shown under CT7.
		0	Ignore auto mode.
		1	Vary parameters using first table.
		2	Vary parameters using second table.
		3	Vary parameters using third table.
			Default is 0.
1502	ZIP	0	Normal mode.
		1	ZIP mode.
		2	Compute and store checksums.
		3	Simulator ZIP mode.
			Default is 0.
1503	STACK	0	STACK option not selected.
		Non-0	Run random instruction loop twice, reading instructions from the stack on the second loop. This option deletes all jumps.
			Default is 0.
1504	OPTO	0	Generate result register number randomly.
		Non-0	Set result register number in sequence. For example, first instruction uses X1, second instruction uses X2, and so on. This may generate a different order of conflict.
			Default is 0.
1506	SKIP	0	Start with pass count equal to zero.
		x	Use lower 10 digits of x as a pass count and then generate a new loop.
			Default is 0.
1510	RANNO	0	If in SMM mode, use prestored number; if in sequencer mode, use clock from APR call.
		x	Use number supplied by x.
			Default is 0.
1511	INSTR	0	Use all instructions in random loop.
		1	Select only those instructions in table (located in programming) starting at INSTR+1 to use in the loop.
		2	Delete instructions in table (located in programming) from random instruction loop.
			Default is 0.

ERROR MESSAGES

If an error is detected during execution, a printout is generated containing the following information:

- Date
- Time
- Diagnostic name
- Random instruction loop

- Input operands in exchange package format
- Simulated results in exchange package format
- Machine results in exchange package format
- Error message
- Simulated results
- Machine results
- Result difference
- Pass count
- Error count

Four asterisks (*) are printed alongside any registers where a comparison fails. The error types are determined by the mode and options selected at the time the error occurs.

Error Message	Description
REGISTER ERROR	The output exchange package is different from the simulated exchange package. The failing register(s) is flagged in the output.
LOOP CHECKSUM ERROR	The checksums formed from the simulated loop and the machine loop do not compare. The failing checksums are displayed.
INCR. WRITE ERROR	The checksums formed from the simulated write buffer and the machine buffer do not compare. The failing checksums are displayed.
CONTROL ERROR	A stop instruction (00) is illegally generated by the program because of a hardware malfunction.
MOVE ERROR	An error was detected and the random loop was simplified to all pass instructions. However, since the answers still do not compare, this error message indicates that a load or store error occurred.
ZIP CHECKSUM ERROR	The checksums formed from the machine loop in ZIP mode do not compare with the prestored values. The two values are displayed. Unless stop-before-simplify is selected, the machine goes into normal mode, skips to the same pass count, and attempts to find the problem in normal mode.

NOTE

The input operands, simulated results, and machine results are printed for all error types.

CT3 AND CT8

CT3 and CT8 are CPU tests which test the CPU control and functional units. They generate and execute random instructions and operands, and check the results against a second set of answers obtained from a simulator using other functional units.

The purpose and description of the CT3 test is identical to the version which executes off-line. Certain parameters and memory locations have been changed to accommodate on-line operation. CT3 does not execute on CYBER 170 Models 815, 825, 835, 845, 855, 865, and 875.

The purpose and description of CT8 is identical to CT3. Certain memory locations have been changed to accommodate CYBER 170 Models 815, 825, 835, 845, 855, 865, and 875. CT8 executes only on CYBER 170 Models 815, 825, 835, 845, and 855.

A detailed description of both CT3 and CT8 may be found in the REL2B source listing.

SIGNIFICANT MEMORY LOCATIONS

The following are the significant memory locations for CT3 and CT8. Locations which appear in parentheses are CT8 locations which differ from corresponding CT3 locations.

Address	Tag Name	Description
453	E.SAVE	Contents of register in exchange package format (fmt) upon entering subroutine E.ERROR.
1500	RANNO	Random number starter: 0 Use number supplied by program. x Use x as random number starter. Default is 0.
1501	CMUTAG	0 Use CMU instructions. Non-0 Do not use CMU instructions. Default is 0.
1502	NUMWD	Number of words in the random loop. 0 Use five words. x 1 through 77_8, use x words. Default is 0.

Address	Tag Name		Description
1503	STOP	0	Stop on error (no instruction isolation).
		1	Stop on error (after instruction isolation).
		2	Loop on error.
		negative number	Stop on error when already looping on error.
		Default is 1.	
1504	STACK	0	Out of stack operation.
		Non-0	In stack operation.
		Default is 0.	
1505	SKIP	Pass count from a previously known error.	
		0	Do not use previous pass count.
		x	Use x as previous pass count.
		Default is 0.	
1506	OPTO	Optimize result registers for minimum conflict.	
		0	Ignore optimization.
		1	Use optimization.
		Default is 0.	
1507	NORPT	Number of passes through random loop before creating a new loop.	
		0	One pass.
		x	1 through 77_8, where x is number of passes.
		Default is 0.	
1510	INSTR	Define instructions to be used or not to be used in the loop.	
		0	Ignore this option.
		1	Use only the instructions in the table (located in programming) beginning at INSTR.
		Default is 0.	

Address	Tag Name	Description
2211 (2217)	FAST	Starting address of random loop.
2313 (2321)	ERDIF	Register difference.
2314 (2322)	ERDIF+1	Simulated result.
2315 (2323)	ERDIF+2	Machine result.
2316 (2324)	ERDIF+3	Address of simulated result.
2317 (2325)	ERDIF+4	Address of machine result.
2320 (2326)	ERCO	Error counter.
2323 through 2342 (2331 through 2350)	SANS ⟶ SANS+17B	Simulated results.
2343 through 2362 (2351 through 2370)	SANS+20B ⟶ SANS+37B	Machine results.
2363 through 2402 (2371 through 2410)	ERREX ⟶ ERREX+17B	Simulated results after initial error only.
2403 through 2422 (2411 through 2430)	ERREX+20B ⟶ ERREX+37B	Machine results after initial error only.
2423 through 2442 (2431 through 2450)	DUMEX ⟶ DUMEX+17B	Operands used by random loop.
6000	SOURCE	CMU source buffer area.
11000	RESULT	CMU machine result area.
14000	SIMBUF	CMU simulated result area.
2204 (2212)	ANSDIF	CMU move result difference.

Address	Tag Name	Description
2205 (2213)	K1FIELD	CMU machine move address.
2206 (2214)	K2FIELD	CMU simulator move address.
2207 (2215)	K1RSLT	CMU machine move result.
2210 (2216)	K2RSLT	CMU simulator move result.

CT7

CT7 is a CPU test that tests the CPU control and functional units. It generates and executes random instructions and operands, and checks the results against a second set of answers obtained from a simulator using other functional units.

Although the purpose and description of the test have not been changed, certain parameters and memory locations have been changed, allowing the test to be run in an on-line mode. A detailed description of this test can be obtained from the REL2B source listing.

SIGNIFICANT MEMORY LOCATIONS

The following are the significant memory locations for CT7:

Address	Tag Name	Description
405	E.SAVE	Contents of register in exchange package format (fmt) upon entering subroutine E.ERROR.
1500	STOP	0 Stop on error after loop shortening. 1 Stop on error before loop shortening. 2 Loop on error. 3 Stop on error after looping on error. Default is 0.
1501	AUTO	Vary parameters automatically in accordance with tables under Variable Parameters Tables. 0 Ignore auto mode. 1 Vary parameters using first table (refer to table 2-1). 2 Vary parameters using second table (refer to table 2-2). 3 Vary parameters using third table. (refer to table 2-3). Default is 0.
1502	ZIP	0 Ignore ZIP mode. Non-0 Do checksums of registers on every pass. Accumulate results and compare with a predetermined result every $10\ 000\ 000_8$ passes. Default is 0.
1503	STACK	0 Out of stack operation. Non-0 Run random loop twice. On the second time, read instructions from the stack. Default is 0.
1504	OPTO	Optimize registers for minimum conflict. 0 Ignore optimization. Non-0 Optimize registers. Default is 0.
1505	NUMWD	Number of words in random loop. 0 Use five words. Non-0 Use only lower six words. Default is 0.
1506	SKIP	Pass count from a previously known error. 0 Do not use a previous pass count. x Lower 30 bits only, where x is previous pass count. Default is 0.
1507	NORPT	Number of passes to be made through random loop before creating a new one. 0 One pass only. x Lower six bits, where x is number of passes. Default is 0.

Address	Tag Name	Description
1510	RANNO	Random number starter. 0 Number supplied by program. x Use x as random number starter. Default is 0.
1511	INSTR	Define instructions to be used in the loop. 0 Ignore this option. 1 Use only the instructions in the table (located in programming) beginning at address INSTR. 2 Use any instructions except those in the table (located in programming) beginning at address INSTR. Default is 0.
2111	FAST	Starting address of random loop.
2256	ERDIF	Register difference.
2257	ERDIF+1	Simulated result.
2260	ERDIF+2	Machine result.
2261	ERDIF+3	Address of simulated result.
2262	ERDIF+4	Address of machine result.
2263	ERCO	Error counter.
2275 through 2314	SANS ⟶ SANS+17B	Simulated results.
2315 through 2334	SANS+20 ⟶ SANS+37B	Machine results.
2335 through 2357	DUMEX ⟶ DUMEX+17B	Operands used by random loop.

VARIABLE PARAMETERS TABLES

The following are the parameter sequence tables used with the AUTO sequence.

Table 2-1. Parameter Sequence I

Sequence	Stack	OPTO	ZIP	Number of Words (Octal)	Number of Passes (Octal)
1	0	0	0	5	10 000
2	SET	0	0	11	20 000
3	0	SET	0	12	20 000
4	0	0	0	3	10 000
5	0	0	SET	5	10 000
6	0	0	0	77	20 000
7	SET	SET	0	11	20 000
8	0	0	0	4	10 000

Table 2-2. Parameter Sequence II

Sequence	Stack	OPTO	ZIP	Number of Words (Octal)	Number of Passes (Octal)
1	0	0	0	12	200 000
2	SET	0	0	11	200 000
3	0	SET	0	12	200 000
4	0	0	0	6	10 000
5	0	0	SET	5	30 000
6	0	0	0	77	100 000
7	SET	SET	0	11	400 000
8	0	0	0	7	10 000

Table 2-3. Parameter Sequence III

Sequence	Stack	OPTO	ZIP	Number of Words (Octal)	Number of Passes (Octal)
1	0	0	0	5	10 000
2	0	0	0	2	10 000
3	0	0	0	12	10 000
4	0	0	0	77	10 000
5	SET	0	0	11	10 000
6	0	SET	0	5	10 000
7	SET	SET	0	11	10 000
8	0	SET	0	12	10 000

CU1

CU1 tests the control hardware and functional units of the central processor.

Although the purpose and description of the test have not been changed, certain parameters and the contents of some memory locations have been changed, allowing the test to be run in an on-line mode. A detailed description of this test can be obtained from the REL2B source listing.

SIGNIFICANT MEMORY LOCATIONS

The following are the significant memory locations for CU1:

Address	Tag Name	Description
405	E.SAVE	Contents of register in exchange package format (fmt) upon entering subroutine E.ERROR.

All pertinent information for this test can be obtained from the A, B, and X registers. The contents of these registers are saved by subroutine E.ERROR of CPUMEM. Failing P register addresses are displayed by CPUMEM.

CU8

CU8 is similar to the CU1 test. CU8 is used with CYBER 170 Models 815, 825, 835, 845, and 855. A detailed description of CU8 may be found in the REL2B source listing.

EC3

EC3 is a test for ECS written specifically for diagnosing and repairing ECS problems on-line under the host operating system. EC3 runs on the system as a user job and tests ECS by writing and reading various selectable data patterns in the system's user direct access area of ECS.

The test runs one pass and exits. It is possible, by selecting various parameters, to create different testing conditions. Parameter selection is accomplished in either of two ways:

- Passing the parameters to the test through control card call.
- Causing the test to pause and entering the parameters through DIS.

EC3 may be run selecting all of ECS, the maintenance portion of ECS, or only the preallocated part of ECS, by selecting ALL, HALF, or PRE, respectively.

To preallocate ECS, bits 8 through 11 of word 12 of the deadstart panel must be set to define the size. Refer to the NOS Operator's Guide for CMRDECK contents.

If the parameters are passed through the control card, they can appear in any order. The only restriction on the control card is that all numbers are octal with a maximum of seven digits. The test pauses if called as EC3(PAUSE).

The parameter list, within the program, is located at RA+1500$_8$. Each of the parameter list elements contains the parameter name (left-justified) and the default quantity (right-justified). When entering a value into the parameter list, it is not necessary to retain the parameter name, since the names are deleted as soon as the test begins. Within the parameter list is a cell containing HRAN NEXT. The following cell (HRAN) is used to hold the random number for the test sections that use random data. A value may be placed in HRAN, through DIS, to recreate a block of random numbers.

Any time the test pauses (pause parameter selected, or stop on error selected), it can be restarted by typing GO.

If an ECS error occurs during execution of the test, the test terminates with an abort. This allows the user the option of killing the output file if no error occurred. On termination, the test does not wait for an operation GO, if stop on error is not selected.

CONTROL CARD CALL

The following is the format for control card call:

EC3(P1,P2,P3,...,Pn)

If the parameter is not specified, default is used. If specified but not equivalent, it is set to 1 (nonzero) for all parameters. There is no restriction on the order in which parameters appear on the control card. All numeric values must be octal.

Parameter	Description
ALL	Select all of ECS (no user ECFL allowed; EST must show ECS down). Default = 0 (not selected).
CKSUM=	Checksum data on every nth pass. Default = 0.
DATA=	Check data on every nth pass. Default = 0.
DISP=	Display end-of-section (EOS) message. Default = 0 (not selected).
HALF	Select maintenance half of ECS (no user ECFL allowed; EST must show ECS in half mode). Default = 0 (not selected).
IGNWA=	Ignore write abort flag. Default = 0 (not selected). Valid only if *WS* (write only) is selected.
LIST=	Line count for dayfile listing. Default = assembly constant *LINES*.
LOOP=	Loop on error. Default = 0 (not selected).
PASS=	Pass count for repeat section or repeat test. Default = 0.
PAUSE=	Pause at start of test to set parameters. Default = 0 (not selected).
PRE	Select preallocated area of ECS (allocated by NOS). Default = 0 (not selected).
REP=	Repeat count for high-speed read. Default = 3.
RETAIN=	Retain last random pattern. Default = 0 (not selected).
RS=	Read only. Default = 0 (not selected). Parameter must be alphabetic.
SECT=	Repeat section selection. Default = 0 (none selected). Parameter must be alphabetic. Overrides WS or RS parameters.
SELECT=	Select sections. Default = 7777 (all sections selected).
	Bit 0 = A — Write and read zeros pattern.
	Bit 1 = B — Write and read ones pattern.
	Bit 2 = C — Alternating ECS words of ones and zeros.
	Bit 3 = D — Alternating ECS words of zeros and ones.
	Bit 4 = E — Write and read ECS word lines of 5-2 pattern.
	Bit 5 = F — Write and read ECS word lines of 2-5 pattern.
	Bit 6 = G — Write and read even parity pattern.
	Bit 7 = H — Write and read odd parity pattern.
	Bit 8 = I — Write and read ECS in 16-word blocks, incrementing through all CM banks and all ECS words.

NOTE

Section I does not have the capability to do a repeat write only or read only (WS and RS parameters).

	Bit 9 = J — Write and read random pattern.
	Bit 10 = K — Write and repeat read with random pattern.
	Bit 11 = L — Write and repeat read with all random parameters.
STOP=	Stop on error. Default = 1 (selected).
WC=	Word count for write and read of ECS. Default = assembly constant SIZE.
WS=	Do write only. Default = 0 (none selected). Parameter must be alphabetic.

SAMPLE JOBS

The following are sample job cards for EC3:

- EC3,PASS=10.

  Run eight passes of the entire test.

- EC3,SECT=J,DATA=4,STOP.

  Execute section J only, checking data every fourth pass, and stop on error. It runs until time limit or until dropped.

- EC3,SECT=J,DATA=10,CKSUM=20.

  Execute section J only, checking data every eighth pass, and checksumming data every 16th pass. It runs until time limit or until dropped.

- EC3(SECT=J,DATA=10,CKSUM=20,PASS=1000).

  Same as job 3, but test exits after 1000 passes.

- EC3,WS=B,WC=100,IGNWA,LIST=0,PAUSE.

  Write all ones (section B) to ECS in 100-word blocks. Do not abort test if a write abort occurs. Do not list any errors. Pause at start of test.

- EC3(SECT=K,REP=5,PASS=400).

  Execute 400 passes of section K, but repeat read five times.

- EC3,PASS=20,SELECT=7000.

  Execute 16 passes of sections J, K, and L only.

- EC3,SECT=I,HALF,DISP.

  Execute section I, using only the maintenance half of ECS. Display the end-of-section message.

Tests may also be run under DIS as follows:

```
N.DIS
ENFE,40000.
EC3,SECT=K,REP=5,PASS=400.
DROP.
```

DAYFILE MESSAGES

The following are dayfile messages issued by EC3:

INVALID INPUT PARAMETER - TEST ABORTED

An invalid parameter is found. The test aborts if one or more of the following conditions occur:

SECT, WS, or RS request nonexistent section.

Repeat count greater than five.

Chassis count equal to zero.

DMPX from abort contains the parameter list.

ECS FL EQUALS ZERO - TEST ABORTED

EC3 TESTING ECS FL OF XXXXX

EC3 USING EC3 ABSOLUTE ADDRESS OF XXXX5

The ECS absolute address output is provided at the beginning of the program. The ECS absolute address is compared and checked if still current (ECS addressing not moved by the system) each time an output error message occurs. If the ECS address has been moved by the system, EC3 outputs a new ECS absolute address message before the original error message.

It is possible for the operating system to move the ECS boundaries between the time EC3 detects an error and the time EC3 determines the current absolute ECS address from the system.

END OF SECTION X (TO B DISPLAY ONLY)

HRAN TO LAST BLOCK XXXXXXXXXXXXXXX
XXXXXXXXX

The initial random number of the last block read is printed in case of a read abort.

READ ABORT S=X
CMSA=XXXXX
A=XXXXXXXX

This message is also used to report WRITE ABT, HSREAD ABT, and CKSUM ERR. The HSREAD ABT message contains the repeat count and an indication of the failing read. S is the section name. CMSA is the start address in CM of the last block. A is the start address (relative) in ECS.

DATA COMPR WRITE
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
A=XXXXXXXX S=X READ
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

The data compare error shows the data written, data read, the current selection name (S), and the address (relative), within ECS, of the failure (A).

MORE THAN ONE - PRE, HALF, OR ALL - SELECTED

PRE, *HALF*, and *ALL* are mutually exclusive; only one may be selected.

CEVAL PERMISSION NOT GRANTED

NOS will not allow EC3 to run; refer to *ALL*, *HALF*, and *PRE* for constraints.

EC8

EC8 is similar to the EC3 test. EC8 is used with CYBER 170 Models 815, 825, 835, 845, and 855. A detailed description of EC3 may be found in the REL2B source listing.

FM2

FM2 is a floating multiply test designed to provide a unit-to-unit comparison test of the multiply function in a 6600 system. It generates two random binary numbers and uses them to test unit 1 against unit 2 in single, double, and rounded precision modes.

Although the purpose and description of the test have not been changed, certain parameters and the contents of some memory locations have been changed, allowing the test to be run in an on-line mode. A detailed description of this test can be obtained from the REL2B source listing.

SIGNIFICANT MEMORY LOCATIONS

The following are the significant memory locations for FM2:

Address	Tag Name	Description
405	E.SAVE	Contents of register in exchange package format (fmt) upon entering subroutine E.ERROR.

All pertinent information for this test can be obtained from the A, B, and X registers. The contents of these registers are saved by the subroutine E.ERROR of CPUMEM. Failing P register addresses are displayed by CPUMEM.

FST AND FS8

FST and FS8 are random number instruction checking and test generating, and checking routines which are capable of 347 362_8 passes in 10000_8 s.

The purpose and description of FST is identical to the version which executes off-line. Certain parameters and memory locations have been changed to accommodate on-line operation. FST does not execute on CYBER 170 Models 815, 825, 835, 845, 855, 865, and 875.

The purpose and description of FS8 is identical to FST. Certain memory locations have been changed to accommodate CYBER 170 Models 815, 825, 835, 845, 855, 865, and 875. FS8 executes only on CYBER 170 Models 815, 825, 835, 845, 855, 865, and 875.

A detailed description of both FST and FS8 may be found in the REL2B source listing.

SIGNIFICANT MEMORY LOCATIONS

The following are the significant memory locations for FST and FS8. Locations which appear in parentheses are FS8 locations which differ from corresponding FST locations.

Address	Tag Name	Description
457	E.SAVE	Contents of register in exchange package format (fmt) upon entering subroutine E.ERROR.
1737 (1743)	FAST	Fast loop instruction sequence.
2117 (2123)	SLOW	Slow loop instruction sequence.
1757 (1763)	ERDIF	Error difference (slow minus fast).
1760 (1764)	ERSL	Slow loop results.
1761 (1765)	ERSF	Fast loop results.
1762 (1766)	ERSLA	Address of slow loop result.
1763 (1767)	ERSFA	Address of fast loop result.
1764 (1770)	ERCO	Error counter.
1776 through 2005 (1772 through 2011)	SANS ⟶ SANS+17	Register contents after slow loop, saved after every run (failure or not).
2006 through 2025 (2012 through 2031)	SANS+20 ⟶ SANS+37	Register contents after fast loop, saved after every run (failure or not).
2026 through 2045 (2032 through 2051)	SAVE ⟶ SAVE+17	Register contents after slow loop, saved only after failure.
2046 through 2065 (2052 through 2071)	SAVE+20 ⟶ SAVE+37	Register contents after fast loop, saved only after failure.

IWS

IWS tests the ability of the central processor to execute no-operation, Boolean, and stack jump instructions that have been organized into stacks. IWS also checks for correct translation of single 15- and 30-bit instructions in all parcels of all stack words.

Although the purpose and description of the test have not been changed, certain parameters and the contents of some memory locations have been changed, allowing the test to be run in an on-line mode. A detailed description of this test can be obtained from the REL2B source listing.

NOTE

Section 3 of this test is not executed when running on-line, because it is impossible to determine when an exchange takes place.

SIGNIFICANT MEMORY LOCATIONS

The following are the significant memory locations for IWS:

Address	Tag Name	Description
405	E.SAVE	Contents of register in exchange package format (fmt) upon entering subroutine E.ERROR.

All pertinent information for this test can be obtained from the A, B, and X registers. The contents of these registers are saved by the subroutine E.ERROR of CPUMEM. Failing P register addresses are displayed by CPUMEM.

LCM

LCM is a CPU test of the large core memory. It tests a field of memory specified by a starting address and a final address.

LCM is derived from the quick-look test sections of the 7000 diagnostic LCM4. It requires a field length of 5000_8 locations.

CONTROL CARD AND CONSOLE CALLS

To call LCM in infinite pass count, the following card deck should be used:

```
JOB CARD        Including EC parameter
LCM.
6/7/8/9         EOF
```

To call LCM in under DIS, the following commands should be used:

```
ENFE,(LCM TO TEST)
X.LCM.
```

TEST SECTIONS

The following test sections are selected using the TESTBL parameter.

Test Section	Description
QSEC1	Write/read eight basic test patterns with a large block count, using the block write/read instructions.
QSEC2	Write/read a floating-one and -zero bit pattern, using the block write/read instructions.
QSEC3	Write/read a true checkerboard pattern, using the block and direct write/read instructions.
QSEC4	Direct write/read an all-zeros pattern, into the memory initially set to all ones.
QSEC5	Write/read a floating-one bit test pattern, using the LCM direct instructions.
QSEC6	Write/read a floating-one bit test pattern and its complement, using the LCM direct instructions.

SIGNIFICANT MEMORY LOCATIONS

The following are the significant memory locations for LCM:

Tag Name	Description
BNKST	First word address. Default is 0.
BNKLMT	Last word address determined by use of system memory macro.
TSTCNTRL	Test control. 1 Stop on error. 2 Repeat current section. Default is 1.
TESTBL	Up to six different tests are selected or deselected by entering or clearing, respectively, the entry address in the following table. Default is QSEC1 QSEC2 QSEC3 QSEC4 QSEC5 QSEC6
SECTN	Current section number.
ERRCODE	Error code.
ERRCOUNT	Error count.
LFDATA	Actual data.
EXDATA	Expected data.
BNKST	First word address.
BNKLMT	Last word address minus one.
TSTCNTROL	Test control.

ERROR MESSAGES

If LCM detects an error, it makes an appropriate entry in the dayfile that includes:

- Test name
- Section number
- Message
- Pass count

- ECRA (as detected at error time)
- CPU diagnostic/LCM failed

The following error messages are displayed and entered in the output file for printing:

```
A DATA ERROR WITH NO PARITY ERROR

RELATIVE ADDRESS xxxxxxxxxx

EXPECTED xxxxxxxxxxxxxxxxxxx

RECEIVED xxxxxxxxxxxxxxxxxxx

AN ERROR EXIT ON A BLOCK WRITE.

A DATA ERROR WITH PARITY ERROR

RELATIVE ADDRESS xxxxxxxxxx

EXPECTED xxxxxxxxxxxxxxxxxxx

RECEIVED xxxxxxxxxxxxxxxxxxx

A PARITY ERROR WITH NO DATA ERROR

READING xxxxxxxxxx WORDS FROM

ADDRESS xxxxxxxxxx
```

If the test is called with no extended memory selected, a message in the dayfile reads:

```
NO EXTENDED MEMORY SELECTED /LCM/

ABORTED - - INVALID INPUT PARAMETER.
```

MDD

The Monitor Display Driver (MDD) is initiated from the system operator's console and supports system analysis through either local or remote access via the CYBER 815, 825, 835, 845, or 855 two-port multiplexer interface. Once initiated, MDD detaches itself from the operating system until directed by command via the MDD console (by default the terminal connected to port 0 on the two-port multiplexer) to terminate. Since MDD does not depend on the operating system after initialization, it is generally still operating if the operating system malfunctions due to hardware or software problems.

MDD provides commands from the MDD console to display and change central memory (60 and 64 bit) and to display, interpret, and change the contents of the maintenance registers present on the CYBER 170 Models 815, 825, 835, 845, and 855. The commands can be entered at any time MDD is up. Using any of the display commands should have no effect on the operating system.

Most MDD commands allow position independent parameters, keywords, or a combination of both. This means that for the command whose syntax is

```
ER element RN=register number RV=register value
```

where element can be one of keywords I or M or P, all of the following do the same thing:

```
ER I RN=30 RV=9
ER I 30 9
ER RV=9 RN=30 I
ER RN=30 I 9
```

The delimiter between parameters in MDD can be either a space or a comma.

Parameters can be placed positionally with multiple commas. All of these examples below are correct.

```
ER M RN=A0 RV=C0015732
ER M Q0 C0015732
ER,M,A0,C0015732
ER,M,RN=A0,RV=C0015732
DR,,,C0015732
```

MDD can be initiated on either port 1 or port 0 of the two-port mux. In addition, when MDD is initiated, it waits 15 min for a Carrier ON status to be detected. If it is not, MDD terminates.

MDD is initiated, after ensuring that the console is unlocked, through operator entry of the following DSD command:

```
X.MDD(d,p)
```

Parameters d and p are optional; if specified, d must be set to any nonzero octal digit (1 through 7). If d is not specified, MDD waits 15 min for a Carrier ON status and then drops if none is detected. This is used only for a dial-up connection.

If parameter p is a one (1), port 0 of the two-port mux is used. If p is a two (2), port 1 is used. Port 1 is used if p is not specified.

MDD COMMAND EXPLANATION AND SYNTAX

Central Memory Dump Commands

All central memory dump commands save the address and word count parameters used. The next central memory dump command obtains the default values for the address and word count parameters from these values. The initial word count default is 10 and the address is 0.

DB Display Bytes

The DB command displays 64-bit memory in byte format (one word per line, eight groups of two hexadecimal digits per word). The address is a hexadecimal byte address and the word count is a hexadecimal word count.

Format:

```
DB AD=byte address  WC=word count
```

Keyword	Parameter Value	Description
AD =	byte address	Starting byte address for the dump (hexadecimal). This value is truncated to the nearest word boundary. Defaults are to the address used by the most recent memory command.
WC =	word count	Number of words to display (hexadecimal). The default is the previously used word count.

DC Display Octal CM

The DC command displays 60-bit memory in octal word format (one word per line, 20 octal digits per word). The address is an octal word address and the word count is in octal.

Format:

DC AD=octal address WC=word count

Keyword	Parameter Value	Description
AD =	octal address	Starting word address for the dump (octal). This parameter defaults to the address used by the most recent memory command.
WC =	word count	Number of words to display (octal). The default is the previously used word count.

DH Display Hexadecimal CM

The DH command displays 64-bit central memory in hexadecimal word format (one word per line, 16 hexadecimal digits per word). The address is a hexadecimal word address.

Format:

DH AD=word address WC=word count

Keyword	Parameter Value	Description
AD =	word address	Hexadecimal starting word address of the dump. Defaults are to the address used by the most recent memory command.
WC =	word count	Hexadecimal count of words to dump. The default is the word count used in the previous memory display command.

DM Display Virtual Memory

The DM command can be used to display memory in a virtual memory environment. When MDD is first initiated, MDD initializes all of its default virtual memory parameters from the hardware registers. These values may be changed if desired. The memory is displayed by first printing the segment number and then displaying memory in byte format (eight groups of two hexadecimal digits) with an eight-digit relative byte offset for an address. The Real Memory Address (RMA) for the first word printed is saved to be used as the default for the DB, DC, and DH commands. This provides a way to determine the RMA for an arbitrary Process Virtual Address (PVA). The PVA entered is kept as the default PVA if that parameter is not entered the next time the DM command is used.

Format:

DM PVA=virtual address WC=word count
xp=hexadecimal address of exchange package ..
PT=page table rma BO=byte offset
PS=page size mask .. PL=page table length

Keyword	Parameter Value	Description
PVA =	virtual address	Process virtual address to use as the starting memory address for the display. This is an 11-digit hexadecimal number consisting of three digits of segment number and eight digits of byte offset.
WC =	word count	Number of words to display (hexadecimal). The default is the previously used word count.
XP=MPS	monitor process state	Exchange package address used to obtain the segment table address used in converting a PVA to an System Virtual Address (SVA) prior to searching the page table. Just specifying the keyword uses the value last specified for that keyword. Specifying xp=hexadecimal address, changes the value associated with the keyword 'xp'. If this parameter is omitted, the keyword used on the last DM command is assumed.
JPS	Job process state	
XPS	Extra exchange package address	
PT =	page table rma	Page table address. The default is the value used on the previous DM command. The initial default is the page table address when the operating system started up.

Keyword	Parameter Value	Description
BO =	byte offset	This parameter can be used to display a different byte offset of the same segment. If this is used, the PVA parameter should not be used.
PS =	page size mask	Hexadecimal number for the new page size mask. The default is the value used on the previous DM command. The initial default is the page size mask when the operating system started up.
PL =	page table length	Hexadecimal number for the new page table length. The default is the value used for the previous DM command. The initial default is the page table length when the operating system started up.

'+' Display Next CM Block

The '+' repeats the last command, if it was a CM display command, using a new starting address computed one of two ways. If no parameter is given then the CM displayed starts where the previous display left off. If the increment is specified then the starting address for the memory display is equal to the current starting address plus increment.

Format:

+ increment

Parameter Value	Description
increment	Optional starting address increment when the CM display command is repeated. If not specified, the next display starts where the previous left off. Increment is interpreted in the same manner as the word or byte count of the previous memory dump command.

'-' Display Previous CM Block

The '-' repeats the last command, if it was a CM display command, using a new starting address computed one of two ways. If no parameter is given then the CM block displayed ends where the previous block started. If the decrement is specified then the starting address for the memory display is equal to the current starting address minus decrement.

Format:

- decrement

Parameter Value	Description
decrement	Optional starting address decrement when the DM display command is repeated. If not specified, the next display ends where the previous started. Decrement is interpreted in the same manner as the word or byte count of the previous memory dump command.

Central Memory Change Commands

The following commands will change central memory. No checking of any kind is done on the addresses. Attempts to write beyond the IOU operating system (OS) bounds address or the memory bounds address stops this PP unless these registers are cleared.

CAUTION

Use care when using these commands to change central memory.

EB Enter Bytes into Memory

The EB command changes memory a byte at a time for as many bytes as provided on the command. The address is an exact byte address and as few as one byte may be changed. The address must be specified. The last address changed is remembered for successive memory display commands.

Format:

EB AD=byte address B1 B2 B3 .. Bn

Keyword	Parameter Value	Description
AD =	byte address	Starting byte address (hexadecimal) to be changed. This is an exact byte addresss and need not be on a word boundary. This parameter is required.
	B1 .. Bn	The new values for the bytes starting at the byte position given in the AD parameter. Each number is a one or two digit hexadecimal number. One or more bytes may be changed at one time.

EC Enter Central Memory

The EC command changes one word of 60-bit memory to the specified octal value.

Format:

```
EC  AD=word address  WV=word value
```

Keyword	Parameter Value	Description
AD =	word address	Word address (octal) to change. This parameter is required.
WV =	word value	The new value to be entered into the address given by AD. This is a 1- to 20-digit octal value. The default for this parameter is 0.

Maintenance Register Display Commands

DR Display Maintenance Registers

The DR command displays either a single maintenance register or a list of registers in a specific element (IOU, Memory, Processor). The display consists of the register number in hexadecimal, the contents of the register in 16 hexadecimal digits, and a description of the contents (list option only).

Format:

```
DR  element  RN=register number
```

Keyword	Parameter Value	Description
	element	Identifies the element from which to read the registers. The only valid keys are: I = IOU, M = Memory, and P = Processor.
RN =	register number	Hexadecimal register number to display. If this parameter is omitted then a predefined list based on the element is used.

The initial default element is I, the IOU. Once another value is specified for this parameter, it becomes the new default.

The default registers displayed depend on which element is specified. For the IOU, the following registers are displayed and labeled accordingly:

00	SS
12	OI
18	MASK REGISTER
21	OS BOUNDS
30	EC
40	STATUS
80	FS1
81	FS2
A0	TM

For the Memory, the default registers are:

00	SS
12	OI
20	EC
21	MEM BOUNDS
A0	CEL
A4	UEL1
A8	UEL2

For the Processor, the default registers vary with the mainframe type. For all mainframes the following are displayed:

00	SS
30	DEC
31	S
40	P
41	MPS
42	MCR
43	UCR
48	PTA
51	MDR
61	JPS
62	SIT
80	PFS

In addition, the following registers are displayed for a series:

815, 825 CPU

93	MCEL

835 CPU

81	PFS1
92	CCEL
93	MCEL

845, 855 CPU

81	PFS1
82	PFS2
83	PFS3
84	PFS4
85	PFS5
86	PFS6
87	PFS7
88	PFS8
89	PFS9

ER Enter New Maintenance Register Value

The ER command allows you to change the value of a register in an element on the maintenance channel.

CAUTION

Not all registers can be safely changed while an operating system is up and not all registers can be written by MDD for hardware reasons.

Format:

```
ER  element  RN=register number  RV=register value
```

Keyword	Parameter Value	Description
	element	Identifies the element where the register to be changed resides. The only valid keys are: I = IOU, M = Memory, and P = Processor.
RN =	register number	Register number to be written (hexadecimal).
RV =	register value	Value to write into the register. This may be a 1- to 16-digit hexadecimal number. The value is to be written to the register right adjusted in 64 bits.

CE Clear Errors On Maintenance Element

The CE command clears errors on the specified element.

Format:

CE element

Parameter Value	Description
element	The element for which the Clear Errors function is executed. The only valid keys are: I = IOU, M = Memory, and P = Processor.

MCR Interpret MCR Bit Settings

The MCR command gives a brief description of each bit set in the provided MCR value.

Format:

MCR RV=mcr value

Keyword	Parameter Value	Description
RV	mcr value	The MCR value to be interpreted. If not specified, the current MCR register is read and its value used.

UCR Interpret UCR Bit Settings

The UCR command gives a brief description of each bit set in the provided UCR value.

Format:

UCR RV=ucr value

Keyword	Parameter Value	Description
RV	ucr value	The UCR value to be interpreted. If not specified, then the current UCR register is read and its value used.

CI Clear Maintenance Register Interlock

The CI command attempts to clear the maintenance channel interlock. This should only be used as a last resort to read or write the maintenance registers when some other PP program has failed with the maintenance channel interlock set.

Format:

CI

DP Display PP Register Values

The DP command displays the selected register for all the PPs in the machine. Twenty PPs are assumed. The registers are displayed as six-digit octal numbers, 10 to a line. The first line displays PPs 0 to 11, the second line displays PPs 20 to 31.

Format:

DP register

Parameter Value	Description
register	Identifies the PP register to be displayed for each PP. The valid keys are P = Program counter, Q = Q register, K = Current instruction, and A = Address register.

IP Idle PP

The IP command idles the selected PP by doing a hardware idle on the PP. Once idled, the PP can only be restarted by doing a RP and specifying the new address. The contents of the A register are destroyed.

Format:

IP PP=pp number

Keyword	Parameter Value	Description
PP =	PP number	PP number to idle. Must be an octal number from 0 to 11, or 20 to 31. This parameter is required.

CAUTION

Do not idle the PP that MDD is in.

RP Run PP Starting at Address

The RP command restarts the PP by deadstarting the PP and sending over a new P register.

CAUTION

The deadstart load destroys the contents of words 0 and 1 of the PP A register.

Format:

RP PP=pp number AD=starting address

Keyword	Parameter Value	Description
PP =	pp number	PP number to start at the new address. Must be an octal number from 0 to 11, or 20 to 31. This parameter is required.
AD =	starting address	Starting address for the PP. Octal number from 0 to 7776.

HP Halt Processor

The HP command unconditionally halts the processor (CPU).

CAUTION

In several CYBER 170-8xx mainframes, many maintenance registers cannot be read if the processor is halted. Contact a customer engineer for information on which registers cannot be altered.

Format:

HP

SP Start Processor

The SP command attempts to restart the CPU by restarting the microcode. Depending on the reason the processor halted, this may or may not be successful. If the processor was halted by the HP command it should always be possible to restart it by the SP command.

Format:

SP

RF Display Register File

The RF command displays one or more consecutive words of the Register File. Note that the CPU must be halted for this command to work on an CYBER 170 Model 845 or 855 machine, and that it cannot be successfully restarted once the RF command is issued. The CPU must be running on all other CYBER Model 815, 825, or 835 mainframes for the RF command to work. If the processor is in the wrong mode when this command is issued, the message:

PROCESSOR IN WRONG MODE

is displayed.

Format:

RF AD=hexadecimal address WC=word count

Keyword	Parameter Value	Description
AD =	hexadecimal address	Hexadecimal starting word address in the Register File. Default is the value given the last time this command was used. The initial default is 0.
WC =	word count	Hexadecimal count of words to display. The default is the last value given when this command was used. The initial default is A, (10 decimal).

Miscellaneous Commands

SR Set Refresh Mode

MDD can be run with refresh mode if running on a CRT with a reset to top of page function. In refresh mode, MDD periodically outputs a control Y to reset to the top of screen and repeats the last display command. Between commands a control X is output to clear the screen. If a 75x terminal is being used, the terminal must be in page mode to use refresh. Refresh mode allows you to watch a register changing or watch an area in memory for the changing hexadecimal or ASCII patterns. A typical use might be to display the trap area in the Environment Interface.

Format:

SR mode

Parameter Value	Description
mode	ON = turn refresh mode on. OFF = turn refresh mode off.

BYE Return MDD PP to Operating System

The BYE command causes MDD to give up the two-port multiplexer and returns the PP to the operating system. MDD must be restarted from the operating system console.

Format:

BYE

When the BYE command is executed, MDD writes a status byte into central memory. Five bits are currently used, as follows:

Bit	Meaning
0	Set if MDD was unable to access the maintenance channel.
1	Set if MDD was unable to access the channel to the two-port mux.
2	Set if MDD was used to write into a Maintenance Register.
3	Set if MDD was used to write into Central Memory.
4	Set if MDD was executed from CTI.

This byte is written into bits 48 through 37 of word INWL in central memory.

ERROR MESSAGES

Whenever an improper command or parameter is given to MDD, it responds with *ILL*, and ignores the command in question.

MDD shares the Maintenance Channel with other utilities. If this channel should be hung and MDD is unable to access it, the message: CHANNEL 17 HUNG appears. When this happens, the DR, ER, and RF commands no longer function correctly. The CI command can be used to clear the channel, but may have adverse side effects depending on why it was hung.

DAYFILE MESSAGES

When MDD is first initialized, the following message is placed in the system dayfile:

MDD - COPYRIGHT 1983 CONTROL DATA

If the console is locked when MDD is initialized, the following message is put in the system dayfile:

CONSOLE MUST BE UNLOCKED FOR MDD

If the user attempts to bring up MDD on a non-170-8xx mainframe, except for 865 and 875, the following message appears in the system dayfile:

MDD NOT ALLOWED ON THIS MAINFRAME

If no connection is made to MDD in 15 min, the following message appears in the dayfile, and MDD drops out:

MDD IGNORED

If the port MDD tries to access is already reserved, MDD drops out and the following message appears in the dayfile:

MUX NOT FREE

EXAMPLE OF MDD COMMAND USAGE

Figure 2-2 shows example of MDD command usage.

```
dr  i                                                   Display default set of
00  0000000000000000  SS                                registers for the IOU.
12  0000FFFAFFFF0F07  OI
18  0000000000000000  MASK REGISTER
21  1F1F1F1F000007FE  OS BOUNDS
30  0000000000000009  EC
40  0000000000008800  STATUS
80  0000000000000000  FS1
81  0000000000000000  FS2
A0  0000000000000000  TM

dr  m                                                   Display default set of
00  0000000000000000  SS                                registers for the memory.
12  0041000000000000  OI
20  0100000002000000  EC
21  400000000BF60000  MEM BOUNDS
A0  0000000000000000  CEL
A4  0000000000000000  UEL1
A8  0000000000000000  UEL2

dr  p 61                                                Display register 61 of
61  0000000000FF8270                                    the processor registers.

dm  pva=500000000 mps wv=5                              Display the first five
    SEGMENT005                                          words of EI.
00000000  00  00  18  00  00  10  81  09
00000008  90  00  00  00  09  09  19  83
00000010  00  00  03  40  00  00  00  18  @
00000018  00  00  00  02  00  FF  89  D0      P
00000020  8D  01  00  4A  0E  12  AC  0F  J   ,

db                                                      Display with DB command.

FFA000  00  00  18  00  00  10  81  09
FFA008  90  00  00  00  09  09  19  83
FFA010  00  00  03  40  00  00  00  18  @
FFA018  00  00  00  02  00  FF  89  D0      P
FFA020  8D  01  00  4A  0E  12  AC  0F  J   ,

dc                                                      Display same block in 60-
07772000  0000600000004100411  F    DHDI                bit octal words.
07772001  0000000001102214601      IBQ A
07772002  0000064000000000030          X
07772003  0000001000777704720   H     P
07772004  6401000450160452601 7 A D ND   0

dh                                                      Display the same block
1FF400  0000180000108109                                with DH command.  Note
1FF408  9000000009091983                                the dump is addressed by
1FF410  0000034000000018  @                             word instead of by byte.
1FF418  0000000200FF89D0      P
1FF420  8D01004A0E12AC0F  J   ,

dh,17EF65,4                                             Display four words of
17EF65  00001000000A0870      P                         memory in hexadecimal
17EF66  0100100100000B28      (                         word format displaying
17EF67  00FF100100000B28     (                          the address as a hexa-
17EF68  FE00FFFF80000000  ~                             decimal address.

+                                                       Advance the display to
17EF69  00001E070000F7C0      w@                        the next block of CM.
17EF6A  0000100000010200
17EF6B  0000100000000180
17EF6C  0000100000000000

db wc=1 ad=bf7b28                                       Display one word in byte
BF7B28  00  00  10  00  00  0A  08  70      p           format.

+30                                                     Advance the display by
BF7B58  00  00  10  00  00  00  01  80                  30 (hexadecimal) bytes.
```

Figure 2-2. Examples of MDD Command Usage (Sheet 1 of 2)

```
-b58                                          Display EI version.
BF7000  90  00  00  13  00  10  41  07

er i rn=30 rv=1                               Disable IOU OS bounds.
30  0000000000000001

er m  21  0                                   Disable memory bounds.
21  0000000000000000

eb  bf7003  1  2  3b  4                       Change memory within EI.

db                                            Check memory change.
BF7000  90  00  00  01  02  3B  04  07

mcr  8010                                     Display bit definitions
MCR = DUE ,SIT ,                              for a MCR value of B010.

ucr 9000                                      Display bit definitions
UCR = PRIV FAULT,PIT ,                        for a UCR value of 9000.

dp  a                                         Display PP A registers.
004000 000073 000127 000145 000104 000153 000154 000001 000147 006421
000022 000001 000002 000056 004276 001460 173542 004563 000111 777771
```

Figure 2-2. Examples of MDD Command Usage (Sheet 2 of 2)

MEM

MEM is a CPU test of the central memory. It tests a field specified by a starting and a final address.

MEM is derived from the 7000 diagnostic SCM1 and requires a minimum field length of 2000_8 locations. However, a field length of at least 5000_8 should be specified.

MEM uses the system memory macro to determine the amount of memory to test.

CONTROL CARD AND CONSOLE CALLS

To call MEM in infinite pass count, the following card deck should be used:

```
JOB CARD
MEM.
6/7/8/9     EOF
```

To call MEM in under DIS, the following command should be used:

```
ENFE,(MEM TO TEST)
X.MEM.
```

TEST SELECTIONS

MEM contains eight different test sections, each of which is selected by setting an appropriate memory location to 1. Setting a location to 0 causes the test section to be bypassed. The memory locations used for this purpose are in the range of T110 through T117, and each one corresponds to a test section 0 through 7.

The individual subtest error counts are also kept in these locations. Therefore, the contents of addresses T110 through T117 are in the following format:

```
xxxx xxxx xxxx xxxx 000y

    xxxx xxxx xxxx xxxx  Error count

                      y  Subtest selection flag

                         1  Select test section.

                         0  Bypass test section.
```

TEST SECTIONS

The test sections for MEM are:

Test Section	Description
0	Ones and zeros test. Words of all ones and all zeros are written and checked four times.
1	Addressing test 1. The complete memory address (or its complement) is written and checked four times.
2	Addressing test 2. An operand in the form ABCD ABCD ABCD ABCD ABCD, or its complement, is written into each address. ABCD is treated as a 12-bit quantity and is incremented by one for each subsequent address. The data is then checked. Four passes are made through this test section.

Test Section	Description
3	Worst-pattern test. Writes and checks worst pattern and its complement worst pattern four times.
4	Checkerboard test. Writes and checks a checkerboard pattern and its complement four times.
5	Random addressing and data test. Divides the selected memory field into halves. Generates a set of pseudo random numbers, which are then duplicated in the upper and lower halves of the field. The field is then addressed in a random manner. The two halves are then read and compared. This test section is repeated four times.
6	Ones test. Clears test area, writes one word of ones in one test location, sweeps area, and tests word of ones. Repeats, testing the word of ones in every location.
7	Single-bit worst-pattern test. Writes worst pattern or the complement worst pattern on each plane individually. This test section is repeated four times.

All tests in the test sections just described sweep and complement the memory field several times after writing the pattern and before checking.

SIGNIFICANT MEMORY LOCATIONS

The following are the significant memory locations for MEM:

Address	Tag Name	Description
405	E.SAVE	Contents of register in exchange package format (fmt) upon entering subroutine E.ERROR.
1742	T101	Starting address. Default is last word address of test code plus 1.
1743	T102	Final address. Default is determined by amount specified by EDITLIB.
1744	T103	Contains a mask quantity that determines which bits of the memory word are to be checked. Bits corresponding to one bits are tested. Bits corresponding to zeros are ignored. Default is 7777777777777777777.
1745	T104	Error stop control. 0 Do not stop on error. Non-0 Stop on error.
1747	T106	Pass count.
1751 through 1760	T110-T117	Test sections select flags and error counts. Default is T110 = 1 T111 = 1 T112 = 1 T113 = 1 T114 = 1 T115 = 1 T116 = 0 T117 = 0

ERROR MESSAGES

When an error is detected during testing, the following information is recorded and an appropriate entry is made in the dayfile:

Error Code	Description
T120	Failing address
T121	Contents of failing address
T122	Correct word
T123	Number of test section that failed

The dayfile entry includes:

- Test name
- Failing test header (for example, one and zeros test)
- Expected data
- Actual data
- Pass count
- CPU diagnostic /MEM/ failed

MRG

MRG is a merging of five existing CPU programs into one single program. The programs merged are:

Program	Description
BGK	30-bit instruction test
FDT	Floating divide test
LAT	Long add unit test
POP	Population counter test
RTJ	Return jump test

There are no selectable parameters for MRG. For this reason, if an error is detected by one of the merged tests, that test should be run independently for further error analysis. Although the purpose and description of each test have not been changed, certain parameters and the contents of some memory locations have been changed, allowing the tests to be run in an on-line mode. A detailed description of MRG and each merged test can be obtained from the REL2B source listing.

SIGNIFICANT MEMORY LOCATIONS

The following are the significant memory locations for MRG:

Address	Tag Name	Description
405	E.SAVE	Contents of register in exchange package format (fmt) upon entering subroutine E.ERROR.
1533	T040	B-register failure (BGK).
1535	T041	X-register failure (BGK).
1537	T042	A-register failure (BGK).
1563		44 instruction failure (FDT).
1576		45 instruction failure (FDT).
2120		36 instruction failure (LAT).
2136		37 instruction failure (LAT).
2233		47 instruction failure (POP).
2271	T27	01 instruction failure (RTJ).

MY1

MY1 is a memory test that writes a value into each location equal to the location address from tag name CKMEM up to the field length. Each location is then read and compared with the current test address five times. The test is then repeated using the complement of each address.

Although the purpose and description of the test have not been changed, certain parameters and the contents of some memory locations have been changed, allowing the test to be run in an on-line mode. A detailed description of this test can be obtained from the REL2B source listing.

SIGNIFICANT MEMORY LOCATIONS

The following are the significant memory locations for MY1:

Address	Tag Name	Description
405	E.SAVE	Contents of register in exchange package format (fmt) upon entering subroutine E.ERROR.

All pertinent information for this test can be obtained from the A, B, and X registers. The contents of these registers are saved by the subroutine E.ERROR of CPUMEM. Failing P register addresses are displayed by CPUMEM.

OPMSG

The OPMSG command allows the customer engineer to communicate with the console operator through the DSD A,OPERATOR display. The following is the format of the OPMSG command.

OPMSG. message

message	Message text that appears on A,OPERATOR display. If null, the command is treated as a no-op. A maximum size of 70 characters is permitted.

If a message is specified, the OPMSG command waits for an operator response via the CFO command. This response is logged in the job dayfile.

This command is valid on NOS Version 2 only. The customer engineer must have maintenance privileges (AW = CMNT) in order to use this command.

RAN

This test is used to test the central processor's ability to translate and execute sets of randomly generated instructions. Each set consists of 10_8 numbers, from which all jump instructions are removed, so that control is maintained within the set. Each set is then executed, first in a slow loop and then in a fast loop. The results of each loop are then compared, one with the other.

Although the purpose and description of this test have not been changed, certain parameters and the contents of some memory locations have been changed, allowing the test to be run in an on-line mode. A more detailed description of this test can be obtained from the REL2B source listing.

SIGNIFICANT MEMORY LOCATIONS

The following are the significant memory locations for RAN:

Address	Tag Name	Description
405	E.SAVE	Contents of register in exchange package format (fmt) upon entering subroutine E.ERROR.
2100	FAST	Beginning address of fast loop.
2313	SLOW	Beginning address of slow loop.
1756	ERRE+15D	Answer difference.
1757	ERRE+16D	Failing fast loop answer.

Address	Tag Name	Description
1760	ERRE+17D	Failing slow loop answer.
1761	ERRE+18D	Failing fast loop answer.
1762	ERRE+19D	Failing slow loop address.
1763	ERRE+20D	Error counter.

Address	Tag Name	Description
2477 through 2516	SANS ⟶ SANS+17B	Slow loop answers.
2517 through 2536	SANS+20B ⟶ SANS+37B	Fast loop answers.

NOS REMOTE DIAGNOSTIC FACILITY INTERACTIVE PROCEDURES

NOS provides CDC customer engineers with the ability to perform maintenance software functions interactively. The remote diagnostic facility (RDF)† can be used from any single-dagger footnote location by dialing the computer's telephone number and is available only at sites with 800 series models.

NOS is a powerful software system that satisfies wide spectrum of computational needs. Its mass storage and advanced file maintenance techniques permit large amounts of information to be stored at high speeds. The system can be used from terminals in remote locations, thereby eliminating travel to the computer site by allowing users to work in their own offices.

TERMINALS

All communication with the system can take place through a remote terminal. Programs or data are sent to the system according to specific rules or commands. The system responds by sending its answers to the terminal. A slash (/) or READY indicates the system is editing for the next command. If the user enters an incorrect command, the system rejects the command and prints the reason for its rejection. For a detailed description of these messages, refer to appendix B of the NOS Version 2 Reference Set, Volume 3, System Commands, publication number 60459680.

ASCII code compatible terminals (such as Texas Instruments Silent 700/735/745, Anderson Jacobsen 830/860, CDC 713/752, CDC Viking X, DEC VT100, Zenith Z19, Heath H19, and so forth) are used to communicate with the system.

INPUT/OUTPUT CONVENTIONS

The following conventions and standards apply to a time-sharing terminal.

Length of Input/Output Lines

The input line can consist of a maximum of 160 characters. The output line can be any length; however, it should not exceed the page width of the terminal being used. If the user attempts to input or output more characters per line than are allowed, the additional characters overprint at the end of the line, unless the terminal being used has an automatic carriage return feature.

System Messages During Input

The following diagnostic messages can appear during input from the terminal.

OVL

Line overflow. This message is issued when more than 160 characters have been entered since the last carriage return (the entire line is lost when the 161st character is entered).

RE-ENTER LAST LINE

Data was lost during the last line of input. The user should reenter the last line.

Termination of Input Line

The user must terminate each line of input information by pressing CR. This tells the system that the current input line is complete. The system responds by positioning the carriage at the beginning of the next line. The user can then enter additional input information on the new line.

Correction of Input Line

The user can correct entry errors in the input line, before pressing CR, by using the backspace character. The backspace character on ASCII code terminals is either the BACK SPACE key or, on terminals that do not have a BACK SPACE key, the CNTL H keys. One character is deleted for each backspace character entered and the backspace character is not printed. A space equals one character. If the beginning of the line is reached, further backspace characters are ignored. For example, when the input line contains:

```
ABL<--C46<--<--23
```

it is interpreted by the system as ABC23. <-- indicates a backspace.

Deletion of Input Line

When the user discovers an error before pressing CR, the current line may be deleted in one of the following ways. On ASCII code terminals, press the ESCAPE key or press the CNTL [keys. The system ignores the entire input line, responds by printing

†Previously known as remote maintenance facility (RMF).

DEL, and positions the carriage at the first character position on the next line. The following example uses the ESC key.

```
new,test

READY.
10 program t(output)
20 print 5
30 6 format 9*this is a*DEL* <--- press ESC key.
30 5 format (*this is it*)
40 end

list

10 PROGRAM T(OUTPUT)
20 PRINT 5
30 5 FORMAT (*THIS IS IT*)
40 END
READY.
```

Input to Executing Program

A ? output to the terminal normally indicates that the executing program has requested input. However, the program may include question marks in its normal output.

Interruption of Executing Program

To interrupt an executing program that is currently transmitting output to an ASCII code terminal, press the I key or the BREAK key. This process is called job suspension and is described later in this section.

Termination of Executing Program

To terminate an executing program that is currently transmitting output, press the S key on an ASCII code terminal. To terminate an executing program that is not currently transmitting output, enter the STOP command.

Block Edit Mode Terminal Use

Block edit mode is designed for use with ASCII code terminals with hardware buffering and editing capabilities. Typically, these terminals allow the user to compose a message which is stored in the terminal buffer until the operator presses the SEND key, at which time it is transmitted from the buffer. Before the message is sent from the buffer, it can be edited or corrected by the operator. The data in the buffer can consist of one or more lines, separated by carriage return codes and terminated by an ASCII end-of-transmission (EOT) code. The EOT code is generated automatically by most terminals.

Because of the editing capabilities of the terminal, data is assumed to be correct as received. Therefore, in block edit mode the escape (ESC) and backspace (BS) characters are regarded as input data rather than as control characters.

The operator can send multiple line messages in text mode or as line numbered source data. Multiple lines of input to a running program or multiple command lines are not permitted; however, the last line of line-numbered source data can be a command.

The following are the special characters that are recognized and processed by the system when the terminal is in block edit mode.

Character	Description
EOT (CNTL D)	The EOT character signals the end of the input transmission. If this code is used to terminate an input line, the system generates a carriage return and a line feed at the terminal.
CR	The carriage return denotes the end of a line of data. Because block edit mode terminals automatically provide a line feed, no line feed is returned by the system.
ETX (CNTL C)	The ETX character is used to exit from text mode. It is sent as the first and only character after an EOT.

RDF INITIATION PROCEDURES

This section discusses the procedures necessarty to initiate the RDF subsystem and perform maintenance software functions. The user cannot harm the system by making incorrect entries or other mistakes at the terminal. Mistakes can have only diagnostic consequences; that is, they only produce error messages (usually INCORRECT COMMAND) and do not damage the system or hardware in any way. Generally, the user is allowed as many chances as necessary to make a corect entry. However, if unsuccessful at login four times in succession, the user is automatically disconnected. If this happens, the user should check the user name and login procedures and try again.

RDF LOGIN PROCEDURE

The RDF login procedure has four basic steps.

Step	Description
Preparation	The user gathers the information needed before login.
RDF Initiation	The user notifies the system operator to enable the RDF subsystem.
Login Initiation	The user connects the terminal to the system.
Login Sequence	The user is identified as a valid user of the system and enters optional commands.

PREPARATION

Before attempting to connect to the system, the user must meet the following conditions:

- The system operator's telephone number must be known, as the user must call the operator to initiate the RDF subsystem.
- The user must be prepared to tell the operator which mode(s) of the system are to be initialized and what equipment is to be made available.
- The site baud rate used to run the two-port mux must be known. This determines the type of terminal that is needed to connect to the computer.
- The user must know how the terminal is connected to the computer (data set, acoustic coupler, or hardwired).
- The computer access phone number must be known if the terminal is not hardwired.
- A user name is necessary to gain access to the system login.
- An associated password is necessary, along with the user name, to gain access to the system.
- When required, a valid charge number and/or project number is necessary to catalog permanent files or submit batch jobs.

INITIATING RDF

To initiate RDF, the user must call the system operator by telephone and request that the RDF subsystem be initiated. The following describes the PRIVILEGED or UNPRIVILEGED and RESIDENT or NONRESIDENT modes. As stated earlier, the user must know the mode(s) in which RDF will be used.

Mode	Description
PRIVILEGED/ UNPRIVILEGED	PRIVILEGED allows the user to perform any maintenance or system functions available at a particular site. When this parameter is not specified (UNPRIVILEGED mode), the system allows the user to perform only those maintenance and system functions listed in the Terminal Control Commands and Additional UNPRIVILEGED Mode Commands sections. Default setting is UNPRIVILEGED.
RESIDENT/ NONRESIDENT	RESIDENT mode must be specified. User is not logged out after any period of inactivity. If the phone line between the computer and the terminal is interrupted, the user may log in any time after the interruption without calling the operator to initiate RDF. RDF stays up until the system goes down or the operator disables RDF. If not specified (NONRESIDENT mode), the user is automatically logged out after a 15 minute period of inactivity and RDF must be reinitiated by the operator for future use.

RDF may be initiated as both PRIVILEGED or UNPRIVILEGED and RESIDENT or NONRESIDENT. The user must be validated by the site to access RDF and must be further authorized to operate in UNPRIVILEGED and/or RESIDENT mode.

LOGIN INITIATION

The user is now ready to perform the steps that lead to the login sequence. The main steps are as follows:

1. Check operating characteristics.
2. Dial in.
3. Identify terminal.

The following paragraphs describe these steps; figure 3-1 summarizes them. Due to the variety of terminals and sites, the action taken in each step may vary considerably from site to site. The user should consult the owner's manual provided with the terminal, and site documentation to determine appropriate action.

Check Operating Characteristics

To check the operating characteristics of the terminal, the user should do the following:

1. Find the power switch on the terminal and turn it to the ON position. If the terminal is a CDC 721 and is already ON, the RESET button should be pressed to disable any host-loaded controlware.
2. If the duplex can be selected, ensure that it is set to HALF duplex. In FULL duplex, characters entered from the terminal are not printed.
3. If the parity can be selected, set it to EVEN for ASCII code terminals.
4. If the line speed can be selected, set it to the value that has been selected for the particular site.
5. If the data set is separate from the terminal, turn it on and check the connection between the data set and the terminal.

Dial In

The dial-in procedure connects the terminal to the computer through a data set or an acoustic coupler. Either one may be located in the terminal or as a separate unit beside the terminal. Hardwired terminals are already connected to the computer so the dial-in procedure is not necessary.

If the data set is located in the terminal, a telephone dial is located on the terminal panel. When the terminal is turned on, the dial tone is heard. Dial the appropriate phone number. When the connection is made, the terminal is ready to begin the login sequence.

If the data set is separate from the terminal standalone), the user must do the following:

1. Pick up the phone receiver.
2. Press the TALK button on the face of the phone; the button lights.
3. Dial the appropriate phone number.
4. When a continuous high-pitched tone sounds, press the DATA button on the face of the phone; the button lights.

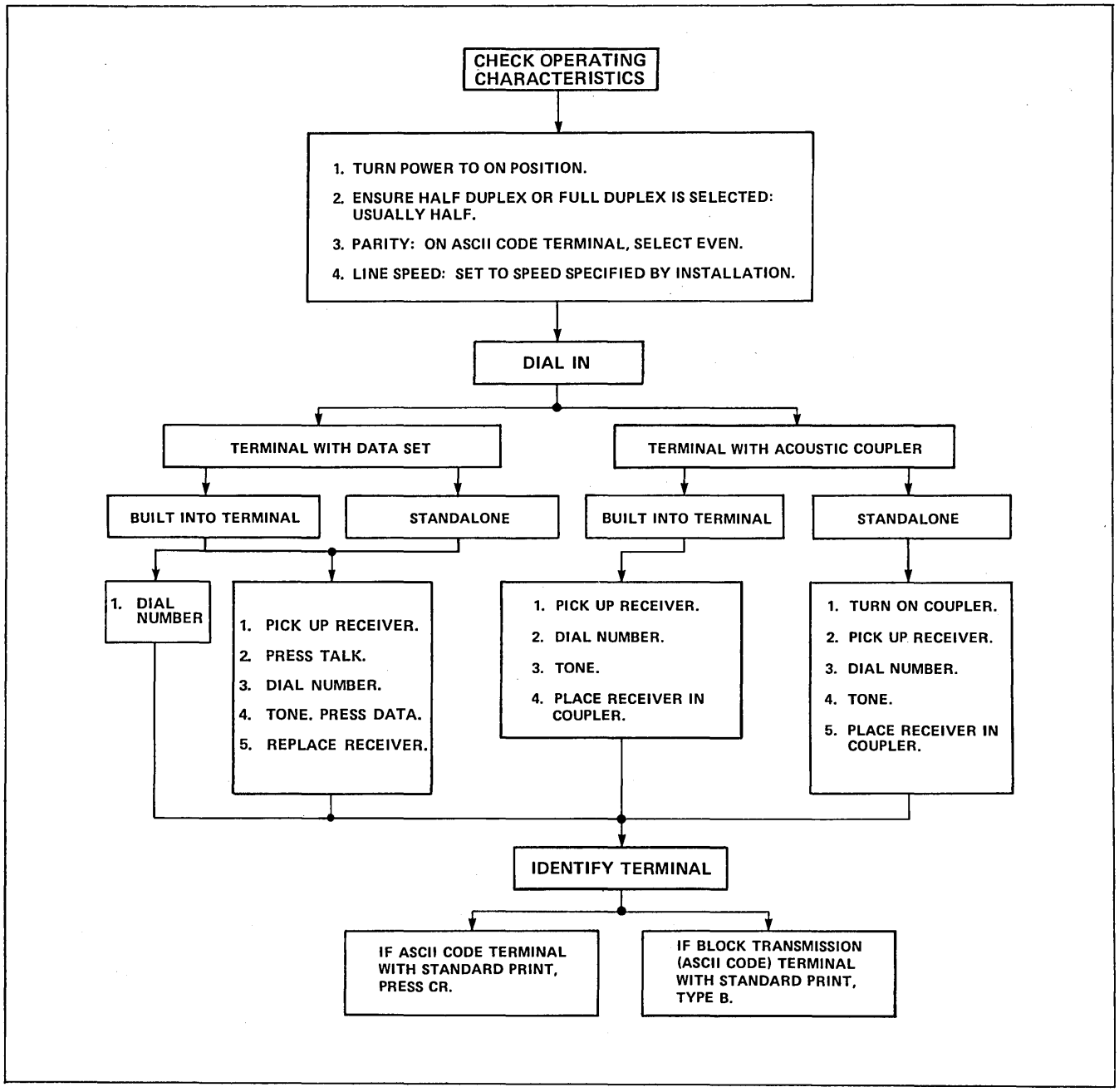

Figure 3-1. Login Procedure

5. Replace the phone handset in the cradle. (After DATA is pressed, replacing the phone handset does not disconnect the user.) Connection is made and the terminal is ready to begin the login sequence.

If the acoustic coupler is located in the terminal, it automatically turns on when power is applied to the terminal. If the acoustic coupler is a separate unit (standalone), it must be turned on after power is applied to the terminal. In either case, a standard telephone is used in the following manner.

1. Pick up the receiver and listen for a dial tone.
2. Dial the appropriate phone number.
3. A continuous high-pitched tone indicates that the call has been answered and connection has been established.
4. Place receiver in rubber suction cups of acoustic coupler with the cord at the end indicated on the coupler.

Identify Terminal

Once the user establishes connection with the computer, it may be necessary to identify the terminal before the login sequence can begin. In some terminals, the login sequence is automatically initiated when the dial-in is completed; in others, it requiresthe typing of a letter, the pressing of a particular key, or both. In all cases, the system indicates that the login sequence may begin by typing out three lines: date and time, header label, and a request for family name or a request for user name. The following paragraphs describe the login sequence in detail.

The following terminal types require the indicated actions to initiate the login sequence.

Terminal Type	Action
ASCII code terminal with standard print.	Press CR
Block transmission (ASCII code) terminal with full display-screen editing capability and standard print (for more information, refer to Block Edit Mode Terminal Use section).	Type the letter B.

Table 3-1 summarizes the functions of the keys as they appear on typical ASCII code terminals.

Table 3-1. Functions of Time-Sharing Terminal Controls (Typical ASCII Code Terminals)

Function of Control Key or Command	Texas Instruments Silent 700	Teletype 37/38	CDC 713 Display	Memorex 1240	CDC Viking X
Delete current input line	CTRL X	ESCAPE or CTRL	ESCAPE or CNTRL	ESC or CNTL	ESC or BREAK
Backspace	CTRL H	BACK SPACE or CTRL H	BACK SPACE or CNTRL H	BACK SPACE or CNTL H	← or CTRL H
Enter text mode	TEXT command	TEXT command	TEXT command	TEXT command	TEXT command
Exit text mode	INTRPT or ETX (CTRL C)	INTRPT or ETX (CTRL C)	BREAK or ETX (CNTRL C)	INT	BREAK† or CTRL C
Interrupt program (during output)	BREAK or I key	INTRPT or I key	BREAK or I key	INT or I key	BREAK or I key
Interrupt program (no output)	BREAK	INTRPT	BREAK	INT	BREAK
Terminate program (during output)	S key	S key	S key	S key	S key
Terminate program (no output)	STOP command	STOP command	STOP command	STOP command	STOP command
Punch leader on paper tape	DELETE or NULL	DELETE RUB OUT	NA or NULL	NA	NA

†Effective only when entered in first character position of line (empty input line).

LOGIN SEQUENCE

The login sequence begins with the terminal typing out three lines. The first line is in the following format.

```
yy/mm/dd.  hh.mm.ss.  termnam
```

The date is given in year/month/day and the time in hours. minutes. seconds, using the 24-hour format and a network-supplied terminal name. For example, if the login began at 12 minutes and 44 seconds after two o'clock in the afternoon on September 3, 1984, the first line would read:

```
84/09/03.  14.12.44.  TERM201
```

The second line is the header identifying the site which may give the company name, the operating system, and the version of the operating system. In the examples in this manual, the following is used:

```
CDC NETWORK OPERATING SYSTEM                NOS 2
```

The third line is a request for a family name, if the site is dividing its mass storage devices into families. The terminal types:

```
FAMILY:
```

The user types, on the same line, the name of the assigned family and presses CR. For example, if the mass storage device for the site is assigned to family AAA, the user types:

```
FAMILY:  AAA
```

The family name identifies the mass storage device(s) that contain(s) the permanent files for a specific NOS system. Each system has its own family of permanent file devices. Thus, if the system is providing backup support for another NOS system, if this login is being made into an alternate NOS system (two or more sets of permanent file devices identified with family names), or if the user is logging into a multimainframe system, the family name entered identifies the device(s) that contain(s) the user's normal permanent file family.

The device containing the validation file for the family must be present in the alternate system in order for the user to log in. The validation file is used to determine if the user name and password entered are valid.

When the assigned family is the default family name for the system, the user need only press CR.

```
FAMILY: CR
```

The system then requests the user name. When family names are not in effect, the request for a family name is omitted and the request for a user name is the third line typed out in the login sequence. The request for user name is:

```
USER NAME:
```

The user types the assigned user name on the same line and presses CR. For example, if the user name is ABC1234, the user types:

```
USER NAME: ABC1234
```

The system then responds:

```
PASSWORD
00000000
```

The second line results from the system overtyping a variety of characters (blackouts). The object is to preserve password secrecy. After creating this row of blackouts, the cursor moves back to the first blackout. If a password is in effect, the user types it over the blackouts and presses CR.

If a user has exhausted the security count on the user name entered, the system issues the message:

```
INCORRECT USER ACCESS - CONTACT SITE OPR.
```

The supplied user name is denied all access to the operating system until the security couint has been reset.

When the request for a user name appears, the user may enter the user name and password on the same line. To do this, the user types a comma after the user name, types in his or her password on the same line, and then presses CR. For example, if the user had previously been assigned the password PASS23, the request for user name could now be completed as follows:

```
USER NAME:  ABC1234,PASS23
```

When this is done, the system omits the two lines composed of PASSWORD and blackouts.

When the request for a family name appears, the user may enter the family name, user name, and password on the same line. If the user's family is the default family name for the system, the user first types a comma and then follows with user name and password. For example:

```
FAMILY:  AAA,ABC1234,PASS23

    or

FAMILY:  ,ABC1234,PASS23
```

When this is done, the system omits the three lines composed of USER NAME, PASSWORD, and blackouts.

If the family name, user name, or password is not acceptable, the system responds:

```
IMPROPER LOG IN, TRY AGAIN.
FAMILY:
```

If the user is unsuccessful at logging in four times in succession, the system responds:

```
ILLEGAL USER.
```

and disconnects the terminal from the system.

If the family name, user name, and password are acceptable, the system prints the job sequence name (JSN) and application on the next line. For example:

```
JSN:  ABCD, TPM
```

This is job sequence name ABCD in the time-sharing configuration for this site. The TPM indicates it is an ASCII code terminal with standard print. If a terminal with a display screen is in use, the user should write down the job sequence name (ABCD in this example) in case recovery is attempted (refer to Recovery section). Possible terminal types are:

Type	Description
TPM	ASCII code terminal with standard print.
BLKTPM	Block transmission (ASCII code) terminal with full display-screen editing capability, and standard print.

If the maximum number of users are logged in, the system replies:

```
SYSTEM BUSY, PLEASE TRY LATER.
```

If the operator has closed the system, the following message is received:

```
SYSTEM CLOSED.
```

The user must wait and try again later.

If the system is up, it responds:

```
READY.
```

LOGOUT PROCEDURE

The user logs out when finished with work at the terminal. The system automatically logs out the user if no activity is registered in any 10-minute period. If RDF is enabled in RESIDENT mode, the user remains logged on the system until RDF is terminated; there is no time limit. To log out of the system, the user types:

```
GOODBYE
   or
BYE
   or
LOGOUT
```

The system responds by printing:

```
UN=user name     LOG OFF    hh.mm.ss
JSN=jsn          SRU-S      s.sss
CHARACTERS=      n.nnnKCHS
```

and then disconnecting the terminal.

The following is a description of the variable items in the logout display.

item	Description
username	User name entered during login.
hh.mm.ss.	Time of logout.
jsn	Job sequence name.
s.sss	Measure of the system resources used while connected to IAF.
n.nnnKCHS	Count of the total number of input and output characters read from, or written to, the terminal.

While the terminal is connected, another user may want to use the terminal. To log the present user out of the system and reinitialize the login sequence, the user types:

```
HELLO
   or
LOGIN
```

The system logs the current user out of the system, issues the normal logout messages, and automatically initiates a new login sequence. The new user proceeds as described in this section, starting with the login sequence.

RDF TERMINATION

If the system operator must terminate RDF while a user is logged in, the following message appears at the user's terminal.

```
RDF DROPPED BY THE OPERATOR
```

The user is automatically logged off the system. In NONRESIDENT mode, RDF is automatically terminated when the user logs off the system. All activities of each user are recorded in the system dayfile.

USER/OPERATOR ON-LINE COMMUNICATION

NOS provides the RDF user with the capability to send messages to the system operator on-line while a job is running. To send a message to the operator, the escape control character (ESC) and the message command (MS) command are used. The format is as follows:

```
(ESC)MS=xxx...  ...xxx(CR)
```

(ESC) signifies the entering of the ESCAPE control character, xxx... ...xxx represents the user's message, and (CR) represents entering the carriage return. To determine the ESCAPE control character of your terminal, refer to the terminal's operations manual or appendix K of the NOS Version 2 Reference Set, Volume 3, System Commands publication number 60459680. This message may not exceed 48 characters.

If the system console is busy, the following is displayed at the user's terminal.

```
CONSOLE BUSY
```

If the system is not busy, the operator responds with an appropriate message which, also, may not exceed 48 characters.

The user may also send a message to the system operator using the OPMSG command. The command format is as follows:

```
OPMSG.xxx...  ...xxx
```

xxx... ...xxx represents a 64-character message which appears on the A,OPERATOR display. The operator response appears in the user's dayfile.

RECOVERY

During job processing, recovery may be necessary when:

- The terminal is accidentally disconnected from the system.
- A system malfunction occurs which requires a restart.

The terminal is placed in recovery state whenever it is disconnected from the system without being logged out (providing that it is not already in recovery state). The user has 10 minutes to initiate recovery.†

If there is a system malfunction which requires a restart, the user has 10 minutes from the time the system is restarted to initiate recovery.

To initiate the recovery process, the user completes the login sequence and, if any of the user's previous jobs are recoverable, the system lists the recoverable job(s) as shown in the following example.

```
RECOVERABLE JOB(S)

JSN    UJN       STATUS       TIMEOUT

AAPE   AKVA      SUSPENDED     8 MIN.

ENTER GO TO CONTINUE CURRENT JOB
      RELIST TO LIST RECOVERABLE JOBS,
      OR DESIRED JSN:
```

The user enters GO to continue current job processing, RELIST to list all recoverable jobs, or specific job sequence name(s) of jobs that the user may want to recover.

When the user enters a job sequence name, and the recovery is successful, the system responds as shown in the following example.

```
   JSN:  AAPE   SYSTEM:  BASIC     SRU:       2.151
FILE NAME: FILE1
   STATUS:        .002 CP SECONDS EXECUTION TIME
   CHARACTER SET: NORMAL   MODES: PROMPT ON
   IDLE.  ENTER COMMAND.

   READY.
```

When the user enters a job sequence name and recovery is impossible, the system responds as shown in the following example.

```
ENTER GO TO CONTINUE CURRENT JOB
      RELIST TO LIST RECOVERABLE JOBS,
      OR DESIRED JSN:acav
ACAV NOT FOUND.
ENTER GO TO CONTINUE CURRENT JOB
      RELIST TO LIST RECOVERABLE JOBS,
      OR DESIRED JSN:
```

†This limit may vary from site to site. Contact the site analyst for the correct limit.

Whenever the user and job sequence name(s) of a user requesting a recovery state match the user and job sequence name(s) of a user already in a recovery state, the user already in the recovery state remains there and the other user is logged out. If a user is disconnected after logging in but before recovery from a previous disconnect, the latter time is processed as a normal logout to protect the user from an intermittent phone line failure. The user can then log in again and recover the job.

The following is an example of a typical recovery.

```
82/01/11. 10.42.08. TERM201
CDC NETWORK OPERATING SYSTEM        NOS 2
FAMILY: famname<--------------------------------User logs in.
USER NAME: abc1234
PASSWORD
password
JSN: AAPE, TPM
READY.
list,file1 <-------------------------------------User lists file named file1.

00100 LET A=1
00110 LET B=10
00120 FOR I=1 TO B
00130 A=A*I
00140 PRINT I," FACTORIAL  IS  ";A
00150 NEXT I
00160 END

basic <------------------------------------------User enters BASIC subsystem.

 READY.
run <--------------------------------------------------User runs program.
 1              FACTORIAL  IS   1
 2              FACTORIAL  IS   2
 3              FACTORIAL  IS   6
 4          | <-------------------------------Phone line interruption occurs.

82/01/11. 10.48.53. TERM201
CDC NETWORK OPERATING SYSTEM      NOS 2
FAMILY: famname                <----------------------User redials phone
USER NAME: abc1234                                    number and logs in.
PASSWORD
password
JSN: AAQG, TPM

   RECOVERABLE JOB(S) <-----------------------------------System prints list
                                                          of recoverable jobs.
   JSN    UJN       STATUS       TIMEOUT

   AAPE   AKVA      SUSPENDED     8 MIN.

ENTER GO TO CONTINUE CURRENT JOB
      RELIST TO LIST RECOVERABLE JOBS,
      OR DESIRED JSN::aape <-----------------------------User specifies JSN.
 JSN: AAPE  SYSTEM: BASIC    SRU:      2.151  FILE NAME: FILE1
 STATUS:        .002 CP SECONDS EXECUTION TIME
 CHARACTER SET: NORMAL   MODES: PROMPT ON
 IDLE.  ENTER COMMAND.

 READY.
run <--------------------------------------------------User reruns program.
 1              FACTORIAL  IS   1
 2              FACTORIAL  IS   2
 3              FACTORIAL  IS   6
 4              FACTORIAL  IS   24
 5              FACTORIAL  IS   120
 6              FACTORIAL  IS   720
 7              FACTORIAL  IS   5040
 8              FACTORIAL  IS   40320
 9              FACTORIAL  IS   362880
 10             FACTORIAL  IS   3628800

RUN COMPLETE.
```

JOB SUSPENSION

A user can suspend a job at any time during job execution by pressing the BREAK key. If the job is actively transmitting output to the terminal, the I key can also be used to suspend the job. The system responds:

```
*INTERRUPTED*
```

A job is also suspended automatically by the system when one of the following occurs:

- The job has exceeded its time limit. The time limit is initially set to 64 CPU seconds at login, although the user can change this value through use of the SETTL command (refer to the NOS Version 2 Reference Set, Volume 3, System Commands, publication number 60459680).
- The job or job step has exceeded its SRU limit. The SRU limits are initially set to 320 SRUs at login, although the user can change this value through use of the SETASL or SETJSL commands (refer to the description in the NOS Version 2 Reference Set, Volume 3, System Commands, publication 60459680).
- A successful recovery has been performed (refer to Recovery in this section).

If the job is suspended by the user (interrupted), the user can perform one of the following:

Entry	Description
CR	Continue. A portion of the output is lost if the job is transmitting output to the terminal when suspended. When output is complete, job execution continues. If the TAPE command is in effect (tape mode), carriage return is ignored when entered on any empty input line. Therefore, P CR must be issued to resume execution.
P	Proceed. If the job is transmitting output to the terminal when suspended, the system discards the data generated by the job before the user interrupts it. The amount of output discarded depends on the job being executed. Program execution continues.
STOP	Stop. Terminates the job step.
Other	Stop. Terminates the job step.

If the job step is suspended because it exceeded its time limit, the message:

```
*TIME LIMIT*
ENTER T TO CONTINUE OR CR KEY TO STOP:
```

is issued and the user can enter one of the following:

Entry	Description
T	Increases the central processor time limit by 64 CPU seconds. Job execution continues.
T,nnnnn	Increases the central processor time limit by nnnnn decimal seconds. Job execution continues. The user can enter octal seconds by specifying a B after the octal number.
T,*	Increases the central processor time limit to the user's maximum. Job execution continues.
	Causes the job to go through normal abort procedures (for example, EXIT processing which can be useful when using procedure files).
Termination sequence	Terminates the job step. Subsequent control statements, if any, are not processed.
STOP	Terminates the job step. Subsequent control statements, if any, are not processed.

Any increase to the central processor time limit through either T or T,nnnn is in effect only for the current job step. When the job step terminates, the central processor time limit reverts to its original value, previously set by default or by the SETTL command.

If the user enters something other than one of the previous entries, the system prompts the user again.

If the job is suspended because a job step or the job itself exceeded its SRU limit, the message:

```
*SRU LIMIT*
ENTER S TO CONTINUE OR CR KEY TO STOP:
```

is issued. The user can enter one of the following:

Entry	Description
S CR	Increases the SRU limit by 320 units. Job execution continues.

Entry	Description
S,nnnnn CR	Increases the SRU limit by nnnnn decimal units. Job execution continues. The user can enter octal units by entering a B after the octal number.
S,* CR	Increases the SRU limit to the user's maximum. Job execution continues.
CR	Causes the job to go through normal abort procedures.
Termination sequence CR	Terminates the job step. Subsequent control statements, if any, are not processed.
STOP CR	Terminates the job step. Subsequent control statements, if any, are not processed.

Any increase to the SRU limit through either S or S,nnnn is in effect only for the current job step. When the job step terminates, the account block SRU limit and job step SRU limit revert to their original values, set by default or by the SETASL and SETJSL commands, respectively. Entering S,* in response to an SRU LIMIT message could cause the time-sharing session in progress to exceed the account block SRU limit set by the SETASL command or by default. The system then issues an SRU LIMIT message after a job step has begun. To remedy this situation, use the SETASL command to raise the account block SRU limit above the current number of accumulated SRUs for the time-sharing session.

If the user enters something other than one of the previous entries, the system prompts the user again.

TERMINAL CONTROL COMMANDS

The terminal control commands allow the user to change the characteristics of the terminal and to vary the source and format of information given to and received from the system. These commands, which can be entered at any time after the user has successfully logged in, are as follows:

Command	Description
ASCII	Selects the full ASCII character set. This command causes subsequent characters entered from the terminal to be translated into 6/12 display code. On an ASCII code terminal, this code set represents the ASCII 128-character set. The standard character set contains only the first 63 or 64 of these 95 characters. The system recognizes all characters of the full ASCII code set except line feed (LF) and delete (DEL). The input control characters for return and backspace are recognized but are not translated. Characters of the standard character set are stored as 6-bit display code characters. The additional characters which make up the full ASCII code set are stored as 12-bit display code characters with an escape code convention. Refer to appendix A for an explanation of how the system interprets characters in both ASCII and normal character modes. The ASCII command must be entered if lowercase letters are to be interpreted by the system. In normal character mode, all lowercase letters are translated to uppercase. Commands may be entered in uppercase or lowercase in either ASCII or normal character mode.
AUTO,nnnnn, iiii	Automatically generates five-digit line numbers. The nnnnn parameter specifies the beginning line number; default value is 00100. The iiii parameter specifies the increment value added for each succeeding line number; default value is 10. The user can exit from auto mode by using either of the following methods. • Delete the current line by pressing the ESCAPE key or CNTL [keys (ASCII code terminals) and then entering a new command on the next line. • Backspace six character positions and then enter a new command on the same line (six backspaces are required to overwrite the five-digit line number and the blank that follows). The line numbering sequence can be altered by deleting the line or backspacing and then entering a new beginning line number rather than a new command (leading zeros are permissible but not required). The user should exercise caution when doing this since the AUTO command is still in effect and continues generating line numbers using the original increment value. Thus, if a line number is generated that already exists in the file, the original contents of that line are lost and must be reentered. The increment value cannot be altered unless a new AUTO command is entered.
BRIEF	Suppresses all full and partial headers such as those issued by the LIST or RUN command.
CSET,c	Sets the character set mode of the terminal to the specified mode. c Specifies terminal character set mode. ASCII ASCII mode (ASCII 128-character set. The CSET command may also be entered through commands included in a procedure file.

NOTE

The CSET,NORMAL command sets only the terminal character set. It does not have an effect on the AUTO, PARITY, or TAPE commands.

Command	Description
FULL	Selects full-duplex mode. Under this mode, each character received by the system is echoed to the terminal just as received. This mode is effective only for terminals that have a full-duplex capability. The system responds: READY. If the HALF/FULL duplex switch on the terminal is in the HALF position when this command is entered, each subsequent character entered is double printed (initially when it is typed and again when it is echoed by the system). Placing this switch in the FULL position allows only the character echoed to the terminal to be printed.
HALF	Clears full-duplex mode (refer to FULL command). Characters received by the system after this command is issued are not echoed to the terminal. The system responds: READY. If switch is in the FULL position, characters entered at the terminal are not printed. Only system-generated output appears at the terminal. Placing this switch in the HALF position enables keyboard entry to be printed.
NORMAL	Reverses the effect of the ASCII, AUTO, BRIEF, CSET, NOSORT, PARITY, and TAPE commands on both input and output. The system initially assumes that the terminal is in normal mode. Normal mode uses the standard 64- or 63-character set. All lowercase letters are converted to uppercase (refer to ASCII command for additional information) and all command headers are printed (refer to BRIEF command).
PARITY,p	Sets terminal parity where p is either: ODD to set odd parity or EVEN to set even parity If no parameters are supplied, odd parity is assumed.

Command	Description
	The system initially assumes that all information is transmitted in even parity to ASCII code terminals.
ROUT,nn	Sets the amount of time required to perform the return function (carriage return and line feed) for an ASCII code terminal. The nn parameter specifies a character count delay. This is the amount of time required by the system to send nn characters to the terminal, where the value of nn can range from 0 to 30. If 0 is specified, a standard delay value is used. The system responds: READY. This command is necessary since the length of time required for the return operation varies depending upon the type of terminal being used. For example, if the system is transmitting output to a terminal, a fixed amount of time is allowed for the return function, after which the system sends the next line of output. If the amount of time is not sufficient, characters may be printed during carriage return function. The delay is accomplished by sending rubout characters for paper tape operations and null characters for all other I/O operations. The number of characters to be sent is initially set by the system at login or at recovery time. If a ROUT command is entered, the number of characters specified for the delay remains in effect until termination of the session (logoff), until recovery, or until another ROUT command is entered.
S key	Terminates a job that is currently transmitting output to the terminal. This key is effective only for ASCII code terminals. The system responds: *TERMINATED* If pressing this key accidentally disconnects the terminal from the system, dial the computer and initiate the job recovery procedure as described earlier in this section.
STOP	Terminates any program that is currently in execution or waiting for input from the terminal, unless the program has disabled terminal control; for example, APL. The system responds: *TERMINATED*

Command	Description
TAPE	Permits subsequent information to be read from the paper-tape reader at an ASCII code terminal. The system sends an X-ON character to the terminal at the end of each program request for data and after execution of each command. The X-ON character turns on the paper-tape reader at the user's terminal. This mode also inhibits the output of header messages from the LIST command and the READY message after the execution of most commands. The system ignores carriage return characters entered on an empty input line while in tape mode.
TERM,t	Allows the user to redefine the terminal characteristics that were established at login. The t parameter specifies the new terminal characteristics as follows:

Mnemonic	Description
TPM	ASCII code terminal with standard print.
BLKTPM	Block transmission (ASCII code) terminal with full display-screen editing capability (available only on certain terminals such as the Hazeltine 2000 terminal) and standard print.

NOTE

All ASCII-code terminals with standard print are designated TPM. The Memorex 1240 is an ASCII code terminal and, therefore, is identified as TPM when using standard print.

	Terminal characteristics may be selected by default. Enter the LIMITS command to obtain the default value.
TIMEOUT	Changes a no-timeout terminal to the standard timeout status. In UNPRIVILEGED RDF mode, the user is automatically logged out after 15 minutes of inactivity. Standard status is in effect when bit 10 in access word AW is clear. (Refer to MODVAL Command in the NOS Version 2 System Maintenance Reference Manual, publication number 60459300.) When the bit is set, the terminal remains connected until the user logs out. The TIMEOUT command clears this bit for the session in progress.

ADDITIONAL UNPRIVILEGED MODE COMMANDS

The following NOS system commands are also available to the user when operating under UNPRIVILEGED RDF mode. For descriptions of these commands, refer to the NOS Reference Set, Volume 3, System Commands, publication number 60459680.

ASSIGN	ATTACH	BEGIN	CATALOG	CATLIST	CHANGE
CHARGE	CLEAR	COPY	COPYBF	COPYBR	COPYCF
COPYCR	COPYEI	COPYER	COPYSBF	DAYFILE	DEFINE
DISPLAY	DMD	DMDECS	DMP	DMPECS	EDIT
ENQUIRE	EXIT	GET	LABEL	LIMITS	LISTLB
MFL	MODE	NEW	OFFSW	OLD	ONSW
OPMSG	PACK	PACKNAM	PERMIT	PRIMARY	PURGE
RECOVER	REDUCE	RENAME	REPLACE	REQUEST	RESOURCE
RETURN	REVERT	REWIND	RFL	ROUTE	SAVE
SET	SETTL	SKIP	SKIPEI	SKIPF	SKIPFB
SKIPR	STATUS	SUBMIT	SWITCH	TDUMP	TEXT
UNLOAD	USECPU	USER	VERIFY	VSN	WRITEF
WRITER	XEDIT				

The following maintenance software commands are allowed under UNPRIVILEGED RDF operation.

MALET	HPA	REGEN	TIO
ALX	CSU	CTB	CU1
FS8	EC3	EC8	ELD
NORM	GETLOG	MRG	

PRIVILEGED MODE OPERATION

When operating in PRIVILEGED RDF mode, the user may utilize any of the NOS system commands or maintenance software functions available at the site. For descriptions of NOS, refer to the following manuals.

Control Data Manual	Manual Number
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

COMMENT SHEET

MANUAL TITLE: CDC NOS On-Line Maintenance Software Reference Manual

PUBLICATION NO.: 60454200 **REVISION:** J

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
ARH219
4201 North Lexington Avenue
Saint Paul, Minnesota 55112

CUT ALONG LINE

FOLD FOLD

CORPORATE HEADQUARTERS, P.O. BOX 0, MINNEAPOLIS, MINN. 55440
SALES OFFICES AND SERVICE CENTERS IN MAJOR CITIES THROUGHOUT THE WORLD

LITHO IN U.S.A.

CONTROL DATA CORPORATION