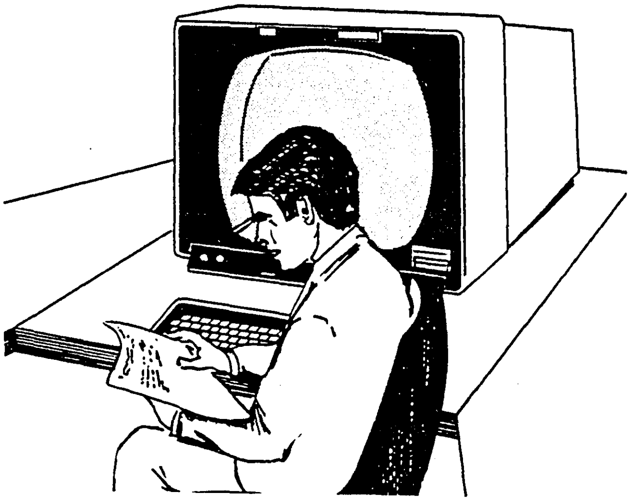
CMSE OPERATION

a computer based education course



STUDENT MANUAL

CONTROL DATA

REVISION RECORD

REVISION	DESCRIPTION
A	First Edition
11/30/78	

Publication No.
754435[redacted]
62

EDUCATION QA, NTHESE

CONTENTS

FIGURES

APPENDIX

INTRODUCTION

CMSE Operation is a computer-based education (CBE) course using computer-assisted instruction (CAI) techniques along with text readings, reference readings, and several simulated exercises. This course is designed to teach the operation of the maintenance software library (MSL) which includes the common maintenance software executive (CMSE).

Management of the course is accomplished via computer-managed instruction (CMI), a system using the PLATO terminal. Basically, the student should progress through the course as follows:

- Read through Appendix A.
- Sign on to the terminal.
- Read through the introductory material.
- Select the first block, "Getting Started."
- If you think you know some of the material, take the test. Otherwise, request a study assignment and read through it before taking the test.
- After mastering the tested objectives, proceed to the next block until all blocks are mastered.

BLOCK 1
GETTING STARTED

The activities in Block 1 are designed to teach you how to deadstart CMSE and how to run stand-alone tests. It shows you many of the displays available.

Throughout this course, you will be asked to read portions of a reference manual entitled "CYBER 17X Off-Line Maintenance System Library (MSL) Reference Manual." The LDS part number for this manual is 60455770. You will only use volume 1 in this course.

The material in this manual is current up to the date indicated by the revision record in the front of the manual. Changes to the diagnostics which occur after this date are not reflected. If you detect any changes or errors, please indicate them on the comment sheet in the back of the manual so that they may be incorporated into the next revision.

YOUR ASSIGNMENT

To receive your assignment, log on to the PLATO terminal. At the block options page, select Block 1, "Getting Started" by entering a and pressing the NEXT key. At the Block 1 description page, press a to receive an assignment or press t to take the test. Your current knowledge may enable you to skip some of the learning activities. Mark those activities assigned and check them off as you complete each activity. You may choose to do all of these activities, or you may choose to do some activities more than once.

Check	Activity	Description	Page
_____	1-A	Text Reading: MSL/CMSE Introduction. This activity teaches hardware requirements, software restrictions, and the purpose of command buffers.	1-3
_____	1-B	Text Reading: Deadstart Procedures. This activity teaches how to deadstart CMSE, how to run stand-alone tests, and how to make console system changes.	1-5
_____	1-C	CAI: Deadstart. This activity gives you practice deadstarting CMSE and running stand-alone tests.	1-8
_____	1-D	Text Reading: Basics. This activity teaches the major displays and shows the functions of special keys on the keyboard.	1-9

Check	Activity	Description	Page
______	1-E	CAI: Basics. This activity gives you practice selecting displays and making special keyboard entries.	1-10

LEARNING ACTIVITY 1-A. TEXT READING: INTRODUCTION TO MSL/CMSE

This learning activity is designed to teach the hardware requirements and software restrictions of CMSE. The purpose of the command buffer is also described.

In the past, the off-line maintenance software for large systems was SMM (system maintenance monitor). This maintenance system is now replaced by MSL (maintenance system library). This new system, MSL, offers several advantages to the user over SMM. Most importantly, where SMM varies significantly depending upon the type of computer system being tested, 6000 vs CYBER 170 for example, MSL provides one system for the entire range of large computer systems, 6000 through CYBER 170. Thus, when a person is familiar with the operation of MSL on one system, its use on any other system poses no problem.

MSL may be divided into three parts.

1. Diagnostics
   a. Peripheral processor
   b. Central processor unit
   c. Memories
   d. Peripheral equipment
2. Stand-alone test loader
3. CMSE (common maintenance software executive)
   a. Monitor
   b. I/O driver(s)
   c. Display driver
   d. Utility routines

The diagnostics portion of MSL consists of the actual tests and their drivers which perform the actual testing on the machine areas in the preceding list. The stand-alone test loader is brought into PPO at deadstart time and handles the loading and execution of deadstart (stand-alone) tests. These tests replace the stand-alone loader and therefore require a deadstart following each use. In addition, the stand-alone test loader provides a means of system (hardware) definition and CMSE loading. This course will discuss the stand-alone test loader from a user standpoint, however, the primary purpose is to cover the CMSE portion of MSL.

CMSE Operation
Learning Activity 1-A

CMSE is a software system comprised of the four basic elements in the preceding list. The function of CMSE is to provide the human interface required for actual execution and interpretation of the diagnostics contained within MSL. You will be examining the various displays and keyboard entries required to use CMSE in solving problems. CMSE is capable of operating either by using a magnetic tape for its library of programs or by using a disk. It should be noted that operation from disk is not only faster, but allows library modification, dayfile generation, error logging, and the generation and use of command buffers.

A command buffer is a series of keyboard commands residing on the disk. Each command buffer is referred to by a unique name. By calling and executing a command buffer, the tedious task of entering all the commands normally necessary to load and control a test are done in the sequence in which they occur in the command buffer without operator. intervention. Several tests can be loaded, executed, and controlled by one command buffer. These tests may be loaded and executed sequentially using a pass count to determine how long each test should run before halting it, loading the next test, and starting it. It is also possible to run one or several PPU/peripheral tests along with one or more CPU tests simultaneously, depending on the particular hardware configuration. For example, a command buffer could be written which ran all the tests needed to satisfy the preventive maintenance (PM) requirements for a particular day. By running the command buffer, one would be assured that no crucial test is omitted from that PM session. Command buffers can be constructed with input provided either via the card reader or by entering the commands from the console keyboard. Command buffers may also be modified via the keyboard. This course will include the procedures necessary to perform both creation and modification, as well as usage of command buffers.

Use of CMSE requires a 6000 or CYBER system with a tape unit and preferably a disk unit. Once executing, CMSE will use three PPs and up to five channels (three communication, one display, one disk). If these hardware components are not operational, the stand-alone test loader will be used in order to get the hardware to the level needed.

DISK

0000 ~~1 0000~~
0000 ~~27513~~
2400
2400
7700
0300
7400
7100 10
7301 11
0530 12
0000
0000

	TAPE
1	0000
2	75CC
3	0000
4	0000
5	77CC
6	E264
7	74CC
10	71CC
11	7301
12	SMRY

CC channel no.
E equip no.
U unit
S = ECS
M MEMORY
R TYPE CPU
Y operating system

LEARNING ACTIVITY 1-B. TEXT READING: DEADSTART PROCEDURES

As with prior diagnostic software systems, CMSE provides the options of deadstarting an 80-column absolute binary deck from the card reader or loading a program as a logical record from a magnetic tape or disk unit.

An 80-column binary deck would be a program which is to be loaded into PPU Ø. A program of this nature would be used for initial testing of the peripheral processors, central memory control, and central memory. The use of card loading is normally restricted to cases of machine failure resulting in the inability to deadstart at a higher level (tape or disk). The panel settings for card loading remain unchanged from those of SMM and can be found in the reference manual.

As discussed in a previous learning activity, MSL may exist either on a tape or on a disk and deadstart from either of these media is possible. MSL can reside on the disk in two different environments with, or separate from, the customer's operating system. The customer engineers may have a removable disk pack used exclusively for maintenance. It would be mounted on a drive whenever MSL was to be used. However, MSL may reside on the same pack as the customer's operating system. In the later case, the deadstart procedure is slightly different, although the deadstart panel setting is the same. In either case, before deadstarting from the disk can be accomplished, it is necessary to copy the MSL tape to the disk. The utility programs TDX (844, 885), TDY (841), and TDZ (808, 6603) are on the MSL tape to perform this task.

The deadstart panel settings are found in the reference manual. You will notice that all settings, whether 6000/C70, C170, active or inactive PP are very similar if deadstarting from RMS, MTS, or ATS. Embedded within these settings (panel word 12) are three octal digits used by CMSE to define CPU type, CM size, and extended memory size. This capability makes most editing via the keyboard unnecessary. Some additional information can be submitted via the deadstart panel, but this is used by the customer operating system when MSL resides on the same pack. There is a separate panel setting for 60X/65X. For this setting, word 13 contains the parameter information for use by CMSE.

Two additional deadstart panel settings are included in the reference manual. These are for the 66X/7021 coldstart and the 669-844/7152 coldstart procedures. Coldstart refers to the down-loading of controlware from a peripheral processor to a buffer controller. Buffer controllers are peripheral equipment controllers which differ from regular controllers in that they are programmable. Each one contains a memory, the logic required to execute the code from this memory, and information (connect/function/status codes) being sent out from a peripheral processor to drive the attached peripheral equipment. When the information in memory is destroyed, develops parity errors, or must be modified, the new data is sent out in the channel from a PP (downloading). This downloading is required at installation time, after a power-off, or when the currently executing controlware is different from that which is required by the present system user. If you had an MSL binary deadstart tape and wanted to write it on a disk for later deadstarting, you would use the following procedure.

CMSE Operation
Learning Activity 1-B

1. Deadstart CMSE from tape.
2. Load TDX, TDY, or TDZ and copy the MSL tape to a disk. You are required to enter a parameter which tells the program whether MSL is to reside on the same pack as the customer's operating system or not.
3. Change the deadstart panel to deadstart from disk. This will require a change in channel number(s) and the function to be used.

It was noted earlier that there is a difference in the deadstart procedure depending upon MSL disk residence (with or separate from the customer's operating system). The panel setting is the same for either case. When MSL is resident with the system, one additional display is presented to you after pressing the deadstart switch. This display is as follows:

```
CR - OS Load Automatic
 I - Deadstart w/Operator Intervention
 U - Utilities
 M - Off Line Maintenance
```

Enter an M to perform off-line maintenance, and you will observe the normal CMSE display. This CMSE display will be the only one presented if MSL exists on a pack separate from the operating system. An "M" is not displayed on MSL A loads.

RUNNING STAND-ALONE MODE TESTS

After a successful deadstart, you'll see a display much like the one illustrated in figure 1-1. The top of the display instructions explain how to load and run stand-alone tests. The following lines indicate that entering the three-letter test name followed by a carriage return will load the test into PPO from the library.

```
OPERATOR ENTRIES (FOLLOWED BY CR)

MNE-LOAD STAND-ALONE TEST MNE
```

Once the test is loaded, it changes the information on the screen. To run most stand-alone tests you will need to press the spacebar. In the next learning activity you'll run the peripheral memory test PMM in stand-alone mode.

System Configuration

The system configuration can be changed from that shown on the left-screen initial display by typing the number at the left of the description and the entry followed by a carriage return. For example, to designate the card reader on channel 12 equipment 4, the system card reader would make the following entry.

```
13.1204 (CR)
```

In the next learning activity, you'll also have an opportunity to make some system configuration changes, however first read through the section labeled "Deadstart Procedures and Routines" in the MSL Reference Manual.

```
                173    MAINT=SYSTEM V01
           COPYRIGHT CONTROL DATA CORP.  1977
DEADSTART STAND-ALONE TEST LOADER

OPERATOR ENTRIES (FOLLOWED BY CR)
   CARRIAGE RETURN TO LOAD SYSTEM
   MNE - LOAD STAND-ALONE TEST MNE
   TDX - TO DO TAPE TO 844 OR 885 DISK
   TDY - TO DO TAPE TO 841 DISK
   TDZ - TO DO TAPE TO 808 OR 6603 DISK
   CA - CLEAR SCR
   S/CBX - SET/CLEAR SCR BIT X
   KP - CLR PP MEM
   KC - CLR CM
   NN.ENTRY - CHANGE SYSTEM CONFIGURATION

       SYSTEM CONFIG

  1.   MACHINE TYPE                                      = 173

  2.   CM SIZE                                           = 05
       0=16K      1=32K      2=49K      3=65K
       4=98K      5=131K     6=198K     7=262K

  3.   EM SIZE                                           = 00
       1=1XXK   2=2XXK   3=5XXK   4=1M   5=1.5M
       6=2M..12=4M....22=8M......42=16M

  4.   NO. OF CPUS                                       = 01

  5.   NO. OF PP S                                       = 24

  6.   MON PP/IO COMM CH                                 = 01

  7.   PP COMM CH                                        = 03

  8.   DISPLAY PP/COMM CH                                = 02

  9.   SCAN CH (CY176 ONLY)                              = 00

 10.   1ST AND LAST PPU (CYBER 176 ONLY)                 = 0000

 11.   LOAD DEV TYPE *                                   = 00
        0=TAPE,    1=844-2X    2=844-4X    3=885
        4=841      5=808       6=6603

 12.   405 CH AND EQ                                     = 1304

 13.   405 MAC/DCC NO.S (MD)                             = 01

 14.   TAPE UNIT/DCC NO.S(UU0D) *                        = 0101

 15.   DISK CH AND EQ *                                  = 0100

 16.   DISK UNIT/MAC/DCC NO.S(UUMD)                      = 0100

 17.   STARTING CYL NO.                                  = 0565
```

Figure 1-1. Typical Left Screen Initial Display

*Entry not displayed if disk deadstart.

LEARNING ACTIVITY 1-C. CAI: DEADSTART

This learning activity is designed to give you practice deadstarting CMSE. You'll have an opportunity to set up the deadstart panel, run a stand-alone diagnostic, and make some system configuration changes by making console entries.

Sign on to the PLATO terminal, select block a, "Getting Started" by entering a and pressing the NEXT key at the block options page. At the block description page, press a to get an assignment and follow the instructions to complete this CAI activity, "Deadstart."

LEARNING ACTIVITY 1-D. TEXT READING: BASICS

This learning activity is designed to teach you what you will see on the console once CMSE is loaded. CMSE header information, the CMSE displays, and the use of special keys are also discussed.

CMSE HEADER INFORMATION

While at the initial CMSE display, pressing the carriage return alone will load the remaining CMSE system. Once the CMSE system is loaded, several displays will be available by entering a request for the display. These displays may be placed on either the right-screen, the left-screen, or both. There is, however, information on the right and left screen that is always displayed and is independent of the information you request. This information is called headers. Also independent of the display shown on the left screen is the keyboard entry area which consists of five lines at the bottom of the screen. Before continuing with this learning activity, read the following paragraphs in the chapter entitled "Display Commands" in your reference manual.

Left-Screen Header
Keyboard Entry Area
Right-Screen Header

CMSE DISPLAYS

Several displays are available once the CMSE system is loaded by pressing the carriage return key at the initial CMSE display. Your reference manual illustrates examples of each of these displays and indicates how commands are entered to request each display. Some of these displays may be altered, and these commands are also described. Before going on to the last part of this learning activity, read the paragraphs headed "Select Display" in the chapter entitled "Display Commands" and read the chapter entitled "Display Alter Commands" in your reference manual.

SPECIAL KEYS

The console keyboard includes several keys that need explanation (for example, the blank keys). Read this explanation in the introduction to the section entitled "CMSE Commands" in your reference manual. To complete this learning activity, read the remaining paragraphs in the chapter "Display Commands".

LEARNING ACTIVITY 1-E. CAI: BASICS

This learning activity is designed to give you a chance to see many of the CMSE displays and make entries to simulate the pressing of special console keys.

Sign on to the PLATO terminal, select block a, "Getting Started" by entering a and pressing the NEXT key at the block options page. At the block description page, press a to get an assignment. When you receive your assignment, follow the instructions to complete this CAI learning activity, "Basics."

BLOCK 2
BASIC COMMANDS

The activities in Block 2 are designed to teach you several CMSE commands and EXC operation.

YOUR ASSIGNMENT

To receive your assignment, log on to the PLATO terminal. At the block options page, select Block 2, "Basic Commands" by entering b and pressing the NEXT key. At the block description page, press a to get an assignment or press t to take the test. Your current knowledge may enable you to skip some of the learning activities. Mark those activities assigned and check them off as you complete each activity. You may choose to do all of these activities, or you may choose to do some activities more than once.

Check	Activity	Description	Page
______	2-A	Text Reading: CM/EM/PP Commands. This activity teaches commands to manipulate CM, EM and PP memory.	2-3
______	2-B	CAI: CM/EM/PP Commands. This activity gives you practice making CM, EM, and PP commands.	2-5
______	2-C	Text Reading: Load and CPU Execution Commands. This activity teaches commands to load and execute tests.	2-6
______	2-D	CAI: Load and CPU Execution Commands. This activity gives you practice loading and executing tests.	2-8
______	2-E	Text Reading: EXC Loading and Execution. This activity teaches control of the EXC monitor program.	2-9
______	2-F	CAI: EXC Loading and Execution. This activity gives you practice running EXC.	2-11

LEARNING ACTIVITY 2-A. TEXT READING: CM/EM/PP COMMANDS

This learning activity is designed to present the commands used to manipulate information in central memory, extended memory, and peripheral-processor memory. Also presented are the commands required to execute and halt PP programs.

CM AND EM COMMANDS

CM and EM commands alter or move contents of central memory and extended memory. All the addresses specified within the command that you enter are absolute.

Blocks of words can be set or cleared by the KC command for central memory and by the KL command for extended memory. Parameters for these commands include the first word address, the last word address +1 and the data you want in the block.

One word of data may be changed by using the EC command for central memory and the EL command for extended memory. The only parameters needed for these commands are the address of the word and the data you want to put into the word. When entering instructions in a central memory word, you will often want to enter an instruction in one byte leaving the rest of the word alone. The EB command provides this capability. For this command, you must specify not only the address of the word and the data, but also the byte number and the byte size.

When setting up an input exchange package in memory, you will need to set different parts of memory for the A register position, the B register position, and the P register position. These entries are accomplished by the XA, XB, and XP commands. The only parameters you need with these commands is the central memory address and the data you want stored.

PP COMMANDS

PP commands enter or transfer data in PP memories and execute or halt PP programs. The EP command is used to enter data into memory. Parameters required are the PP number, the address where you want the data to go, and, of course, the data you want to enter. Just like CM and EM, blocks of memory in PPs can be set or cleared. The KP command requires the PP number, the first and last word address+1, and the data to be stored as parameters. The contents of one PP can be copied to another PP by entering an MP command. Of course the two PPs involved must be named in the command.

You can choose not to have the activity of a PP monitored by the monitor program (usually in PP 1) by entering DN followed by the PP number and then a carriage return. To resume monitoring a PP, you enter UP followed by the PP number then a carriage return. PP errors may be cleared with a CE command, and if a PP is indicating a failing (F) status, you can deadstart it via a DP command.

To start a PP running, enter a RU command followed by the PP number and the starting address. You can halt a program by entering HP followed by the PP number.

The commands presented in this activity are described in detail in the paragraphs entitled "Central Memory/Extended Memory Commands and PP/PPU Commands" in the reference manual. Before going to the next activity to practice these commands, spend a little time going over this material to familiarize yourself with the command formats. Throughout this course you may want to reference the small plastic card of command formats included with this course.

LEARNING ACTIVITY 2-B. CAI: CM/EM/PP COMMANDS

This learning activity is designed to give you practice entering CM, EM, and PP commands while at a 17X console.

Sign on to the PLATO terminal, select block b, "Basic Commands" by entering b and pressing the NEXT key at the block options page. At the block description page, press a to get an assignment. When presented with your assignment, follow the instructions to complete this CAI learning activity, "CM/EM/PP Commands."

LEARNING ACTIVITY 2-C. TEXT READING: LOAD AND CPU EXECUTION COMMANDS

This learning activity is designed to present the commands to load programs and overlays into PP memories, central memory, or extended memory and initiate execution of tests.

LOAD COMMANDS

If you have a PP program punched in the standard binary format on cards, the load command LP followed by the PP number, the first word address, and any optional parameters will load the program into the PP memory. Standard binary formatted cards may also be loaded into central memory and extended memory using the LC and LL commands respectively. These two commands require only the first word address as a parameter. Programs can be brought into PP or CP memory from the library with the CP and the CC command respectively. With these commands you'll need to enter the name of the program as a parameter. Some PP tests are too long to fit in a PP memory and need overlays to run. When calling these tests, the overlays can be either copied into central memory at load time or read from the library as they are needed by the test.

The LT commands load the test you name along with any parameters included into the specified PP. If faster access to the overlays is preferred or when loading from tape, the TL command can be used in place of the LT command. The only difference being that the overlays are copied into central memory when the test is loaded into the PP and loaded from there as needed by the test.

CPU EXECUTION COMMANDS

Later on in this course, you'll learn about EXC commands that are used for CPU test controls. It is possible, however, to set up your own exchange package and run keyed-in programs without loading EXC. Memory can be changed as required using the commands you learned in the last two learning activities. Once you have memory set up, a DC command will start the CPU. For this command, you'll have to indicate the CPU number and the first word addresses of the input and destination exchange packages. You can halt the program by entering the HC command. You may specify the address of the destination CPU exchange package if desired. Because sometimes an instruction fails in such a way that program execution fails to complete, it is nice to have a command to repetitively start the CPU, the RD command. For this command, you'll have to enter the first word addresses of the input and destination exchange packages, the number of CM words to be transferred if not 20_8, and the octal time in microseconds between successive starts.

If your CPU program makes system calls to the monitor or you want the monitor to record status errors in central memory, you can use a command (the CN command) to turn on this capability. The parameters for this command include a CPU number and the first word address of the PU/CMSE interface block in CM. Once this capability is turned on, it will stay on until turned off by a CF command. Before going on to the next learning activity, a CAI activity where you'll have an opportunity to write and execute a CPU program, read over the paragraphs headed "Load Commands" and "CPU Execution Commands" in the reference manual.

LEARNING ACTIVITY 2-D. CAI: LOAD AND CPU EXECUTION COMMANDS

This learning activity is designed to give you practice entering CMSE commands to load programs and tests as well as CMSE commands to start execution of CPU programs and tests. Sign on to the PLATO terminal, select block b, "Basic Commands" by entering b and pressing the NEXT key at the block options page. At the block description page press a to get an assignment. When you receive your assignment, follow the instructions to complete this CAI learning activity, "Load and CPU Execution Commands."

LEARNING ACTIVITY 2-E. TEXT READING: EXC LOADING AND EXECUTION

This learning activity is designed to introduce you to the EXC (Executive) program. EXC is a PP-based, CPU monitor program which loads and controls from 1 to 4 CPU tests. You will use EXC any time you want to run a CPU diagnostic. It will format exchange packages for the tests requested, and when these programs are executing, EXC monitors the testing process. EXC provides the user with debugging capabilities for use upon detection of an error.

EXC LOADING

As previously mentioned, EXC is a PP program. Therefore, you will have to use one of the commands for loading a PP program. You should use the CP or LT command when loading EXC. Four optional parameters exist to allow selection of CPU(s), exchange address, and field length. Once loaded, assign the display (PPXX Command) and then you can start the PP executing by entering a RU command.

EXECUTION

When you have EXC running, it will generate a display describing the commands and parameters you may now use to control its execution. In all, three displays are generated by EXC. These are called the parameter, (above) running (normal), and running (fast) displays, and are selectable by using a H, N, or T respectively. The normal running display includes the CPU test names, A, P, RA, and FL register, an exchange package, a pass count, and an error count for each selected CPU test. Assume you want to run the two CPU tests, ALX and CT3. Keep in mind that you may run from one to four tests. Observe the H display, and by utilizing the L command, you can load the two tests ALX and CT3. The N display is automatically selected as the tests load. You can now start the tests running by depressing the space bar. Successful execution of the CPU tests is indicated by a changing P register and no error message. You may want to halt test execution and depressing the S key will do this. Any test that detects an error will halt. However, error halting is selectable and toggles between on to off each time you depress the E key.

When a test fails, or you desire to manipulate any CPU program, you will be using some additional EXC features and commands. You may set a breakpoint address within your CPU program with a BKP command. The only required parameter is the central memory address where you want to breakpoint. When a program running under control of EXC accesses this breakpoint address, it will stop. This is accomplished because EXC sets the breakpoint address equal to 00--00 (program stop). When this stop is executed, EXC restores the original contents and exchanges the program out of the CPU. The P register of the program will equal the breakpoint address. To utilize breakpoint, you will want to view the input and output exchange packages (X display) and the program to be executed (C or D display). When you have the displays selected, the input exchange package properly constructed, and the program in central memory, you can enter the breakpoint address and depress the space bar. The program runs from P in the exchange package to the breakpoint address, at which point execution terminates and the CPU is exchanged out. The output package contents should reveal whether or not the program executed correctly up to the breakpoint address. Should you desire to run the program again, use the space bar. The original input exchange package is used and execution will be up to the breakpoint address. By entering the test command at this point, followed by a space bar, this operation will repeat (that is, exchange in, run until breakpoint, exchange out). Exchange in, run, and so on. Since you will be able to see the program results in the output exchange package between each restart, they can easily be checked for correctness or an intermittent result. Again, operation is terminated with the S key.

The step mode of operation exists to assist you in isolating CPU failures. It is selected by entering STEP on the keyboard. You can set up to run in this mode just the same as you would for test mode. The program will execute exactly as before when you start it. The difference exists when the breakpoint address is reached. The CPU stops as before, but when the space bar is depressed the second time, the CPU is started using the output exchange package information instead of the input as was previously the case. In addition, the breakpoint address is automatically incremented by one. This means that only one 60-bit CPU instruction word will be executed before the breakpoint is reached. The operation continues this way each time the space bar is depressed, allowing you to check results after the execution of one CPU instruction word.

A complete write-up of EXC is found in section 6 of the reference manual. Read the paragraphs on EXC before proceeding to the next learning activity.

LEARNING ACTIVITY 2-F. CAI: EXC LOADING AND EXECUTION

This learning activity is designed to give you practice making the entries required to load and run CPU tests via EXC. You'll also have an opportunity to change modes and run the computer in the different modes.

Sign on to the PLATO terminal, select block b, "Basic Commands" by entering b and pressing the NEXT key at the block options page. At the block description page press a to get an assignment. When you receive your assignment, follow the instructions to complete this CAI learning activity, "EXC Loading and Execution."

BLOCK 3
COMMAND BUFFERS

The activities in Block 3 are designed to teach you library and command buffer commands and how to write and execute command buffers.

YOUR ASSIGNMENT

To receive your assignment, log on to the PLATO terminal. At the block options page, select Block 3, "Command Buffers" by entering c and pressing the NEXT key. At the block description page, press a to get an assignment or press t to take the test. Your current knowledge may enable you to skip some of the learning activities. Mark those activities assigned and check them off as you complete each activity. You may choose to do all of these activities, or you may choose to do some activities more than once.

Check	Activity	Description	Page
______	3-A	Text Reading: Library and Command Buffer Commands. This activity teaches the commands needed to change the library and create and maintain command buffers.	3-3
______	3-B	CAI: Library and Command Buffer Commands. This activity gives you practice making library changes and creating command buffers.	3-5
______	3-C	Text Reading: Write and Execute Command Buffers. This activity teaches procedures required to construct and manipulate command buffers on disk.	3-6
______	3-D	CAI: Write and Execute Command Buffers. This activity gives you practice creating command buffers.	3-8

LEARNING ACTIVITY 3-A. TEXT READING: LIBRARY AND COMMAND BUFFER COMMANDS

This activity is designed to teach you how to make changes to the library and describes commands used to create and maintain command buffers and programs on the CMSE RMS device.

LIBRARY COMMANDS

A program in a PP memory or the CP memory can be written to the disk or tape library with *WP and *WC. For the *WC command, you will have to indicate the name you want the program to be called and the first and last word address+1 of the program. The *WP requires the PP number and the name of the program. If you write a program in COMPASS, the punched output can be read and put on the library with an *EP command. A program on the library may be deleted by a *DP command. Of course you'll need to name the program you want to delete. To document what you are doing, you can make an entry into the dayfile. The dayfile consists of error and information messages generated by the various tests and routines within CMSE and can be listed on a printer. The monitor requires overlays to be loaded at certain times. If these overlays are on tape, this greatly increases the time required to run. To have these programs loaded into central memory where access is considerably quicker, you can enter the *OV command. The only parameter required is to state where in memory you want the overlays loaded. Command buffers, lists of command, if punched in Hollerith format on 80-column cards, can be read and written to the library with the *AB; there are no parameters. Command buffers may be deleted from the library with a *DB command. The name of the command buffer to be deleted is a parameter. To clear the command buffer area on the disk, use an *DB *. (Note the space between the B and the *)

COMMAND BUFFER COMMANDS

The GO command is used to load and execute the commands in a command buffer. The name of the command buffer is the only parameter. A TB command terminates a command buffer and a comment command (a space followed by a comment) will halt the processing of the commands until the GO command is entered. The GO command which is used to continue a command buffer, needs no parameters. The SQ command halts the execution of the commands in a command buffer until a memory location is equal to a desired pass count. The parameters for this command are the address of the desired pass count, the PP number or CM indicator, the pass count, the address of the monitored word, and a comment. Execution of commands can also be halted temporarily by the MG command. A parameter specifying the amount of time needed to delay is required. Many tests require the press of the space bar to start execution. The SP command provides this capability and makes command buffers automatic.

CMSE Operation
Learning Activity 3-A

COMMAND BUFFER EDIT COMMANDS

A command buffer brought up in the E display can be modified by two commands, the IN command inserts a command and the DE command deletes or replaces a command. Both commands require an ordinal number as parameters and the new command, if there is one.

The commands mentioned in this learning activity are described in the reference manual. Read over the material presented in the chapters entitled, "Library Commands," "Command Buffer Commands," and "Command Buffer Edit Commands" before continuing to the next learning activity.

LEARNING ACTIVITY 3-B. CAI: LIBRARY AND COMMAND BUFFER COMMANDS

This learning activity is designed to give you practice using the library and command buffer commands while at a 17X console.

Sign on to the PLATO terminal, select block c, "Command Buffers" by entering c and pressing the NEXT key while at the block options page. At the block description page, press a to get an assignment. When you receive your assignment, follow the instructions to complete this CAI learning activity, "Library and Command Buffer Commands."

LEARNING ACTIVITY 3-C. TEXT READING: WRITE AND EXECUTE COMMAND BUFFER

This learning activity describes the procedures required to construct and manipulate command buffers on the disk. There are three sections to this activity: creating a command buffer with a card deck, creating a command buffer at the console, and command buffer modification.

CREATING A COMMAND BUFFER WITH A CARD DECK

The first card of command buffer deck must have *AB punched in the first three columns with the remaining columns blank. You may recall the *AB command is used to add a Hollerith command buffer deck to the library.

The second card of the deck and the first card of every command buffer within the deck must contain a $ in column 1 followed by the name of the command buffer following. A period must be placed just behind the command buffer name.

The remainder of this card and all the cards following the command buffer name card are cards containing individual commands that make up the command buffer. Each card may contain several complete commands each followed by a period. These cards do not require a $ in column 1.

The last card in the deck can be either an EOR or an EOF card.

The cards will be written onto the library by putting them into the card reader, making it ready, and entering CR, ON (CR) at the console while CMSE is running.

CREATING A COMMAND BUFFER AT THE KEYBOARD

The first step in creating a command buffer is to enter an AE or a BE command with the name of the command buffer as a parameter.

As an example, AE, ABC would result in a command buffer named ABC. The second step is to enter INØ followed by the first command.

The third step is to press the forward key, and enter the next command which repeatedly enters successive commands. After entering the last command, press the CR key. The completed command buffer is then written to the library with a *WB command. The DE command can be used to delete or replace commands just before writing the command buffer to the library.

COMMAND BUFFER MODIFICATION

Once the command buffer is written to the library, it must be read into the edit file with the AE or the BE command with the command buffer name used as a parameter. The IN and DE commands are used as if they were originally creating a command buffer from the keyboard. When modification is complete, the modified command buffer is written to the disk using the *WB command. If the name of the command buffer is not changed, it will be necessary to delete the original using the *DB command before writing the modified version to the disk.

The procedures for command buffer operation and modification are described in the reference manual, section 4. Review this section before going on to the next learning activity.

LEARNING ACTIVITY 3-D. CAI: WRITE AND EXECUTE COMMAND BUFFERS

This learning activity is designed to give you practice writing and running command buffers. Included in this activity is a command buffer to load and run a CPU diagnostic via EXC. Sign on to the PLATO terminal, select block c, "Command Buffers," by entering c and pressing the NEXT key while at the block options page. At the block description page, press a to get an assignment. When you receive your assignment, follow the instructions to complete this CAI learning activity "Write and Execute Command Buffers."

For the last part of this activity, you'll be asked to write and execute a command buffer called EXC3. Please enter and run the following command buffer:

```
EXC3
HP3
CP3, EXC
PP3
EP3,75,0
RU3,100
SQ75,1,3
L,ALX,CT3
SQ75,1,3
SP
```

BLOCK 4
SPECIAL OPERATIONS

The activities in Block 4 are designed to teach you the miscellaneous commands, some of the utilities you will be using, and the PP programming interface with CMSE.

YOUR ASSIGNMENT

To receive your assignment, log on to the PLATO terminal. At the block options page, select Block 4, "Special Operations" by entering d and pressing the NEXT key. At the block description page, press a to get an assignment or press t to take the test. Your current knowledge may enable you to skip some of the learning activities. Mark those activities assigned and check them off as you complete each activity. You may choose to do all of these activities, or you may choose to do some activities more than once.

Check	Activity	Description	Page
______	4-A	Text Reading: SCR, CR, Channel Control, 176 Commands. This activity teaches how several miscellaneous commands are used.	4-3
______	4-B	CAI: SCR, CR, Channel Control, 176 Commands. This activity gives you practice using several miscellaneous commands.	4-5
______	4-C	Text Reading: Utilities. This activity gives a general description of several utilities used with CMSE.	4-6
______	4-D	CAI: Utilities. This activity gives you practice running utility programs used with CMSE.	4-9
______	4-E	Text Reading: PP Interface. This activity teaches how to make your PP program interface with CMSE.	4-10
______	4-F	CAI: PP Interface. This activity gives you practice writing PP programs that interface with CMSE.	4-12

LEARNING ACTIVITY 4-A. TEXT READING: SCR, CR, CHANNEL CONTROL, 176 COMMANDS

The card reader is used by CMSE for reading in programs or command buffers for addition to the current library. Two commands control reader monitoring. You control monitoring by entering CR, ON or CR, OFF.

You may have a situation requiring manual manipulation of an I/O channel which is hung-up due to either hardware or software failure. CMSE provides commands to activate, deactivate, function, input, and output in a channel. All these commands require you to enter the channel number and, in the case of a function or output, it will be necessary to provide data.

The status and control register (SCR) commands are provided for manipulating the register bits. It should be remembered that all functions described here are accomplished by clearing and/or setting SCR bits. Also, SCR commands do not apply to 6000 or CYBER 70 machines. You can set a breakpoint address in the SCR using the BK command. Along with the address, there is a parameter to define the condition(s) to be met before breakpointing occurs. Three commands provide you with a means of directly clearing SCR bits. The CA command clears all bits allowable by the hardware. The CB command will clear only the bit which you define as a part of the command. If you want to clear the error bits for CM, EM, and channel parity errors, use the CE command. Individual bits can also be modified with the SB (set one bit) and the SC (set-clear one bit) commands. If a PP indicates a failing status (F), you are able to do a force exit on that PP via the FE or DP command. It is necessary to supply the number of the failing PP as a parameter.

Running diagnostics with some form of margins applied is a part of all preventive maintenance. CMSE provides a number of commands for applying, controlling, and monitoring these hardware margins. You can vary clock margins with the CM command. Parameters are required to define clock pulse width and frequency. When you are testing a 175 or 176 type machine, reference voltage margins may be applied to the logic. The MA, MS, and MQ commands are used to define the physical modules to be tested, the application of high or low margins, and the address in central memory to be monitored for a failure. This information is all entered as parameters when formatting each command. Every selected module can have the reference voltage margin applied to it for a varying length of time. You control this by use of the VS command.

The LE command controls logging EM and CM errors to the RMS device. The parameters you enter include the type of memory to monitor and which error type to record. For PP errors, enter PM and the number of the PP to be monitored. Once an error is detected, module scanning halts at the current position.

CMSE Operation
Learning Activity 4-A

There are a group of commands for use exclusively on a CYBER 176. They have no effect on any other machine type. With the CL command, the size of EM can be modified. The desired size and configuration are entered as command parameters. You can modify exchange package biasing by entering an EX command accompanied by the desired bias. The mode for CM and EM operation is controllable with a pair of CMSE commands. You may choose to run in maintenance mode or in parity mode, and can determine the modes with the MM and PA commands respectively. The required parameters are the selection of affected memory (CM or EM) and whether to enable or disable the mode. The RS (restart operating system) and SY (system interrupt) commands provide some interface between CMSE and the customer's operating system and are used primarily by the operating system.

To complete this learning activity go over the commands discussed here in the CMSE section of the MSL Reference Manual.

LEARNING ACTIVITY 4-B. CAI: SCR, CR, CHANNEL CONTROL, 176 COMMANDS

This learning activity is designed to give you practice entering SCR, CR, channel control, and 176 commands.

Sign on to the PLATO terminal, select block b, "Special Operations" by entering b and pressing the NEXT key at the block options page. At the block description page, press a to get an assignment. When you receive your assignment, follow the instructions to complete this CAI learning activity, "SCR, CR, Channel Control, 176 Commands."

LEARNING ACTIVITY 4-C. TEXT READING: UTILITIES

Part of MSL consists of a group of programs called utility programs. These programs do not test any piece of equipment nor are they a part of CMSE control or system programs. The utilities serve as an aid to the user in creating problems and diagnosing them.

CMC - Central Memory Conflict Program

CMC performs either READS (default) or WRITES of central memory from the PP into which it is loaded. As its name implies, its primary purpose is to produce conflicts with central memory and central memory control, thereby increasing the probability of hardware failure. It is often executed in more than 1 PP and simultaneously with central memory or CPU tests. Default execution is a 1000 word read of central memory with no checking of data. Other possibilities are selectable at load time when the display is assigned to the PP containing CMC. If selecting central memory writes, caution must be exercised to avoid clobbering the central memory being used by a CPU program.

DDD - Deadstart Dumping Routine

This dumping routine loads via the stand-alone loader. (Enter DDD [CR]). It is used to dump PP and central memories. Dumping is usually to the printer, however, central may also be dumped to a mag tape for later retrieval (also via DDD). Loading a 512 or 580 image memory and a 580 PFC memory is also possible with DDD. Operation is very straight forward and controlled via the console. After entering the printer channel and equipment numbers, the desired operation is selected via a single key. Several operations can be performed without a deadstart, but deadstarting is required to load a new program.

DMP - Dayfile and Memory Dump

This dumping routine runs under CMSE and performs its dumping, and so on, without a deadstart. All the entries for dumping are displayed by the PP containing DMP. The following are possibilities for dumping.

PP Memory

Current A Display Contents

Current B Display Contents

Central Memory using absolute first and last addresses

Central Memory using relative addresses

Extended Memory using absolute first and last addresses

Extended Memory using relative addresses

Contents of the CMSE Dayfile

Any file which exists on the system

There are about eight messages which are generated during the dumping process. These are all self-explanatory and can be found in the CMSE reference manual.

LDC - Tape to Disk Controlware Loader for the 7152

The 7152 is a combined disk/tape controller. It requires controlware to perform its function. There is a PROM (programmable read only memory) contained in each half of the controller. It is possible to load controlware into either controller via a PROM coldstart load function. The MTS controlware would be loaded from a tape and the disk controlware from a disk pack if this PROM coldstart load function were performed. The purpose of LDC is threefold.

1. Load MTS controlware from a magnetic tape to the MTS buffer controller. This is accomplished by performing an MTS PROM coldstart function.
2. Load disk controlware from a 9-track tape unit (by using the MTS buffer controller) into the 844 buffer controller.
3. Load disk controlware from a 9-track tape unit, reformat it, and write it to the disk. It would now be possible to perform an 844 PROM coldstart at a later time should it be necessary to reload controlware.

At load time, all the parameters are displayed, and no further explanation will be given here. Should any error occur during the execution of LDC, an appropriate message is displayed. There is also a DDLT in the CMSE manual to assist in using LDC.

RMD - Remote Assistance Driver

RMD is a PP program that interfaces the CMSE monitor to a terminal. Once loaded and running at the local site, RMD allows two-way communications between the local console and a remote terminal. With RMD, it is possible to do any operation which can be normally performed from the local console. In addition, there are some commands that are unique to terminal operation. These are listed in the CMSE manual and, in general, deal with defining the source and amount of output to be sent to the terminal. These commands may be viewed at the terminal by entering HELP.

CPMTR - Central Processor Monitor for all Machines Except the CYBER 176

CPMTR is loaded by EXC when the latter is initialized. This monitor provides the user with two useful tools. It should be kept in mind that many problems occur in the machine because of the exchange jumps which usually occur very frequently during the normal execution of the customer operating system. CPMTR provides a means of increasing the exchange activity during maintenance, thereby improving the chance of producing failures. Error CEJs are also handled by CPMTR and make looping on an error exit condition possible.

The exchange conflicts can be produced by performing a CPU exchange with a MA set equal to either 340 (CPU0) or 360 (CPU1). This executes CPMTR. CPMTR does a CEJ to (Bj + k) to return to the currently executing program at address+1 of the CEJ which was executed to produce the conflict.

Should an error condition be detected in the CPU (and CEJ is enabled), an error sequence is initiated. After the storing of error information at RA or RA+1, a CEJ occurs. CPMTR detects P = 0 or 1 (an error occurred), stores the output package, and enters an idle loop. The program could now be restricted, which would probably run to error exit again. By repeating the above, a looping condition exists and scoping will be possible. CMSE is informed whenever an error CEJ occurs.

CPUM - CPU Monitor for the CYBER 176

CPUM serves generally the same purpose for the 176 as CRUMTR does for the remaining machines.

STM - Stand-Alone Test Manager

STM is actually not a utility program. It is a control program or interface for 170 diagnostics which execute in a stand-alone mode. It provides the user with the following features.

1. Loads tests in PP1
2. Monitors and processes keyboard commands
3. Drives the display
4. Monitors the SCR to provide information to the running test

STM does not interface all stand-alone tests. Most tests written specifically for a 170 machine will be using STM. Other older tests will not. STM has a considerable repertoire of commands for the displays, test control, test execution, SCR control, and so on. These are in the CMSE manual and will not be covered in this text.

UTL - Hardware Interface Utility

This utility provides the user with a method of interfacing with the hardware for the purpose of entering a hand-written program into a PP. It also provides SCR interface, CM communication, inter PP communication, and various displays for the user. UTL has many commands available. These commands are the same as, or very similar to, those of STM and will not be discussed here.

All these utilities are covered in greater detail in the MSL Reference Manual. Refer to it if you would like.

LEARNING ACTIVITY 4-D. CAI: UTILITIES

This learning activity is designed to give you practice running some of the utility programs included on the maintenance system library. Sign on to the PLATO terminal, select block d, "Special Operations" by entering d and pressing the NEXT key at the block options page. At the block description page, press a to get an assignment. When you receive your assignment, follow the instructions to complete this CAI learning activity, "Utilities".

LEARNING ACTIVITY 4-E. TEXT READING: PP INTERFACE

The following information will help you understand the necessity for interface or service routines when dealing with PPs. It is necessary to see what you are doing. The display is treated as a normal I/O device and therefore must be driven by a PP. Each PP must be started before it can drive the display console, or for that matter, communicate with the operator or another PP. The interface routines supply the user with a display driver, a keyboard processor, and control routines for communication and channel control. The user is presented with everything he/she needs to enter a machine language program without the necessity of doing the housekeeping work.

All service routines require a certain amount of memory space within the PP to perform these tasks. This area of memory is not available to the user. In addition, each routine has fixed entry points for performing tasks for the user. The general flow within a PP while executing a user program would be 1) display screen(s), 2) check for keyboard input, 3) jump to execute user program, display, and so on. If the user program hangs the PP, it is not possible for it to return to the display driving code, and consequently, a blank screen results. When the execution of the user program is halted, monitoring of displaying and keyboard continues, but no jumping to execute the user's program is performed. This code, which is loaded at deadstart time, is referred to as the PP resident. The contents of the resident and its usage of core can be found in the reference manual. This information is important because it defines the areas of PP core available to the user.

This PP resident can be used, when it is desired, to enter a machine language program into a PP. The first step in this procedure is to obtain a PP from CMSE and display its memory. This could be accomplished with the following commands (assume PP #6).

DP06 - Destarts PP6 and loads the resident.

PP06 - Assigns the right screen to PP6. Since PP6 will not have any information to display, its memory will be displayed.

Now the machine language program can be entered. Assume that it will start at memory location 3000:

D13,3000	Displays addresses 3000 - 3077 in the upper block, right screen.
EPb,3000,xxxx	This enters the xxx data into location 3000. If this is entered via the forward key, the address will update to 3001, thus saving keyboard entry time.

Continue entering data until the entire program has been written into the PP.

It will be necessary to terminate execution of the program by interfacing with the resident. If only one pass through the program is desired, the last two instructions should be a long jump to 7606, (0100, 7606). To loop on the program, the last two instructions should be a return jump to 7604, (0200, 7604). With contents of A=O.

To start the program, enter RU6, 3000. This starts the program running and stores the starting address at memory location 7604. This is done so that when looping is required, the resident will know where to restart the program each time. Once looping, the PP program would be halted by entering:

HP6

The resident has the ability to call upon CMSE to perform certain functions. The procedure for this is not covered here, but basically, the call is formatted within the PP. The address of the call is entered in the PP A register, and a jump to the resident is performed. When running a machine language program similar to the one described above, it is mandatory that the A register always equals 0000. When jumping to the resident, this avoids an unwanted function request to CMSE. These function requests can be accomplished either via a 0100 - 7600, or more commonly via a 0200 - 7604. Each will accomplish the function request.

Example:

This program loads the A register, stores it, and halts.

```
    3000 - 1410     Load A with 000010

    3001 - 3440     Store into location 0040

    3002 - 0100

    3003 - 7606     Jump to resident and don't return
```

This program would be started with an RU, 3000 Command.

LEARNING ACTIVITY 4-F. CAI: PP INTERFACE

This learning activity is designed to give you practice writing PP programs that interface with CMSE.

Sign on to the PLATO terminal, select block d, "Special Operations" by entering d and pressing the NEXT key at the block options page. At the block description page, press a to get an assignment. When you receive your assignment, follow the instructions to complete this CAI learning activity, "PP Interface."

APPENDIX A
INDIVIDUALIZED INSTRUCTION

CONCEPT

The term "individualize" means to adapt to the needs or special circumstances of an individual. Individualized instruction is a method of instruction which takes into consideration your current ability, knowledge, rate of learning, and other important factors. It then presents the material in a way that allows you to work at your own rate of speed. By working independently you can focus on areas where you need the most instruction while other students are working in the areas where they require instruction.

In the traditional classroom an instructor presents all of the material to a group of students at a single location although only a few really need all of it. In comparison, individualized instruction does not require an instructor. You may bypass material that you already know. In many cases, you do not have to attend a formal classroom session. The role of the instructor is taken by an administrator who provides assistance and answers your questions. You may ask for assistance whenever you need it.

Pretesting indicates what material you already know and, therefore, can bypass. Results of the Pretest determine the assignment of lessons that you need to complete the course. You do not waste time on material that you have previously learned. Materials for individualized instruction are easily transported and can be studied at any remote location.

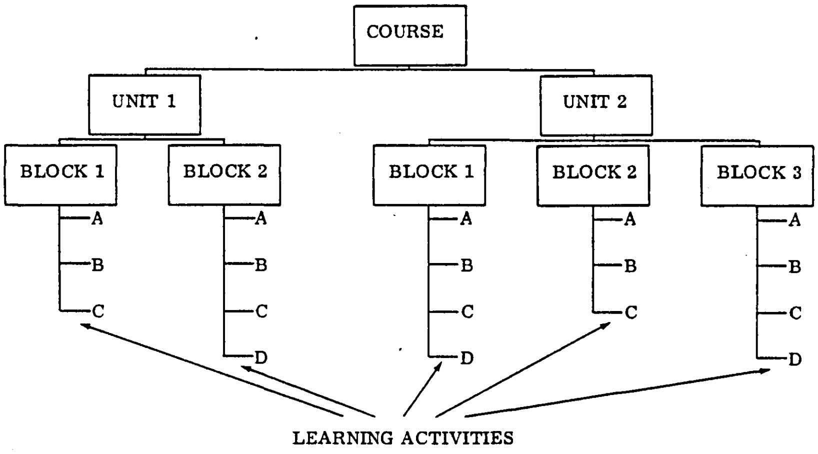
This is just a brief introduction to individualized instruction. The following paragraphs give you a more detailed picture. They also acquaint you with the terminology that is used in individualized instruction.

Space on the inside of the front cover of this manual has been provided for you to record the names of the various individuals who can assist you during the course. If you have not yet recorded these names, do so at this time.

COURSE STRUCTURE

Individualized courses usually have a hierarchical structure consisting of units, blocks, and learning activities. A course may have one or many units, depending upon the range of the subject matter. A unit covers one major topic of the course within a separate student manual. A course with only one major topic would require only one unit. Each unit contains a number of blocks (chapters). The average unit has at least two blocks and no more than six. Each block consists of a number of short lessons called learning activities. Each learning activity covers a specific aspect of the subject matter in the block. The learning activities use a variety of methods and media to present information in a helpful and interesting way.

The chart below gives an example of course hierarchy.



TYPES OF LEARNING ACTIVITIES

Learning activities use a variety of media.

- Text Reading - a reading assignment in your student manual
- Reference Reading - a reading assignment in a reference book or other text
- Videotape - a motion picture of the subject (with audio)
- Exercise - a written exercise in your student manual which provides practice in using previously learned concepts
- Audio-filmstrip - a slide presentation synchronized with audio
- Programmed text - material in your student manual that requires an immediate response
- Laboratory - hands-on activity to give you actual experience
- Progress checks - periodic checks (in your student manual) of your knowledge
- CAI - Computer-assisted instruction - lessons on the PLATO terminal, including graphics, animation, and simulation. It may require keyboard entries by the student, as requested by the PLATO terminal.
- CAI/Audio - A CAI lesson synchronized with an audiotape presentation
- Audiotape - an audio message about the subject material
- Microfiche - audiotape/microfiche lesson
- Audio/text - audio message synchronized to pictures or graphs in the student manual

SIGN-ON PROCEDURE FOR A CBE COURSE

In order to begin the course you must first sign on (or log in) to the PLATO Computer System. Depending on where your terminal is located, you may first have to dial up the PLATO computer in order to establish communication between the computer and your terminal. If your terminal is on a direct connection, skip steps 2 and 3.

1. Turn power on.

*2. Dial the number of the PLATO system using the data phone. When the system answers, you will hear a tone. Perform the next step within 3 seconds after you hear the tone.

*3. Lift exclusion button (usually located on the telephone cradle). Place the receiver on the table or in front of the cradle. Do not hang up.

NOTE

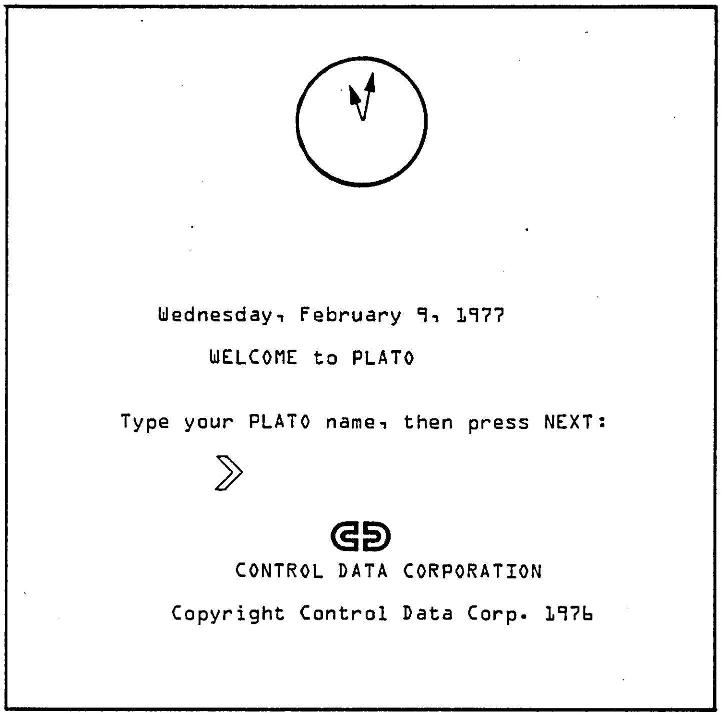
Some data phones require that you lift the exclusion button to dial and hang up when you hear the tone.

4. Press the STOP key on the PLATO terminal keyboard.

5. Press the NEXT key on the PLATO keyboard. This causes the "Welcome to PLATO" page to be displayed. (See Figure A-1).

6. Enter your name exactly as you are registered in the course. Normally this will be your last name, followed by a space, your first initial, another space, and then your middle initial. Do not use capital letters. Example: smith j v. (See Figure A-2.) If you make a mistake, erase it by using the ERASE key on the right side of the keyboard.

*Dial-up system only.



Wednesday, February 9, 1977

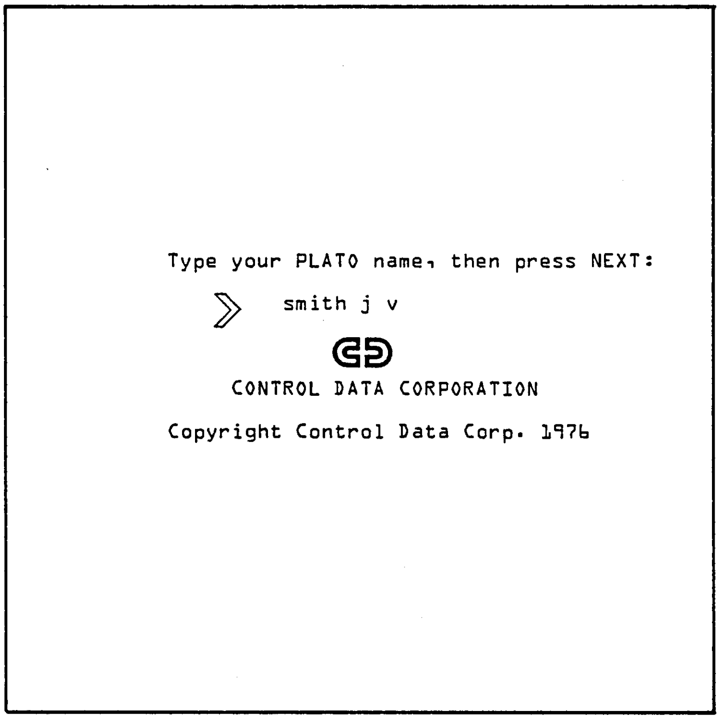
WELCOME to PLATO

Type your PLATO name, then press NEXT:

CONTROL DATA CORPORATION

Copyright Control Data Corp. 1976

Figure A-1



Type your PLATO name, then press NEXT:

smith j v

CONTROL DATA CORPORATION

Copyright Control Data Corp. 1976

Figure A-2

7. Press the NEXT key.

8. When asked to enter the course in which you are registered, enter it with the exact spelling of the name.

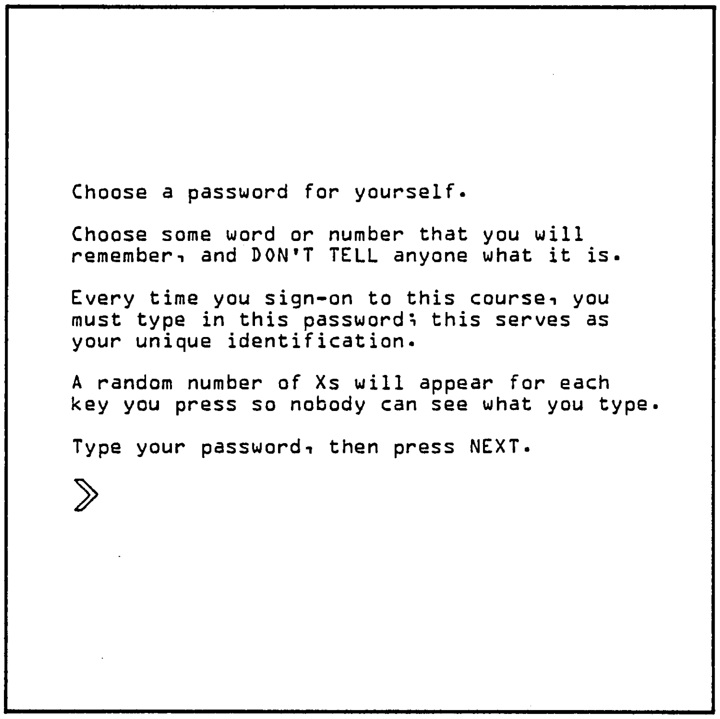
9. Hold down the SHIFT key and press the STOP key.

10. When asked to enter your password, enter it with the exact spelling. (See Figure A-3.) Your password can be anything that you choose, but it should be something that you can remember easily such as a name, phone number, employee number, and so on. Do not forget the password that you have selected. It is your key for gaining access to your lessons.

11. Press the NEXT key.

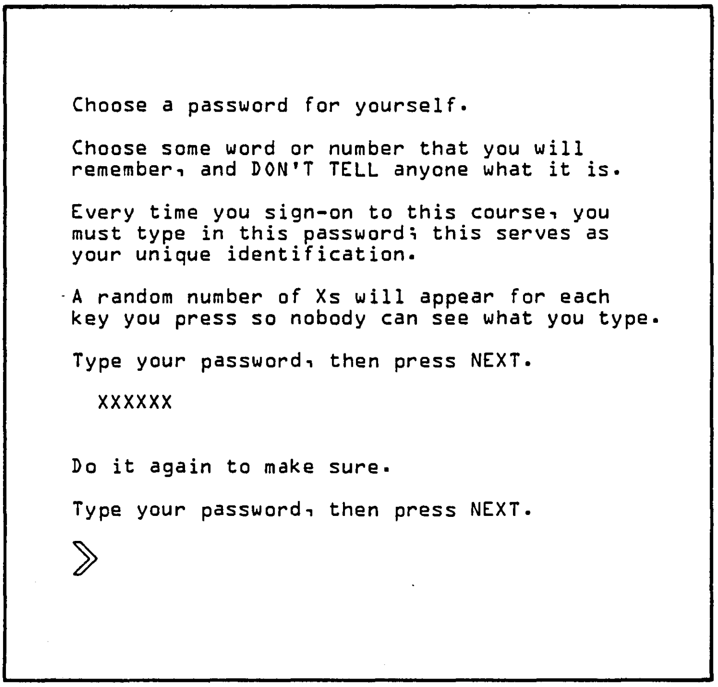
12. If this is the first time that you have signed on to this course, you will be asked to enter your password again. (See Figure A-4.) Enter your password again just as you did before.

13. Press NEXT. You are now signed on to the system. Follow the instructions on the PLATO screen. The PLATO terminal will now serve as your guide to the course.



```
Choose a password for yourself.

Choose some word or number that you will
remember, and DON'T TELL anyone what it is.

Every time you sign-on to this course, you
must type in this password; this serves as
your unique identification.

A random number of Xs will appear for each
key you press so nobody can see what you type.

Type your password, then press NEXT.

>
```

Figure A-3



```
Choose a password for yourself.

Choose some word or number that you will
remember, and DON'T TELL anyone what it is.

Every time you sign-on to this course, you
must type in this password; this serves as
your unique identification.

A random number of Xs will appear for each
key you press so nobody can see what you type.

Type your password, then press NEXT.

  XXXXXX


Do it again to make sure.

Type your password, then press NEXT.

»
```

Figure A-4

HOW TO OBTAIN HELP

Although this is an individualized course, there are people who can assist you if you run into problems. The various people and their function are:

Learning Center Administrator
Shows you how to sign on to the terminal at the Learning Center, operate audio/visual equipment, and locate required materials. Usually used only for CBE courses.

Course Administrator
Enrolls students and monitors student progress. May be able to answer questions pertaining to this specific course.

Technical Advisor
Has a high degree of knowledge about the course you are taking. If you have technical questions pertaining to the subject matter, this person should be able to answer them.

Laboratory Instructor
Arranges all necessary equipment for lab projects, helps during the lab, and ensures that the equipment is operational at the end of the lab project.

On the inside cover of this manual, write the names of the people who fill these positions if you have not already done so. In some cases, one person may fill more than one role.

COMMENT SHEET

MANUAL TITLE CMSE Operations

PUBLICATION NO. 75443587 REVISION A Prod. No. I5500

FROM: NAME:

BUSINESS ADDRESS:

COMMENTS:

This form is not intended to be used as an order blank. Your evaluation of this manual will be welcomed by Control Data Corporation. Any errors, suggested additions or deletions, or general comments may be made below. Please include page number references and fill in publication revision level as shown by the last entry on the Record of Revision page at the front of the manual.

CUT ALONG LINE

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

FOLD ON DOTTED LINES AND STAPLE

STAPLE STAPLE

FOLD FOLD

FIRST CLASS
PERMIT NO. 8241

MINNEAPOLIS, MINN.

BUSINESS REPLY MAIL
NO POSTAGE STAMP NECESSARY IF MAILED IN U.S.A.

POSTAGE WILL BE PAID BY

CONTROL DATA CORPORATION

Engineering Services Education
P.O. Box 0
Minneapolis, MN 55440

ATTN: EDUCATION QA
NTHESE

CUT ALONG LINE

FOLD FOLD

CORPORATE HEADQUARTERS, P.O. BOX 0, MINNEAPOLIS, MINNESOTA 55440

CONTROL
DATA

The MSL reference manual (PN# 60455770A) used with this course is not up to date with respect to the deadstart material. When deadstart information is needed, please refer to the enclosed sheets. Also note that the HT command has been changed to an HP command.

The deadstart procedures in this section are as follows:

- Hardware deadstart
- Deadstart options
- Loading procedures
- Deadstart panel settings

HARDWARE DEADSTART

Deadstart of the peripheral processor subsystem (PPS) consists of the following sequence of events.

1. The CPU is stopped, but the contents of all registers remain intact. This facilitates troubleshooting from a dump of central memory upon an error or failure.

2. Each PP is assigned to input on the channel that corresponds to its number (that is; PP 5 is assigned to input on channel 5). Channels 12 and 13 (and channels 32 and 33 on a 20-PP system) have no corresponding PPs, are not assigned, and are convenient for deadstarting.

3. All I/O channels are master cleared. A master clear removes all equipment selections except the deadstart panel, selects all data channel converters (6681's) in the system, and sets all channels to the active and empty condition. In the active and empty condition, the channels are ready for input. Channels other than 12, 13, 32, and 33 on 14- and 17-PP systems that are not associated with PPs are inactive and empty.

4. On a 6000/CDC CYBER 70, a master clear is sent on channel 0, through the equipment, and back to the deadstart synchronizer. On a CDC CYBER 170, a separate circuit clears the deadstart synchronizer without going through external equipment.

5. Each PP is set to execute an Input (71) instruction.

6. Address 0000 is cleared, the P register is cleared, and the A register is set to a value of 10000 octal. On a CDC CYBER 170, bits 80 through 95, 120 through 127, and 141 through 143 of the status and control register (SCR) are cleared.

The magnetic tape deadstart procedure is used when the library is on a deadstart tape and before it has been transcribed to a disk or other mass storage device.

The first program loaded at deadstart time is a PP program (CEY) which loads the command test (CED). CED is a modified version of the diagnostic PC1 which hangs PP 0 (0300 instruction) if it detects an error condition.

NOTE

> If the deadstart panel settings are correct and a deadstart leaves the display screen blank, CED or CEY probably detected a failure in PP 0. Refer to the description of CED and note the stop locations for CEY specified below.

TABLE 2-1. CEY STOP LOCATIONS

Location	Description
7422	LDD failure (A)=0000
7425	LPC failure (A)=0000
7427	SHN failure (A)=0000
7431	PJN failure (A)=0000
7435	LDN failure (A)=0000
7440	STD failure (A)=0000
7445	ADC failure (A)=0000
7450	LJM failure (A)=0000
7451	LJM failure (A)=0000

Upon successful completion of CED, the test program loads the monitor tape driver (CET) into PP 0 and starts it running. CET then presents the initial display which permits the operator to make system parameter changes.

Deadstart from Disk Library

The MSL with the disk driver resides on a disk with all the programs and overlays residing on the disk. CMSE can be loaded directly from an 885 or 844 disk through the PPS deadstart panel. A tape deadstart is required before loading CMSE from an 841, 808, or 6603 disk.

7. A zero word is transmitted, followed by the 12 or 16 words from the deadstart panel toggle switches, over channel 0 into locations 0000 through 0014 or 0020 octal of PP 0. Data channel 0 is then disconnected, causing PP 0 location 0015 octal to be cleared. PP 0 reads the contents of location 0000, adds 1, and begins execution at the resulting address (0001). The remaining PPs are still set to the input (71) instruction and may receive data from PP 0 over their assigned channels.

DEADSTART OPTIONS

Typically, the deadstart program will load:

- An 80-column absolute binary deadstart card deck containing one of the several diagnostic programs that can be loaded in this fashion (absolute binary load)

- A logical record from a tape or a disk

Deadstart from Cards

Some diagnostics are available in absolute binary (80-column binary) card decks. These decks load from the card reader into PP 0. These tests can accurately determine the adequacy of the peripheral processor subsystem (PPS) and make a relative assessment of CM and central memory control (CMC).

Deadstart from Tape Library

The MSL without the disk driver resides on magnetic tape with all programs and overlays residing on the tape. CMSE can execute a set of keyboard commands that are read from the card reader.

Hardware Requirements

- One CDC CYBER 170, CYBER 70, or 6000 Peripheral Processor Subsystem

- One CC545 or DD60 Display Station

- One 405 Card Reader

- One magnetic tape subsystem, or 60X or 65X Tape

  Unit with appropriate tape controller

Software Restrictions

MSL without the disk driver cannot generate a dayfile, error logging files, or command buffers, and cannot execute command buffers.

Hardware Requirements

- One CDC CYBER 170,CYBER 70, or 6000 Peripheral Processor Subsystem
- One CC545 or DD60 Display Station
- One 405 Card Reader
- One 844 or 885 Disk Subsystem

Software Restrictions

The controlware must be running in the 844 or 885 Disk Subsystem before the disk can be accessed.

The program TDX (Tape-to-Disk Utility) must have been run previously to place the monitor and the diagnostic library on the 844 or 885 disk. Disk operation is similar to tape operation, except that CED is not executed. (Refer to the TDX description in section 7, Utilities.)

When the PPS deadstarts from the disk, the 844 or 885 controlware reads the contents of the deadstart sector (cylinder 410, track 0, sector 3 on an 844-21; or cylinder 822, track 0, sector 3 on an 844-41) from the disk into PP 0. The deadstart sector contains an 844 or 885 disk deadstart program (CEX). When CEX begins execution, it moves itself to address 74008 and reads the bootstrap loader (MSB) which reads the disk driver program (CE1) into the PP. MSB then enters CE1 and begins execution of the disk I/O driver. CE1 then presents the initial display which permits the operator to make any system parameter changes.

When deadstarting from a disk which is shared with the operating system, the following display appears on the left screen.

```
                 *A*

        CR - OS LOAD AUTOMATIC

         I - DEADSTART WITH OPERATOR
             INTERVENTION

         U - UTILITIES

         M - OFFLINE MAINTENANCE
```

Figure 2-1. CTI Initial Display

Type M to load the CMSE disk deadstart loader program.

Enter a carriage return to load the operating system.

Type I to present the B display and allow running of the Deadstart Diagnostic Sequencer.

Type U to present the U display and allow running of the various operating system utilities.

NOTE
The M option is not present on MSL A loads.

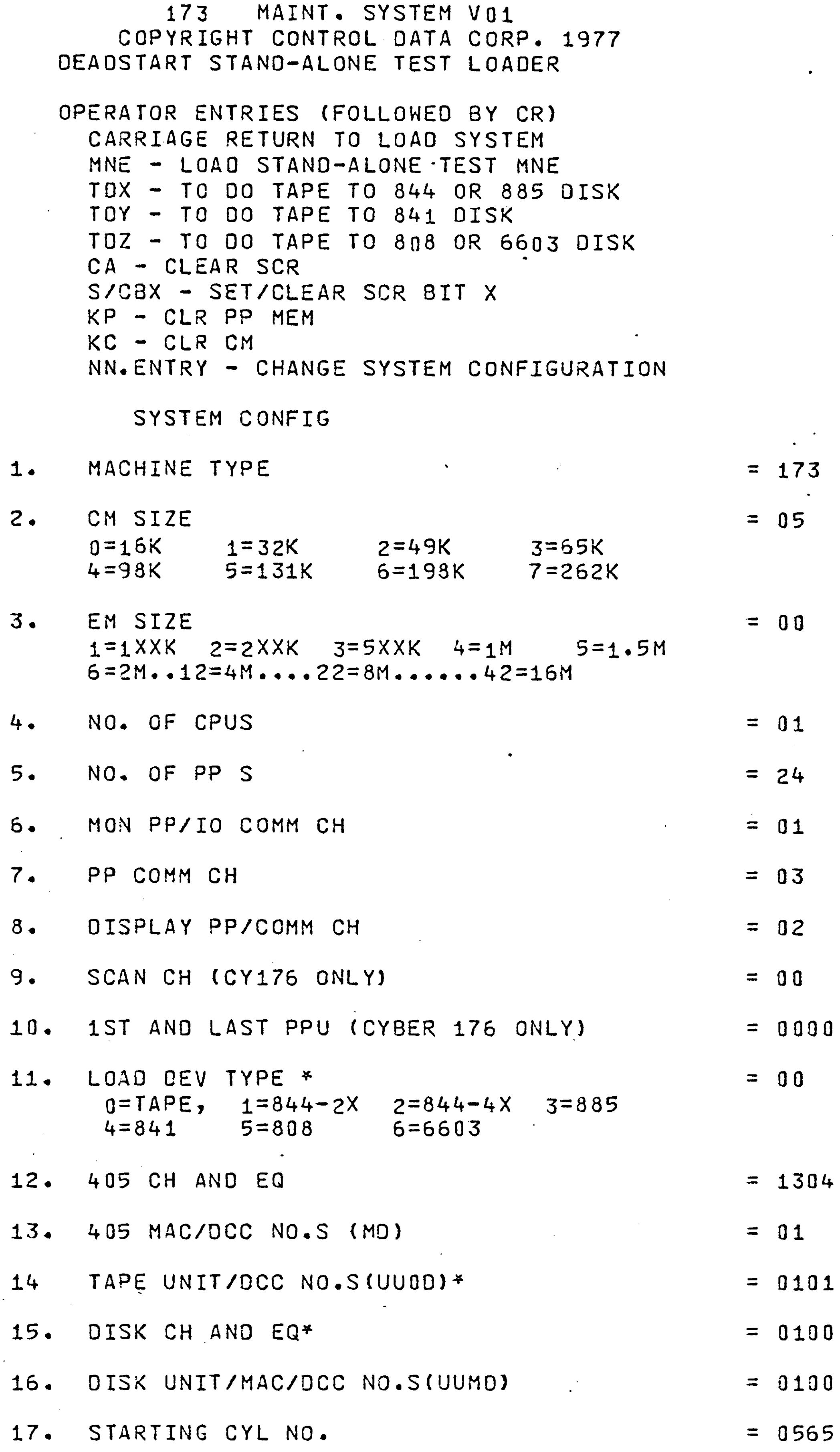
LOADING PROCEDURES

The stand-alone test loader is the means by which either a stand-alone test or the CMSE system may be loaded.

Stand-Alone Test Loader

At deadstart time, the programs CET and CE1 are also referred to as stand-alone test loaders. When a stand-alone test loader begins execution, it presents the left and right displays shown in figures 2-2 and 2-3.

The stand-alone test loader interprets the keyboard commands and performs the requested functions. The keyboard directives can: change a parameter in the system configuration table; clear PP memories or CM; clear the SCR; set or clear a bit in the SCR; load a PP program or utility program into PP 0, PP 1, or CM; or load CMSE.



```
              173   MAINT. SYSTEM V01
           COPYRIGHT CONTROL DATA CORP. 1977
        DEADSTART STAND-ALONE TEST LOADER

        OPERATOR ENTRIES (FOLLOWED BY CR)
          CARRIAGE RETURN TO LOAD SYSTEM
          MNE - LOAD STAND-ALONE TEST MNE
          TDX - TO DO TAPE TO 844 OR 885 DISK
          TDY - TO DO TAPE TO 841 DISK
          TDZ - TO DO TAPE TO 808 OR 6603 DISK
          CA - CLEAR SCR
          S/CBX - SET/CLEAR SCR BIT X
          KP - CLR PP MEM
          KC - CLR CM
          NN.ENTRY - CHANGE SYSTEM CONFIGURATION

             SYSTEM CONFIG

     1.   MACHINE TYPE                                  = 173

     2.   CM SIZE                                       = 05
          0=16K     1=32K     2=49K     3=65K
          4=98K     5=131K    6=198K    7=262K

     3.   EM SIZE                                       = 00
          1=1XXK  2=2XXK  3=5XXK  4=1M    5=1.5M
          6=2M..12=4M....22=8M......42=16M

     4.   NO. OF CPUS                                   = 01

     5.   NO. OF PP S                                   = 24

     6.   MON PP/IO COMM CH                             = 01

     7.   PP COMM CH                                    = 03

     8.   DISPLAY PP/COMM CH                            = 02

     9.   SCAN CH (CY176 ONLY)                          = 00

     10.  1ST AND LAST PPU (CYBER 176 ONLY)             = 0000

     11.  LOAD DEV TYPE *                               = 00
           0=TAPE,  1=844-2X  2=844-4X  3=885
           4=841    5=808     6=6603

     12.  405 CH AND EQ                                 = 1304

     13.  405 MAC/DCC NO.S (MD)                         = 01

     14   TAPE UNIT/DCC NO.S(UUDD)*                     = 0101

     15.  DISK CH AND EQ*                              = 0100

     16.  DISK UNIT/MAC/DCC NO.S(UUMD)                  = 0100

     17.  STARTING CYL NO.                              = 0565
```

Figure 2-2. Typical Left Screen Initial Display

* Entry not displayed if disk deadstart.

```
SCR 0               SCR 1

DATA       LSB      DATA

XXXX       000      XXXX
XXXX       014      XXXX
XXXX       030      XXXX
XXXX       044      XXXX
XXXX       060      XXXX
XXXX       074      XXXX
XXXX       110      XXXX
XXXX       124      XXXX
XXXX       140      XXXX
XXXX       154      XXXX
XXXX       170      XXXX
XXXX       204      XXXX
XXXX       220      XXXX
XXXX       234      XXXX
XXXX       250      XXXX
XXXX       264      XXXX
XXXX       300      XXXX
```

Figure 2-3. Right Screen Initial Display (CYBER 17X only)

The stand-alone test loader initializes the system configuration parameters, shown at ordinals 1, 2, 3, and 4 on the display, from settings on the deadstart panel.

The number of PPs shown at ordinal 5 is determined by the status of associated channels.

The I/O channels at ordinals 6 through 9 cannot be identical to the channel connected to the library device, nor can any channel assigned at ordinals 6 through 9 be identical to a channel assigned to any other of these ordinals. If the library device connects to a channel assigned to ordinals 7 through 9, the stand-alone test loader modifies the channel assignments at the ordinals so they are not identical to the library device channel or identical to a channel assigned to another one of these ordinals.

The stand-alone test loader recognizes the following list of keyboard commands. Unless the command is a single CR, all commands are terminated by a CR.

Command	Description
CR key	Load the maintenance system from the designated library device.
mne	Load the program named mne from the deadstart library medium into PP 0 and initiate execution of the program at address 0001 in the PP.
TDX	Load the tape-to-disk stand-alone routine (TDX) into PP 0 to dump the MSL binary tape to an 844 or 885 disk. All operating instructions for this dump are displayed by TDX. Refer to TDX decription in section 7, Utilities. Use TDY for 841 disks and TDZ for 808 disks.
CA	Perform a Clear All function (6000) on the status and control register.
SBxxx	Set bit number xxx in the SCR. The bit number is assumed to be octal.
CBxxx	Clear bit number xxx in the SCR. The bit number is assumed to be octal.
KP	Clear all PP memories.
KC	Clear central memory.
nn.pp	Change the system configuration parameter (pp) where nn is the ordinal number on the system configuration display.

Test Load Procedure

1. The operator alters the system parameters, if necessary. The type of CPU and sizes of CM and EM are obtained from the deadstart panel. The loader determines the number of PPs from the number of channels.

2. The operator enters the test name followed by a CR.

3. The test loader examines the first 12-bit word of the program (word 0).

4. If word 0 is 0000, the test is loaded into PP 0 and execution begins at address 0001. The contents of PP 0 words 7754 through 7760 contain the information described in the CMSE/PP/PPU Interface section in section 5.

5. If word 0 is 0001, the test loader assumes that a load table exists at words 4 through 20, and it is saved. Words 4 through 10 of the program are replaced with the PP 0 contents of words 7754 through 7760. (Refer to the CMSE/PP/PPU Interface section in section 5. The program is then transferred to PP 1.

6. The loader examines the first four words of the load table for the following format:

Word	Contents	Description
0	na	First two characters of program to be loaded, or 0000, if loading is to stop. If 0000, the loader will start PP 1 at address 0002 and idle PP 0 on channel 0.
1	me	Final characters, left-justified, zero-filled, of program to be loaded.
2	x0yy	If x is 4, then yy is the upper six bits of a CM load address. If x is 0, then program, name, is to be loaded into PP 0, PP 1 started at address 0002, and PP 0 started at address 0001.
3	yyyy	Specifies the lower 12 bits of a CM load address.

7. Loading continues until a four-word entry in the table causes loading to terminate. (Refer to description of table entry words 0 and 2.)

CMSE Load Procedures

Upon typing a carriage return, the following sequence of events is initiated to load CMSE.

1. CE1 or CET will load MMP into the PP specified by ordinal 6.

2. MMP will then request that DSP be loaded from the library device into the PP specified by ordinal 8.

3. MMP will then output an idle package to each of the remaining PPs.

4. At this point the system is up and MMP is communicating to DSP on the channel specified by ordinal 8 and to the idle PPs on the channel specified by ordinal 7.

CE1 or CET always remains in PP 0. After the system is brought up, they are no longer able to load stand-alone tests, but respond to requests from the monitor program.

DEADSTART PANEL SETTINGS

80-Column, Absolute Binary Deadstart Cards

The two deadstart panel settings below are for use with 80-column, absolute binary deadstart cards.

	1	2
Word	Contents	Contents
0001	75cc	75cc
0002	77cc	77cc
0003	e000	e000
0004	77cc	77cc
0005	0001	0001
0006	77cc	77cc
0007	1500	1500
0010	2000	2000
0011	7760	7760
0012	74cc	74cc
0013	71cc	71cc
0014	0000	7773

Code	Description
cc	Card reader channel number (12 or 13)
e	Card reader equipment number (4-7)

Setting Usage

1 Used when there is no header word on the first card of the deck.

2 Used when there is a five-word, 6-PPM header table on the first card of the deck.

Magnetic Tape Deadstart Procedure

The magnetic tape deadstart procedure is used when the library is on a deadstart tape and before it has been transcribed to a disk or other mass storage device.

60X/65X Tape Deadstart Panel Settings

The two deadstart panel settings below are for use with a 60X or 65X non-programmable tape controller.

	1	2
Word	Contents	Contents
0001	75cc	0000
0002	77cc	7700
0003	e0uu	e0uu
0004	77cc	7700
0005	0010	0010
0006	77cc	7700
0007	1400	1400
0010	74cc* #	7400* #
0011	71cc	7100
0012	7301	7301
0013	smry	smry
0014	e0uu	e0uu
0015		

(See parameter settings which follow)

Setting	Usage
1	Used when the tape channel is 12, 13, 32 or 33.
2	Used when the tape channel is zero.

66X/67X Warmstart Settings

The four following deadstart panel settings are for a warmstart of a 66X or 67X programmable tape controller system. In order to accomplish a warmstart, the controlware must be loaded and executing. Loading of the controlware is discussed under Coldstart Procedure.

CYBER 70/6000 Settings

	1	2
Word	Contents	Contents
0001	1402	0000
0002	73cc	75cc
0003	0013	0000
0004	75cc	0000
0005	77cc	77cc
0006	e26u*	e26u*
0007	74cc#	74cc#
0010	71cc	71cc
0011	7301	7301
0012	smry	smry
0013	0000	
0014	7112	
0015		

CYBER 170 Settings

	3	4
Word	Contents	Contents
0001	1402	0000
0002	73cc	75cc
0003	0017	77cc
0004	75cc	2100
0005	77cc	77cc
0006	e26u*	e26u*
0007	74cc#	74cc#
0010	71cc	71cc
0011	7301	7301
0012	smry	smry
0013	yyyy	
0014	yyyy	
0015	0000	
0016	0000	
0017	0000	
0020	7112	

Code	Description
cc	Tape channel number
e	Tape Controller equipment number (4-7)
u	Tape Unit number (0-17)
s	Extended Memory Size: 0=none 2=2xxK 4=1M 6=2M 1=1xxK 3=5xxK 5=1.5M 7=2.5M
m	Central Memory Size: 0=16K 2=49K 4=98K 6=193K 1=32K 3=65K 5=131K 7=262K
r	Type of CPU: CYBER 17X 0=CYBER 171N without CMU 4=CYBER 174 1=CYBER 171 with CMU 5=CYBER 175 2=CYBER 172 6=CYBER 176 3=CYBER 173 7=CYBER 177 CYBER 7X 0=CYBER 711z 4=CYBER 731z 1=CYBER 712z 5=CYBER 732z 2=CYBER 721z 6=CYBER 741z 3=CYBER 722z 7=CYBER 742z 6X00 0=6000 2=6200 4=6400 6=6600 1=6100 3=6300 5=6500 7=6700
y	Used by operating system.

#	Bit 5 set indicates a dual CPU system.
*	Use e12u for 67X units.

Setting	Usage
1	Used when the tape channel is 1-11 or 20-31 and the 66X controller is not connected to the pass-on of a DCC.
2	Used when the 66X tape channel is 0, 12, 13, 32 or 33 and the 66X controller is not connected to the pass-on of a DCC.
3	Used when the 66X tape channel is 1-11 or 20-31 and the 66X controller is not connected to the pass-on of a CYBER 170 Data Channel Converter (DCC), 6681, or 6684.
4	Used when the tape channel is 0, 12, 13, 32 or 33 and the 66X controller is connected to the pass-on of a CYBER 170 Data Channel Converter (DCC), 6681, or 6684.

Disk (844/885) Deadstart Procedure

The program TDX (Tape-to-Disk utility) must have been run previously to place the monitor and the diagnostic library on the 844 or 885 disk. (Refer to the TDX description in section 7.)

The disk operation is similar to the tape operation, except that CED is not executed.

Disk Warmstart Panel Settings

The three following deadstart panel settings are for a warmstart of an 844 or 885 programmable disk controller system. In order to accomplish a warmstart, the controlware must be loaded and executing. Loading of the controlware is discussed under Coldstart procedures following.

NOTE

> Level A09 or higher controlware must be loaded into the 844 buffer controller.

Settings

Word	1 Contents	2 Contents	3 Contents
0001	0000	0000	1402
0002	75cc	0000	73cc
0003	2400	2400	0013
0004	2400*	2400*	75cc*
0005	77cc	7700	77cc
0006	03du	03du	03du
0007	74cc**	7400**	74cc**
0010	71cc	7100	71cc
0011	7301	7301	7301
0012	smry	smry	smry
0013	yyyy	yyyy	0000
0014			7112

Code	Description
cc	Disk Channel number
d	Disk drive.. 844=0. 885=4.
u	Disk Unit number

---

* Bit 5 set indicates NOS/BE,NOS diagnostic sequencer usage for CYBER 176.

** Bit 5 set indicates dual CPU operation.

---

s Extended Memory Size:

0=none	2=2xxK	4=1M	6=2M
1=1xxK	3=5xxK	5=1.5M	7=2.5M

m Central Memory Size:

0=16K	2=49K	4=98K	6=198K
1=32K	3=65K	5=131K	7=262K

r Type of CPU:

CYBER 17X

0=CYBER 171N without CMU	4=CYBER 174
1=CYBER 171 with CMU	5=CYBER 175
2=CYBER 172	6=CYBER 176
3=CYBER 173	7=CYBER 177

CYBER 7X

0=CYBER 711z	4=CYBER 731z
1=CYBER 712z	5=CYBER 732z
2=CYBER 721z	6=CYBER 741z
3=CYBER 722z	7=CYBER 742z

6Xnn

0=6000	2=6200	4=6400	6=6600
1=6100	3=6300	5=6500	7=6700

y Used by operating system.

Setting	Usage
1	Used when the disk channel is 12, 13, 32 or.33.
2	Used when the disk channel is zero.
3	Used when the disk channel is 1-11 or 20-31.

Coldstart Procedure

Coldstart orocedure is used to write controlware to the memory of a programmable controller so that a warmstart may be accomplished.

66X/7021 Coldstart Procedure

The following two deadstart panel settings are used for the coldstart of a 66X Magnetic Tape System (MTS) on various channels. Coldstart accomplishes the load of the controlware into the MTS from the card reader. Once the MTS has been coldstarted, the warmstart procedure should be followed. The warmstart procedure is a deadstart procedure that assumes the controlware has been loaded and is intact. .

An MTS (66X) tape subsystem controlware deck must be available.

The following MTS coldstart deck structure is used.

CEJ	One binary card record (unprefixed)
7/8/9	EOR
CEY	Binary deck (unprefixed)
7/8/9	EOR
MTS	Controlware (unprefixed)
7/8/9	EOR
6/7/8/9	EOF

CEJ routine is read from the card reader by the deadstart panel. It checks the PP 0 SAVE switch on the panel and saves the contents of PP 0 in CM if it is not set. It then reads the first binary card of segment CEY and permits it to execute. CEJ must be exactly one binary card long. The CEJ and CEY cards must be obtained from the operating system for that site.

	1	2
Word	Contents	Contents
0001	75cc	73cc
0002	2400	0013
0003	2400	75cc
0004	77cc	77cc
0005	ertt	ertt
0006	77cc	77cc
0007	14ds	14ds
0010	74cc	74cc
0011	71cc	71cc
0012	7664	7664
0013	00uu	0000
0014		7112*

Code	Description
cc	Card reader channel number
e	Card reader equipment number (4, 5, 6, or 7).
u	Tape Drive unit number (if card reader is on channel 12 or 13)
s	PP 0 Save Switch (1 if PP 0 is not to be saved)
r	CM bypass if 7 octal
d	7-track tape density (0=556,1=800)
tt	Magnetic Tape System channel number (12, 13, 32, or 33)

* If MTS is on channel 12, change word 0014 to 7113.

Setting	Usage
1	Used when the card reader is on channel 12 or 13.
2	Used when the card reader is on channels 1 - 7 or 11.

669-844/7152 Coldstart Procedure

The 669-844/7152 is intended to load controlware into either the tape or disk side of the 7152 Low-Cost, Disk/Tape Controller (LCDT). The controlware must reside on either a 9-track tape or have been pre-recorded on the disk pack by using the LDC utility. (Refer to the LDC description in section 7, Utilities.)

Settings

	1	2	3	4
Word	Contents	Contents	Contents	Contents
0001	75cc@	1402	75cc@	1402
0002	1701	73cc	77cc	73cc
0003	0576	0013	01uu	0007
0004	2400*	75cc	0300	75cc
0005	2400*	1701		77cc
0006	77cc	0576		01uu
0007	007u**	2400*		0000
0010	0300	2400*		0300
0011		77cc		
0012		007u**		
0013		0000		
0014		0300		

Code	Description
cc	Tape or Disk Channel number
uu	Disk Unit number
u	Tape unit number **

@ If cc=0, this instruction must be 7540.

* If a data channel converter, 6681, or 6684 is on the channel, these two instructions must be:

77cc
2100

** Tape unit number must be in the range 10-17 octal. u is an octal digit representing the value of the lower three bits.

Setting	Usage
1	669/7152 Tape Controller Coldstart used when the tape controller channel is 0, 12, 13, 32, or 33.
2	669/7152 Tape Controller Coldstart used when the tape controller channel is 1-7, 11, or 20-31.
3	844/7152 Disk Controller Coldstart used when the disk controller channel is 0, 12, 13, 32, or 33.
4	844/7152 Disk Controller Coldstart used when the disk controller channel is 1-7, 11, or 20-31.

CPU Deadstart Exchange Procedure

This deadstart panel setting performs repeated exchanges of the CPU for maintenance purposes. The CPU program and exchange package must be set up before using this deadstart panel setting.

Setting

Word	Contents	Description
0001	20xx	xxxxxx is the desired exchange.
0002	xxxx	address.
0003	6106	Read input exchange package.
0004	0020	
0005	6304	Write output exchange package.
0006	0020	
0007	1720	Set exchange address.
0010	260y	Exchange CPU y.
0011	0367	Loop.