HONEYWELL

DPS 6 GCOS 6 MOD 400 SYSTEM BUILDING AND ADMINISTRATION

SOFTWARE

DPS 6
GCOS 6 MOD 400 SYSTEM BUILDING AND ADMINISTRATION
ADDENDUM A

SUBJECT

Additions, Deletions, and Changes to the Manual

SPECIAL INSTRUCTIONS

This is the first addendum to CZ02-02, dated March 1986. Insert the attached pages into the manual according to the collating instructions on the back of this sheet. Change bars in the margin indicate new or changed information; asterisks indicate deletions.

Note:

Insert this cover sheet behind the front cover to indicate the updating of the document with Addendum A.

SOFTWARE SUPPORTED

ORDER NUMBER

CZ02-02A

September 1986

45935
0886
Printed in U.S.A.

Honeywell

COLLATING INSTRUCTIONS

To update this manual, remove old pages and insert new pages as follows:

Remove	Insert
v through xvi	v through x
xvii, blank	xi, blank
1-1 through 1-8	1-1 through 1-8
2-3, 2-4	2-3, 2-4
2-13, 2-14	2-13, 2-14
3-1 through 3-20	3-1 through 3-20
5-7 through 5-20	5-7 through 5-14
	5-14.1, blank
	5-15 through 5-20
5-27 through 5-32	5-27 through 5-32
5-67 through 5-70	5-67 through 5-70
6-1 through 6-6	6-1 through 6-6
6-9 through 6-16	6-9 through 6-16
6-19 through 6-24	6-19 through 6-24
6-27 through 6-42	6-27 through 6-42
6-45 through 6-54	6-45 through 6-54
6-57 through 6-64	6-57 through 6-64
6-73, 6-74	6-73, 6-74
6-81, 6-82	6-81, 6-82
6-87, 6-88	6-87, 6-88
6-93, 6-94	6-93, 6-94
E-1, E-2	
E-3, blank	E-1 through E-4
G-1 through G-4	G-1 through G-4
G-5, blank	
J-1 through J-6	J-1 through J-6
	J-7, blank
O-1, O-2	O-1, blank
O-3, blank	
	h-1, blank
i-1 through i-6	i-1 through i-6

USER COMMENTS FORMS are included at the back of this manual. These forms are to be used to record any corrections, changes, or additions that will make this manual more useful.

File No.: 1R13

09/86
CZ02-02A

DPS 6
GCOS 6 MOD 400
SYSTEM BUILDING
AND ADMINISTRATION

SUBJECT

Procedures to Configure a MOD 400 System and to Register Users

SPECIAL INSTRUCTIONS

This manual supersedes the *DPS GCOS 6 MOD 400 System Building and Administration* manual, dated July 1984.

SOFTWARE SUPPORTED

This manual supports Release 4.0 of the MOD 400 Executive.

ORDER NUMBER

CZ02-02

March 1986

Honeywell

PREFACE

This manual is written for the person who is responsible for configuring the GCOS 6 MOD 400 system to meet the characteristics and requirements of an installation site. This person is assumed to be well acquainted with the GCOS 6 MOD 400 Executive and with DPS 6 hardware. In addition, this person is asssumed to have read the Systems Concepts manual and, if the installation supports communications devices, the Communications Processing manual.

This manual describes in detail the procedures necessary to build the system and register users, the directives from which the MOD 400 Executive is built, and the utility programs used in the building process.

Section 1 of this manual introduces the subject of system building and briefly describes the contents of the document.

Section 2 describes the two stages of system startup and the purpose of each stage. Section 2 also describes the operator's and the system's actions during system startup.

Section 3 outlines the information that the system builder must gather prior to configuring the system.

USER COMMENTS FORMS are included at the back of this manual. These forms are to be used to record any corrections, changes, or additions that will make this manual more useful.

File No.: 1S13, 1R13

CZ02-02

Section 4 describes the registration of users with the Edit Profile utility.

Section 5 describes in detail the noncommunications directives that are processed by the Configuration Load Manager as it configures the system to desired specifications.

Section 6 describes communications-related directives processed by the Configuration Load Manager.

Appendix A describes the system halts possible during startup.

Appendix B discusses the configuration of timeslicing.

Appendix C discusses the minimum hardware requirements for MOD 400.

Appendix D discusses configuration considerations relative to the Power Resumption facility.

Appendix E describes the installation and activation of the Listener and login capabilities.

Appendix F discusses the configuration of the error logging capability.

Appendix G discusses the configuration of Display Formatting and Control software.

Appendix H discusses assigning channel numbers for various devices.

Appendix J is a glossary of user registration and administration terms.

Appendix K discusses the configuration of the Disk Cache.

Appendix L describes the configuration of the Decision Data 8045 Card Reader/Punch.

Appendix M describes the configuration of the FACIT 4042 Paper Tape Reader/Punch.

Appendix N describes the configuration of the MEMODYNE M-80 Cassette Tape unit.

Appendix O describes how to transfer system software to a fixed platter and how to boot the system from the platter.

Appendix P describes how to install and operate command accounting.

Appendix Q describes the configuration of the Terminal Presentation Facility (TPF).

Appendix R describes the configuration of the preallocated buffer copy queues.

The following symbols are used in this manual:

Square brackets [] indicate an optional entry.

Braces { } enclose information from which the user must make a choice.

Lowercase letters (e.g., id) indicate a symbolic variable whose exact value must be supplied by the user.

Uppercase letters (e.g., MEMPOOL) indicate commands or directives that must be reproduced exactly as shown.

The character Δ (delta) or the word "blank" indicates that the entry so identified should be blank.

Each Section/Appendix of this document is structured according to the heading hierarchy shown below. Each heading indicates the relative level of the text that follows it.

Level	Heading Format
1 (highest)	ALL CAPITAL LETTERS, UNDERLINED
2	Initial Capital letters, underlined
3	ALL CAPITAL LETTERS, NOT UNDERLINED
4	Initial Capital Letters, not underlined

User Keyins

Indicates user input to the system.

CONTENTS

CONTENTS

CONTENTS

CONTENTS

ILLUSTRATIONS

TABLES

REMOVE THIS PAGE AND PLACE TAB FOR

TAB 1

INTRODUCTION

Section 1
INTRODUCTION

HOW TO USE THIS MANUAL

Section 1 of this document provides a general description of system building and outlines the information a first-time user needs to know before building a specialized system. Section 2 describes the two stages of system startup that must be followed to create an operating environment. Sections 1 and 2 should be read carefully before you begin system building procedures. You should also read Sections 3 and 4 carefully before you configure the system and before you register users.

If your system supports communications devices other than the communications-connected operator console, you should be familiar with the System Programmer's Guide.

The configuration of certain devices is described in appendices of this manual, as is moving the system to the fixed platter.

In general, this manual does not include conceptual information; if necessary, see the System Concepts manual and/or a manual specific to the desired subject area. Commands mentioned in this manual are described in the Commands manual. System macrocalls are described in the System Programmer's Guide. *

SYSTEM BUILDING

System building consists of several discrete operations, each described in different sections of this document. This section places these operations into perspective to enable you to obtain a general view of the system building process, and discusses certain considerations that you should be aware of before you start to build your system.

Your system is delivered with one or more disk volumes containing the software you ordered plus an operational MOD 400 Executive. System building consists of specifying your system variables, identifying your peripheral devices and (optional) communications environment, and tailoring main memory to suit system and user needs. A "system build," starting with the initially supplied system, consists of the operations described below.

MOD 400 PROGRAM MATERIALS AND SOFTWARE INSTALLATION

This subsection describes the organization of the system disk for MOD 400 and the installation of MOD 400. It also describes installing new software components on an already installed MOD 400 system. For a detailed description of MOD 400 software installation, refer to the Software Installation Guide.

MOD 400 Distribution Media

Honeywell Distribution Services delivers software on the media type you request. If the software includes the operating system, the media can be removable disk or tape. If the media is disk, the volume ID corresponds to the six-character system number under which you ordered the software. If the media is tape, the volume ID is SAVTAP. All Executive modules are present on the volume.

Add-on software products are delivered on removable disk, tape, or diskette. The volume ID of the media is ZINSTL. For a detailed description of the software distribution media, refer to the Software Installation Guide.

Disk Directory Structure

Figure 1-1 illustrates a sample disk directory structure for MOD 400. This structure consists of reserved directories representing your master system pack. The figure shows all of the reserved directories that come with every system, plus some that are required by separetely priced products. (Some separetely priced products have reserved directories that are not shown in Figure 1-1.) The following paragraphs describe some of the directories illustrated in Figure 1-1.

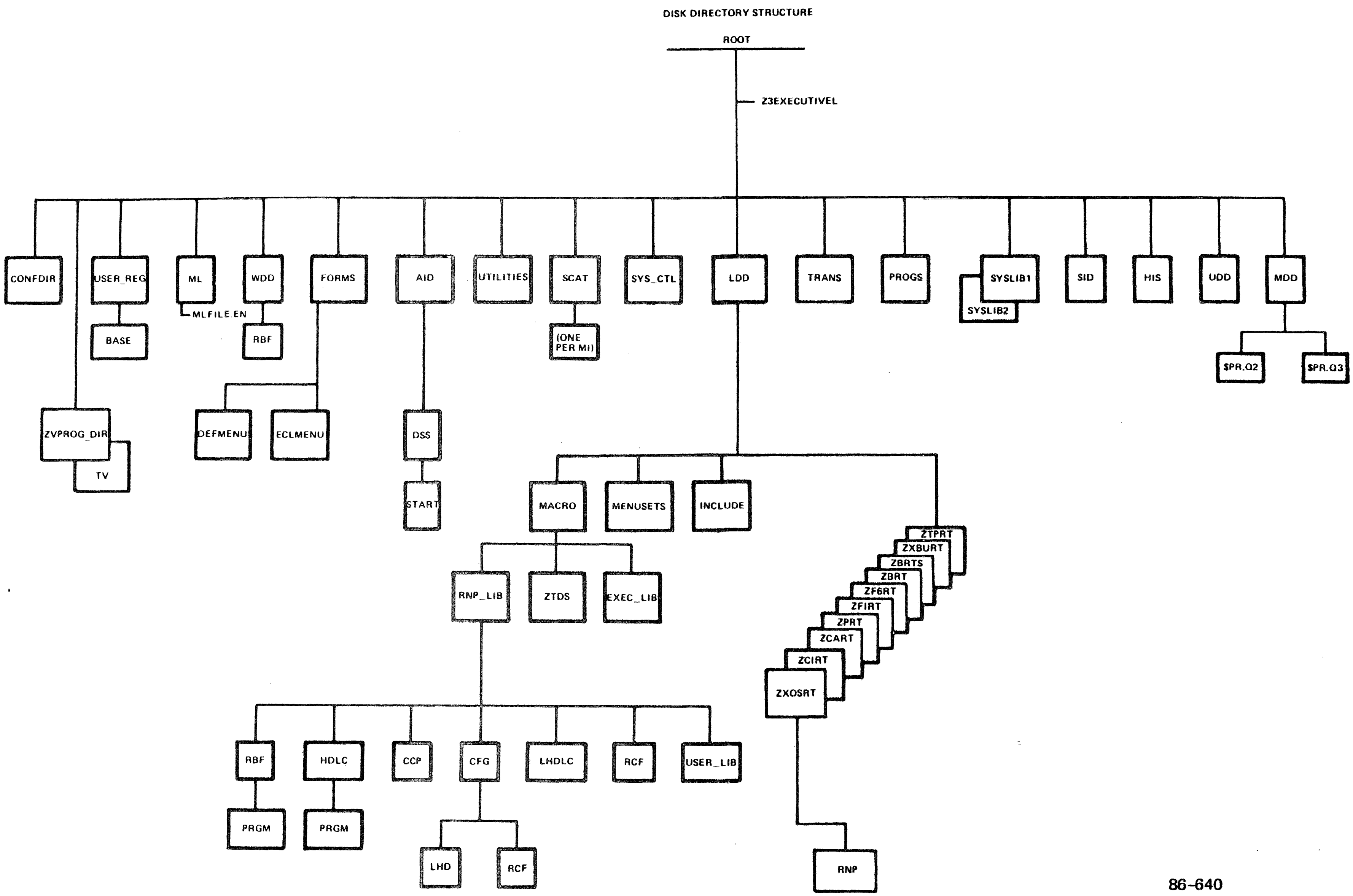


Figure 1-1. Disk Directory Structure

The system initialization directory (SID), contains the software components required to bring the system up. SID contains such components as the communications subsystem and the Configuration Load Manager (CLM) processor.

SYSLIB2, system library 2, contains Honeywell-supplied components such as utilities, the command processor and language compilers.

SYSLIB1, system library 1, can be used for user-defined commands.

TV, the test and verification directory, contains the Honeywell-supplied online test and verification programs.

ZVPROG_DIR contains Honeywell-supplied executable T&V Operating System (TVOS) and offline test and verification programs.

The user directory directory (UDD), contains the directory of user accounts. ACCOUNT is the user account subdirectory, I_D_S is the DM6 I-D-S/II subdirectory, and REPORTS is the reports subdirectory.

The library directory directory (LDD), consists of subdirectories that contain Honeywell-supplied libraries. The libraries are:

- MACRO (macrocall library)
- MENUSETS (menu catalog directory)
- INCLUDE (COBOL INCLUDE directory)
- ZXOSRT (Common run-time library directory)
- ZCART (COBOLA run-time library directory)
- ZCMRT (COBOLM run-time library directory)
- ZBRT (BASIC run-time library directory)
- ZF1RT (FORTRAN run-time library directory)
- ZF6RT (FORTRAN run-time library directory for 6X processors)
- ZXBURT (bound unit run-time directory)
- ZTPRT (transaction processing run-time library directory)
- ZBRTS (alternate BASIC run-time library directory).

The application initialization directory (AID), contains software components used during application initialization. For example, the common central processor intialization directory CCP resides in AID.

MDD is the mailbox directory; all mailboxes reside here.

FORMS, the forms directory, contains Honeywell-supplied forms; user forms, if any, also reside in FORMS. FORMS contains such subdirectories as DEFMENU (the DEF-II menu directory) and ECLMENU (the system command menu directory).

ML, the message directory, contains the Honeywell-supplied message libraries as well as any user additions to the message library.

HIS, the Honeywell directory, is the home directory for the Honeywell-supplied user group.

SYS_CTL, the system control directory, contains the Honeywell-supplied privileged instruction control files PRIV1, PRIV2, and PRV_GROUP.

USER_REG, the user registration directory, contains the Terminals file used by LISTENUR (the bound unit for Listener) and the Profiles file. (See Sections 3 and 4.)

For a detailed description of the contents of these directories, see the Software Release Bulletin.

Creating a User Root Volume

You can separate the directory structure to create a User Root volume and a System Root volume. Once you have separated the volumes, you must inform the system by entering the Change System Directories (CSD) command with the -ROOT argument each time the system is bootstraped. The best way to accomplish this is including the CSD command in the system startup file (>>START_UP.EC). When you separate the volumes, the User Root volume generally contains user-oriented files and directories. The rest of the original directory structure, referred to as the System Root volume, contains Honeywell-supplied files and directories necessary for the operation of the system. This is the volume required to boot the system (sometimes referred to as the bootstrap volume).

To address a file or directory on the User Root volume, you can begin the pathname with the greater-than sign (>) or the volume id (e.g., ^USER>). To address a file or directory on the System Root volume, you can begin the pathname with two greater-than signs (>>) or the volume id (e.g., ^SYS76>).

The following files or directories must always remain in the System Root directory:

- Z3EXECUTIVEL
- SID
- AID
- HIS
- USER_REG

To create a User Root volume, you must move the following files and directories into the User Root directory:

- UDD
- LDD
- MDD
- FORMS
- PROGS
- SYS_CTL
- TRANS.

The user can also move SYSLIB1 or SYSLIB2 or both to the User Root volume. It is recommended, however, that both of these directories remain on the System Root volume.

Software to be Placed on the Bootstrap Volume

Certain modules must be present on the bootstrap volume in order to achieve system startup. Other modules must be present on the bootstrap volume if certain types of processing are to be supported. See the Software Release Bulletin for further information.

Installing Software Packages on an Existing System

To install software on the system disk of an existing system, refer to the Software Installation Guide.

PRELIMINARY CONSIDERATIONS FOR SYSTEM BUILDING

Before you actually begin to build your system, you should first compile the information you need to configure the system to your specifications. The categories of information required are broken into hardware, software, and communications considerations and are discussed in this section. Figure 1-2 is a flowchart of the recommended procedure for gathering information.

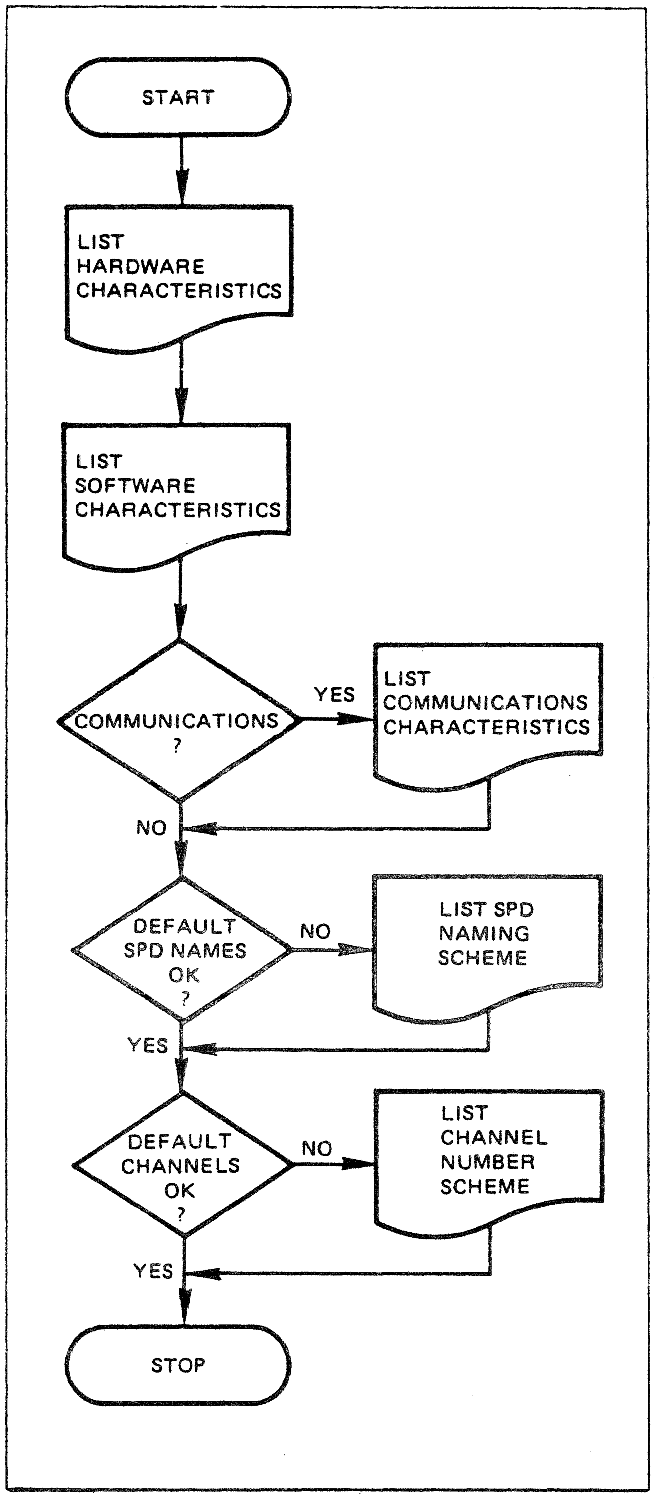


Figure 1-2. Preliminary Considerations Flowchart

Hardware Characteristics

Prior to starting the system-building process, you should gather the necessary information about the hardware devices in your configuration. You can obtain the information about the hardware characteristics by consulting the appropriate hardware reference manuals. The following list provides a representative sampling of some of the hardware characteristics you should be aware of to configure and fine-tune your system. In most cases, you will be able to take the default value for these characteristics during the system-build process.

For the central processor, you should know:

- Whether the central processor is a commercial model or includes a Commercial Instruction Processor (CIP)
- Whether the central processor includes the Scientific Instruction Processor (SIP)
- The size of main memory in words
- The number of communications controllers.

For peripheral devices, you should know:

- The maximum number of devices that you want to configure (regardless of how many are on the Megabus network)
- How many of each type of peripheral device you have in your configuration (you must know the marketing identifier of each peripherial)
- For each terminal: the transmission mode, line speed, desired I/O characteristics, and modem type
- * For disk devices, whether each cartridge module disk has a fixed or removable platter as well as the storage capacity of each device
- For tape drives, the recording density and number of tracks.

Software Characteristics

Once you have compiled a list of hardware characteristics, you should gather information on the software characteristics of your configuration. You must determine what software subsystems your installation will use. Equipment requirements and characteristics of individual software packages are available in the appropriate software reference manuals. The following list provides a representative sampling of some of the software characteristics you should be aware of to configure and fine-tune your system. In most cases, you will be able to take the default value for these characteristics during the system build process.

Specifically, you should know:

- The anticipated maximum number of concurrent users *
- The anticipated number of 512-word system overlay areas *
- Whether or not the system includes the commercial or scientific instruction simulators and whether or not commercial or scientific instructions are used by the software you will execute
- Whether or not the system incorporates Display Formatting and Control software
- An assignment scheme for symbolic peripheral device names, channel numbers, communications priority levels, and logical resource numbers (see the System Concepts manual)
- The number of communications line protocols (the recommended maximum for each MLCP is two; there is no practical maximum for the MLC-16 for Honeywell-supplied line protocols)
- Whether or not the system incorporates the menu subsystem
- Whether or not the system requires national language support *
- The approximate number of Trap Save Areas (TSAs) used by the system
- The approximate number of Indirect Request Blocks (IRBs) used by the system.

Communications Characteristics

If you intend to include communications devices other than the communications-connected operator's console in your configuration, you must read the System Programmer's Guide first, and be thoroughly familiar with the hardware and software characteristics of your optional communications devices.

Specifically, you should know:

- The channel numbers of the communications lines and/or terminals
- The characteristics of the communications lines:
  - Synchronous or asynchronous
  - Line speed
  - Half or full duplex
- The type of modem interface

* Station or endpoint characteristics:

  - Point to point
  - Multipoint

* The terminal characteristics (if applicable):

  - Model number

  - Speed, parity, number of stop bits (asynchronous terminals)

  - Speed and poll address (synchronous terminals)

* The communications link characteristics (if applicable)

  - Endpoint characteristics
  - Poll address(es) (if applicable)
  - Select address(es) (if applicable)

* The hardware priority level allocation

* The number of Channel Control Programs (CCPs) per communication controller. (The recommended maximum is two CCPs per MLCP; there is no recommended maximum per MLC-16.)

This is meant to be only a representative sampling of some of the software and hardware characteristics of which you must be aware. For a full description of each communications protocol, see Section 6.

INITIAL SYSTEM STARTUP

System building is performed in two discrete stages, each stage initiated by an appropriate type of system startup. To build your new MOD 400 system, you must first start up the initially supplied system on your hardware configuration. The system is designed to start up without difficulty on most hardware configurations.

The two stages of system startup and the purpose of each stage are described in Section 2. Each stage of system startup involves:

1. Bootstrapping the system from disk into main memory

2. Executing the Configuration Load Manager (CLM), which reads a file of CLM directives and causes the system to be configured accordingly

3. Creating a system task group.

Additional actions are possible at system startup, depending on whether a special file (named START_UP.EC) exists in the initial working directory of the system task group. If you boot your system from the fixed platter, see Appendix O.

CREATING A CLM FILE

After you collect all the hardware and software characteristics of your system, you must create a file containing a series of system-building (CLM) directives that completely specify the characteristics of the system. These characteristics include the hardware options and physical memory present in the central processor, the complement of peripheral and communications devices present in your configuration, and how memory is allocated among system and application tasks. (The system-building directives are described in detail in Sections 5 and 6).

You create this directive file on disk using the text editor. Once created, the file contains configuration directives that when executed configure a system that corresponds to the actual installation hardware and that supports the software requirements of the installation.

REMOVE THIS PAGE AND PLACE TAB FOR

TAB 2

SYSTEM STARTUP

Section 2
SYSTEM STARTUP

Before you can perform any processing on a DPS 6, you must first perform a system startup, using either the bootstrap volume supplied when your system was delivered (Stage 1 startup) or a bootstrap volume that has been specialized for your installation (Stage 2 startup):

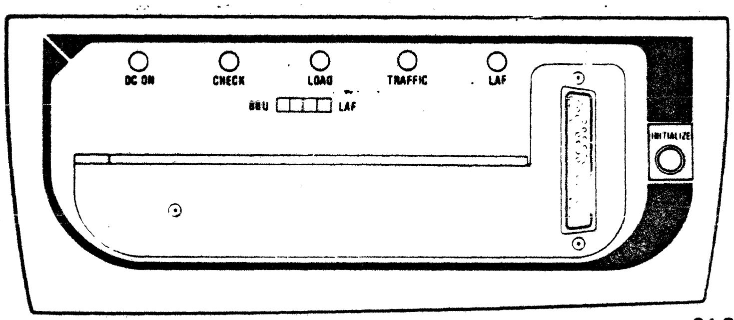
1. Stage 1. System Startup with the Supplied Volume: A Stage 1 startup normally is performed during initial system installation. A Stage 1 startup provides you with the resources necessary to create a CLM_USER file (see Sections 5 and 6) to specialize the system to your installation's requirements. You use the text editor to create the file.

2. Stage 2. System Startup with a Specialized Volume: A Stage 2 startup is used after you have created a file named >>SID>CLM_USER of Configuration Load Manager (CLM) directives (described in Sections 5 and 6) that describes the characteristics of your installation. After you have performed a Stage 2 startup, you have access to all of the resources of your system, unlike the restricted environment created in a Stage 1 startup.

The exact steps that you must follow depend on which stage of startup that you wish to perform and on whether you have a Basic Control Panel, a Full Control Panel or the System Control Facility (SCF) on your DPS 6. If you wish to boot your system from the fixed platter, refer to Appendix O.

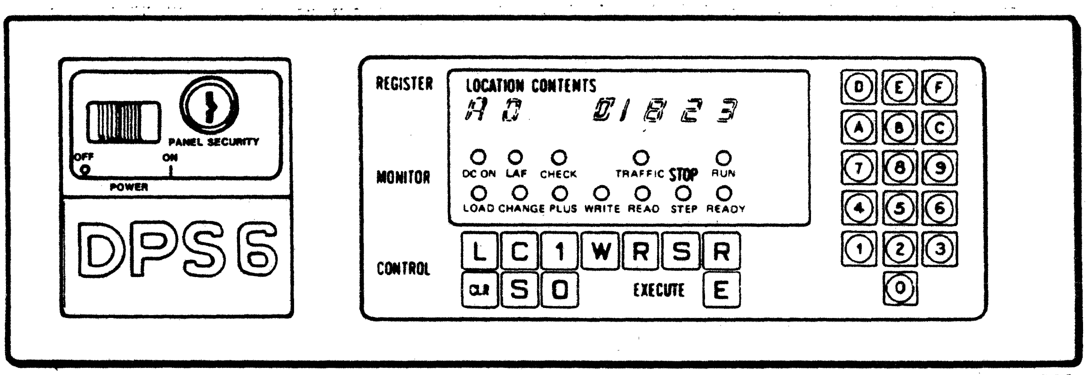
Before you perform either startup, determine which of the following is on your system:

- Basic Control Panel (see Figure 2-1)
- Full Control Panel (see Figure 2-2)
- System Control Facility (SCF).



84-887

Figure 2-1. Basic Control Panel



84-888

Figure 2-2. Full Control Panel

Next decide whether you wish to perform a Stage 1 startup (system bootstrap volume is the one supplied when the system was delivered) or a Stage 2 startup (the system bootstrap volume is specialized to conform to your hardware).

To determine which of the paragraphs you should read for
* procedures specific to starting up your system, see Table 2-1.

Table 2-1. Startup Procedure Information Guide

Type of Volume	Type of Control Panel	Paragraph Reference
Supplied	Basic	"Supplied Volume Startup--Basic Control Panel"
Supplied	Full	"Supplied Volume Startup--Full Control Panel"
Supplied	None (SCF)	"Supplied Volume Startup--System Control Facility (SCF)"
Specialized	Basic	"Specialized Volume Startup--Basic Control Panel"
Specialized	Full	"Specialized Volume Startup--Full Control Panel"
Specialized	None (SCF)	"Specialized Volume Startup--System Control Facility (SCF)"

NOTE

For a detailed description of the features of the SCF, see the TACDIAL Remote User's Guide.

SYSTEM STARTUP WITH THE SUPPLIED VOLUME (STAGE 1 STARTUP)

A startup using the supplied system volume creates a limited processing environment in which the operator terminal is the only terminal at which you can work, and the system volume is the only volume to which you have access. In this environment, the storage capacity of the supplied volume limits the number and sizes of the files you can create. To create a processing environment that reflects the hardware and software of your system, see Sections 3 through 6 of this manual.

The operator terminal is attached to a communications controller. In a Stage 1 startup, the system decides which terminal is the operator's terminal by selecting the communications-connected terminal on channel C000. If no terminal is connected to channel C000, then the communications-connected terminal on channel 0800 is selected. If there is no communications-connected terminal on channel C000 or 0800, then the terminal with the highest communications controller channel number is selected.

A Stage 1 startup uses the CLM directives in the supplied CLM_MCP file. The CLM directives for the communications-connected operator terminal set the operating speed of the terminal to a speed of 9600 bits per second. The operator's terminal configuration switches must be set to match this speed.

In a Stage 1 startup, two task groups, known to the system as $S and $H, are created. The $S (system) task group supports only the execution of operator commands. The operator terminal is the only terminal through which you can communicate with the system task group. The $H (user) task group allows you to use all the system utilities (such as the editors, Linker, and compilers) and the commands not restricted to the system task group; that is, $H is used for application programming.

NOTES

1. For a description of how to set the speed of a terminal, see the appropriate hardware manual for your terminal.

2. If the system supports TACPAC and the operator terminal speed is not 9600 bits per second, the display speed of the OPERATING SYSTEM COMING UP message must be changed; see the <u>TACPAC User's Reference Manual</u> (CJ69).

3. C/R in the procedures is an abbreviation for "Press the RETURN key."

<u>Supplied Volume Startup--Basic Control Panel (Stage 1)</u>

Perform the following steps to start up the system if your DPS 6 has a basic control panel and you wish to use the supplied bootstrap volume (Stage 1 startup):

1. Determine the channel number of the communications-connected operator terminal.

2. Ensure that the operator terminal is powered off.

3. If you have not changed the operator terminal speed in the CLM_MCP file, set the terminal speed to 9600 bits per second. If you have changed the CLM_MCP file, set the speed to the value specified in your CLM_MCP file.

4. Power on the system.

5. Mount the system disk (on the device on channel 0400).

6. Power on the operator terminal.

7. Unlock the control panel and press INITIALIZE on the control panel.

8. If your system does not support TACPAC or you do not intend to use TACPAC, go to step 10. Refer to the TACPAC User's Reference Manual (CJ69) for operating TACPAC.

9. Key in T at the operator terminal within 5 seconds of the OPERATING SYSTEM COMING UP display.

10. When loading is complete, the following messages are displayed at the operator terminal:

    ($S) GCOS6 MOD400-Rrrr-mm/dd/hhmm
    ($H) Group ready!
    ($H) $H

Supplied Volume Startup--Full Control Panel (Stage 1)

Perform the following steps to start up the system if your DPS 6 has a full control panel and you wish to use the supplied bootstrap volume (Stage 1 startup):

1. Determine the channel number of the communications-connected operator terminal. *

2. Ensure that the operator terminal is powered off.

3. If you have not changed the operator terminal speed in the CLM_MCP file, set the terminal speed to 9600 bits per second. If you have changed the CLM_MCP file, set the speed to the value specified in your CLM_MCP file.

4. Power on the system.

5. Mount the system disk on the boot channel, noting the channel number.

6. Power on the operator terminal.

7. Unlock the control panel.

8. Press S (Step).

9. Press CL (Clear).

10. Wait for the Check light to extinguish.

11. Press L (Load).

12. Press E (Execute).

13. Wait for the Check and Traffic lights to extinguish.

14. Press S (Step).

15. Press S (Select).

16. Key in D1.

17. Press C (Change).

18. Key in the boot channel number with bit 14 ON (for example, if boot channel is 0400, key in 0402).

19. Press S (Select).

20. Press R (Ready).

21. Press E (Execute).

22. If your system does not support TACPAC or if you do not intend to use TACPAC, go to step 24. Refer to the TACPAC User's Reference Manual (CJ69) for operating TACPAC.

23. Key in T at the operator terminal within five seconds of the OPERATING SYSTEM COMING UP display.

24. When loading is complete, the following messages are displayed at the operator terminal:

```
($S) GCOS6 MOD400-Rrrr-mm/dd/hhmm
($H) Group ready!
($H) $H
```

Supplied Volume Startup--System Control Facility (Stage 1)

Perform the following steps to start up the system if your DPS 6 has the SCF and you wish to use the supplied bootstrap volume (Stage 1 startup):

1. If you have not yet powered on the system:

   a. If you have not changed the operator terminal speed in the CLM_MCP file, set the terminal speed to 9600 bits per second. If you have changed the CLM_MCP file, set the speed to the value specified in your CLM_MCP file.

   b. Mount the system disk.

   c. Unlock the control panel.

   d. Power on the system.

   e. Go to step 14.

   If you have already powered on the system, proceed to the next step.

2. Mount the system disk.

3. Unlock the control panel.

4. If the SCF is in PANEL mode, go to step 6.

5. Key in ESC ESC # C/R.

6. Press STEP (function key F6).

7. Press DUMP (function key F10).

8. Press CTL/I.

9. Press LOCAL within 1 second of pressing CTL/I.

10. If you have not changed the operator terminal speed in the CLM_MCP file, set the terminal speed to 9600 bits per second. If you have changed the CLM_MCP file, set the speed to the value specified in your CLM_MCP file.

11. Press CTL and CL (Clear) together.

12. Press LOCAL.

13. Press STEP (function key F6).

14. Press CLEAR.

15. Wait for CHECK message to disappear from the screen.

16. Press LOAD (function key F1).

17. Press EXECUTE (function key F12).

18. Wait for CHECK and TRAFFIC messages to disappear from the screen.

19. Press STEP (function key F6).

20. Press SELECT (function key F2).

21. Key in D1.

22. Press CHANGE (function key F3).

23. Key in the bootstrap channel number with bit 14 ON. (For example, if the bootstrap channel number is 0400, key in 0402.)

24. Press SELECT (function key F2).

25. Press RUN (function key F11).

26. Press EXECUTE (function key F12).

27. Key in ESC ESC # C/R (puts SCF in CONSOLE mode).

28. If your system does not support TACPAC or if you do not intend to use TACPAC, go to step 30.

29. Key in T at the operator terminal within 5 seconds of the OPERATING SYSTEM COMING UP display. Refer to the TACPAC User's Reference Manual (CJ69) for operating TACPAC.

30. When loading is complete, the following messages are displayed at the operator terminal:

```
($S) GCOS6 MOD400-Rrrr-mm/dd/hhmm
($H) Group ready!
($H) $H
```

After System Startup With the Supplied Volume

The message that is displayed at the operator terminal when loading is complete:

```
($S) GCOS6 MOD400- Lrrr-mm/dd/hhmm
($H) Group ready!
($H) $H
```

indicates that the two task groups $S and $H are ready for use. The first line indicates the software release number for this system (rrr), and the month, day, and time that the Executive was linked (mm/dd/hhmm). At this point, the $S task group is the operator terminal's default task group; all input is directed to $S. Before you can do any work in the $H task group, you must either (1) change the default task group from $S to $H or (2) address the $H task group by using the $H command format. See the System User's Guide for a description of both of these options, as well as a description of some of the commands and procedures that can be used in the $H task group.

SYSTEM STARTUP WITH AN INSTALLATION-SPECIALIZED VOLUME (STAGE 2 STARTUP)

After you have created a CLM_USER file, you have an installation-specialized volume. This subsection describes the procedures for starting up a system with such a volume.

Specialized Volume Startup--Basic Control Panel (Stage 2)

Perform the following steps to start up the system if your DPS 6 has a basic control panel and you wish to use a specialized bootstrap volume (Stage 2 startup):

1. Determine the channel number of the communications-connected operator terminal as specified in the CLM_USER file. *

2. Set the communications-connected operator terminal speed to the speed specified in the CLM_USER file.

3. Power on the system.

4. Mount the system disk (on the device on channel 0400).

5. Power on the operator terminal.

6. Unlock the control panel and press INITIALIZE on the control panel.

7. If your system does not support TACPAC or if you do not intend to use TACPAC, go to step 10.

8. Key in T at the operator terminal within 5 seconds of the OPERATING SYSTEM COMING UP display. Refer to the <u>TACPAC User's Reference Manual</u> (CJ69) for operating TACPAC.

9. When loading is complete, the following messages are displayed at the operator terminal:

```
($S) GCOS6 MOD400-Rrrr-mm/dd/hhmm
($H) Group ready!
($H) $H
```

Specialized Volume Startup--Full Control Panel (Stage 2)

Perform the following steps to start up the system if your DPS 6 has a full control panel and you wish to use a specialized bootstrap volume (Stage 2 startup):

1. Determine the channel number of the communications-connected operator terminal as specified in the CLM_USER file. *

2. Set the communications-connected operator terminal speed to the speed specified in the CLM_USER file.

3. Power on the system.

4. Mount the system disk on the boot channel, as specified in the CLM_USER file.

5. Unlock the control panel.

6. Press S (Step).

7. Press CL (Clear).

8. Wait for the Check light to extinguish.

9. Press L (Load).

10. Press E (Execute).

11. Wait for the Check and Traffic lights to extinguish.

12. Determine channel number of the boot device as specified in the CLM_USER file.

13. Press S (Step).

14. Press S (Select).

15. Key in D1 (select the D1 register).

16. Press C (Change).

17. Key in the boot channel number (for example, if boot channel is 0400, key in 0400).

18. Press S (Select).

19. Press R (Ready).

20. Press E (Execute).

21. If your system does not support TACPAC or if you do not intend to use TACPAC, go to step 23.

22. Key in T at the operator terminal within 5 seconds of the OPERATING SYSTEM COMING UP display. Refer to the <u>TACPAC User's Reference Manual</u> (CJ69) for operating TACPAC.

23. When loading is complete, the following messages are displayed at the operator terminal:

```
($S) GCOS6 MOD400-Rrrr-mm/dd/hhmm
($H) Group ready!
($H) $H
```

Specialized Volume Startup--System Control Facility (Stage 2)

Perform the following steps to start up the system if your DPS 6 system has the SCF and you wish to use a specialized bootstrap volume (Stage 2 startup):

1. Power on the system.

2. Mount the system disk.

3. Unlock the control panel.

4. If the SCF is in PANEL mode, go to step 6.

5. Key in ESC ESC # C/R.

6. If the terminal speed is set as specified in the CLM_USER file, go to step 15.

7. Press STEP (function key F6).

8. Press DUMP (function key F10).

9. Press CTL/I.

10. Press LOCAL within 1 second of pressing CTL/I.

11. Set the terminal speed to the speed specified for the operator terminal in the CLM_USER file.

12. Press CTL and CLEAR together.

13. Press LOCAL.

14. Press STEP (function key F6).

15. Press CLEAR.

16. Wait for CHECK message to disappear from the screen.

17. Press LOAD (function key F1).

18. Press EXECUTE (function key F12).

19. Wait for CHECK and TRAFFIC messages to disappear from the screen.

20. Press STEP (function key F6).

21. Press SELECT (function key F2).

22. Key in D1.

23. Press CHANGE (function key F3).

24. Key in the bootstrap channel number (for example, if the bootstrap channel number is 0400, key in 0400).

25. Press SELECT (function key F2).

26. Press RUN (function key F11).

27. Press EXECUTE (function key F12).

28. Key in ESC ESC # C/R (puts SCF in CONSOLE mode).

29. If your system does not support TACPAC or if you do not intend to use TACPAC, go to step 31.

30. Key in T at the operator terminal within 5 seconds of the OPERATING SYSTEM COMING UP display. Refer to the TACPAC User's Reference Manual (CJ69) for operating TACPAC.

31. When loading is complete, the following messages are displayed at the operator terminal:

    ($S) GCOS6 MOD400-Rrrr-mm/dd/hhmm
    ($H) Group ready!
    ($H) $H

After System Startup with a Specialized Volume

When the system software is ready for use, a $S ready message is displayed and possibly other ready messages for user-defined user task groups with names most likely different from $H.

After startup, using a supplied system volume, only the operator terminal can be used. An installation can specialize the startup so that user task groups are accessible through the operator terminal, user terminals, or both. If after startup, the message indicates that the $S group is ready but no user task group is ready, determine what user task groups are defined by entering the command line:

    STS -GROUP

If a user task group is accessible through the operator terminal, read "Operator Terminal Procedures After Startup" in the System User's Guide. If a user task group is accessible through a user terminal, go to that terminal and follow the procedures described in Section 3 of the System User's Guide.

IF THE SYSTEM FAILS TO OPERATE

If you have performed a Stage 2 startup procedure as described earlier in this section and the operator terminal fails to come online:

1. Reboot the system using one of the procedures defined under "System Startup with the Supplied Volume," except key a boot channel number of the form

   XXX2

   into register D1 (XXX3 if booting from a fixed platter). XXX is the first three digits of the boot channel number.

2. When the operator terminal is online, change from the $S group to the $H group and view the file >>SID>CLM_USER on the terminal.

   If the operator terminal is a video display terminal, use the following Peruse File (PF) command:

   ```
   PF  >>SID>CLM_USER
   ```

   If the operator terminal is a hard-copy terminal, use the following Print (PR) command:

   ```
   PR  >>SID>CLM_USER
   ```

   At this point, the person responsible for building the system can analyze the CLM_USER file for errors.

3. If changes need to be made to the >>SID>CLM_USER file, you can invoke the text editor, make the changes, and then reboot the system.

   If you cannot find discrepancies in the >>SID>CLM_USER file, contact your service representative.

STARTUP HALTS

If messages do not appear at the operator terminal during a system startup, the system may be in a startup halt. See Appendix A for more information about startup halts.

Startup halts can be classified in three categories:

1. Halts related to the bootstrap operation.

2. Error halts related to the CLM. (CLM is described in Sections 3, 5, and 6 of this manual.)

3. Error halts related to other aspects of system initialization.

A halt related to the bootstrap operation may have been intentionally requested, or it may reflect an error condition. A bootstrap halt is intentionally requested by setting ON bit 13 of the 16-bit (four hexadecimal digits) bootstrap channel number (see Table 2-2). In the event of this type of bootstrap halt, the following register contents are significant:

- D1 register contains bootstrap channel number.
- D2 register contains address mode flag: 1.
- D3 register identifies the bootstrap device type: 0 indicates cartridge module disk, or mass storage unit; 1 indicates diskette.

Table 2-2. Bootstrap Options

Bits in Low-Order Digit of 4-Digit Hexadecimal Bootstrap Channel Number	Meaning if Bit is Set ON
15 (low-order bit)	After the initial bootstrap record is read from the *removable* platter of the cartridge disk whose channel number is contained in bits 0 through 9, the remaining records are read from the *fixed* platter.
* 14	Ignore CLM_USER file (if it exists) under directory >>SID. Instead, use the supplied file of CLM directives CLM_MCP.
13	The central processor halts after the system is bootstrapped and before the system configuration process begins. To continue after the halt, press E (Execute) or function key F12 (if using SCF) on the control panel.
12	Dump memory rather than bootstrap the system.

Error halts during bootstrap result in a 16nn value in the D1 register. See the *System Messages* manual for a description of the 1611, 1612, and 1616 error halts.

NOTE

> If a bootstrap halt occurs with a 1616 error condition, there is a possibility that the D7 register contains no error status. In this case, select D7 prior to retrying the operation and observe D7 during bootstrap processing.

Error halts related to the Configuration Load Manager (CLM) result in a 13nn value in the D1 register; usually additional information relative to the halt is available in or through other registers. Error halts related to other aspects of system startup result in a 99nn value in the D1 register. In some cases, additional information relative to the halt is available in or through other registers. Refer to the System Messages manual.

Bits 13 through 15 can be set ON in any desired combination. However, bit 12 overrides them. Thus, the value of the low-order digit of the 4-digit hexadecimal bootstrap channel number can range from 0 (no options) to 7 (all options), or 8 (dump memory).

REMOVE THIS PAGE AND PLACE TAB FOR

TAB 3

LISTENER

Section 3
LISTENER COMPONENT AND LOGIN

A user can access the system from a terminal in one of two ways, depending on whether or not the terminal is configured for login. Login terminals are reserved for initial user access to the system through a system component called the Listener. This component monitors all terminals that are listed in its Terminals file. Such terminals cannot be directly reserved by system applications. On the other hand, non-login terminals (i.e., those not listed in the Terminals file) can be directly reserved by system applications.

NOTE

> A login terminal can be dynamically "disabled" with respect to Listener--that is, made into a non-login terminal--either by Listener under certain error conditions or by the operator using the SET_LISTEN -OFF command, or as the terminal's initial state using the restriction character D in the T-record (see below). Disabled terminals can later be enabled with SET_LISTEN -ON.

There are two types of login terminals: manual login and direct login. Manual login terminals can use Banner Login, in which the user must enter a Login command line, or Forms Login, in which the user must fill in a login display form. Direct login terminals take their Login commands from the Terminals file as soon as they are physically connected; this process is invisible to the user.

For both types of login terminals, Listener automatically creates a task group for the user at that terminal. In certain cases, the Login command can tell Listener to connect the terminal to a task group that already exists (refer to the description of the destination_id argument of the Login command).

Listener is activated either by commands in the system START_UP.EC or by commands entered from the operator terminal after system startup is complete and the system is operational. The Listener must run as the lead task in the task group $L. Once it is activated, users can log into the system from all login terminals; that is, from all terminals listed in the Terminals file.

The bound unit for Listener is >>SYSLIB2>LISTENUR. LISTENUR supports user registration, as described in Section 4. Only users who have been registered using the Edit Profile utility can log on to the system. The steps for activating Listener are described later in this section.

There are configuration requirements that must be in place before Listener can be activated. You must configure memory pools and user terminals for Listener-spawned groups. If you use the files GROUP$L.EC and TERMINALS described in this section, you must define an operator's CONSOLE and a memory pool AB for the $L group. It is recommended that this pool be a swap pool so that users will not interfere with each other. See the descriptions of the MEMPOOL, SWAPPOOL, and DEVICE directives in Section 5 and "Groups and Memory Pools for Login Tasks" later in this section. For definitions of terms associated with user registration and administration, see Appendix J.

PREPARING TO SUPPORT LISTENER

Your system comes supplied with all of the files needed for support of Listener. To support user registration, the following files are required:

- >>GROUP$L.EC
- >>SYSLIB2>LISTENUR
- >>USER_REG>TERMINALS
- >>USER_REG>PROFILES
- >>SYSLIB2>LOGINH
- >>USER_REG>LOGINHELP.EN

See Figures 3-1, 3-2, and 3-3 for samples of GROUP$L.EC, Terminals, and Profiles files, respectively.

The steps that follow outline the preparation required to support user registration. These procedures are designed to get your installation up and running in user registration mode as easily as possible. For additional security considerations, see Appendix E.

```
& GROUP$L.EC to bring up LISTENUR   85-7-29
&IF [EXISTS ARGUMENT &1] &THEN &G ARG1
&
& No argument: request LISTENUR with default terminals path.
&
CG $L 10 -POOL AB -EFN LISTENUR
EGR $L LISTEN.SYS_ADMIN -OUT !CONSOLE -ARG '>>USER_REG>TERMINALS ' &
 0 'M4-4.0 LISTENUR ' P1 AB
&Q
& -----------------------
& One argument: test to exclude R3.X-style use of ER or UR.
&
&L ARG1
&IF [EQUAL '&1' UR] &THEN $G ER_OR_UR
&IF [EQUAL '&1' ER] &THEN $G ER_OR_UR
&
& Assume arg is intended as Terminals file pathname.
&IF [EXISTS FILE '&1'] &THEN    &ELSE &G NO_TERM_FILE
&
& Arg is good filename:  request LISTENUR with specified terminals path.
CG $L 10 -POOL AB -EFN LISTENUR
EGR $L LISTEN.SYS_ADMIN -OUT !CONSOLE -ARG '&1 ' &
 0 'M4-4.0 LISTENUR ' P1 AB
&Q
& -----------------------
&L ER_OR_UR
&P GROUP$L.EC has been called with the argument &1 .
&P Mod400 R4.0 has only one Listener: LISTENUR (with user registration).
&P GROUP$L.EC accepts one optional argument, the pathname of a terminals file:
&P              the default is >>USER_REG>TERMINALS  .
&P Correct usage is now     EC GROUP$L    or    EC GROUP$L pathname  .
&P You may need to modify your system START_UP.EC file.
&IF [EQUAL '&1' ER] &THEN $G NOGO
&
& Call recursively to filter out the UR argument.
&P GROUP$L.EC continues.
EC GROUP$L &2
&Q
& -----------------------
&L NO_TERM_FILE
&P GROUP$L.EC has been called with a nonexistent TERMINALS file: &1.
&P You may need to modify your system START_UP.EC file.
&L NOGO
&P LISTENUR is not started up.  GROUP$L.EC terminates.
&Q
```

Figure 3-1. Honeywell-Supplied Command File >>GROUP$L.EC

```
G 2 9
A ? L HELP.LOGIN
T CONSOLE
```

Figure 3-2. Honeywell-Supplied File >>USER_REG>TERMINALS *

1. Edit (with the line editor, ED, or the screen editor, SCORPEO) the supplied skeleton file >>USER_REG>TERMINALS to include the names of the terminals on your system that are to be monitored by Listener. See "Tailoring the Terminals File" later in this section for more information. Terminals specified in the Terminals file must, of course, be configured using CLM directives (refer to Sections 5 and 6).

2. Execute the supplied command file >>GROUP$L.EC from the $S group as follows:

   $S EC >>GROUP$L

```
*****************************************************************
 Registration section: ADMIN.SYS_ADMIN.ADM
 Last updated: 1986/07/11 1018
     Section attributes
-----------------------------------------------------------------
 login id                     ADMIN
 default login arguments      -HOME_DIR >>HIS -GROUP AB
 current terminal             CONSOLE
 current language key
 login traits                 SINGLE-USER PROFILE
 password status              PASSWORD REQUIRED
                              IMMUNE TO PASSWORD OBSOLESCENCE
 bad password attempts        0
 session started              1986/07/11 1018
 active users                 0
 stats updated                1986/07/11 1015
 last session begun           1901/01/01 0000
 last session ended           1901/01/01 0000
 last session length          00000:00:00   HHHHH:MM:SS
 total session time           00000:00:00   HHHHH:MM:SS
 number of sessions           0
 last terminal used
 date registered              1986/07/11 1015
 stats reset                  1986/07/11 1015
 max memory allocated         0
 CPU seconds                  0
 comm I-Os                    0
 disk I-Os                    0
 other I-Os                   0
*****************************************************************
 Registration section: USER.USER.INT
 Last updated: 1986/07/11 1021
     Section attributes
-----------------------------------------------------------------
 login id                     USER
 default login arguments      -HOME_DIR >>HIS
 current language key
 login traits                 MULTIPLE-USER PROFILE
 password status              NO PASSWORD REQUIRED
 bad password attempts        0
 session started              1986/07/11 0925
 active users                 0
 stats updated                1984/06/28 0950
 number of sessions           0
 date registered              1982/04/26 1307
 stats reset                  1984/06/28 0950
 max memory allocated         0
 CPU seconds                  0
 comm I-Os                    0
 disk I-Os                    0
 other I-Os                   0
*****************************************************************
```

Figure 3-3. Edited Listing of Honeywell-Supplied Profiles File

```
*******************************************************************
 Registration section: HELP.LOGIN.
 Last updated: 1986/07/11 1022
     Section attributes
-------------------------------------------------------------------
 login id                        ?
 default login arguments         -HOME_DIR >>USER_REG -EFN LOGINH
 current language key
 login traits                    MULTIPLE-USER PROFILE
                                 LOGIN WITH DEFAULT ARGUMENTS
                                 SECONDARY USER
 password status                 NO PASSWORD REQUIRED
 bad password attempts           0
 session started                 1986/07/11 1022
 active users                    0
*******************************************************************
 Registration section: SWIT.SWIT.SW
 Last updated: 1983/12/09 1333
     Section attributes
-------------------------------------------------------------------
 login id        .               SWITSWIT
 default login arguments         SS -MU !LOGIN
 current language key
 login traits                    MULTIPLE-USER PROFILE
                                 LOGIN WITH DEFAULT ARGUMENTS
                                 SECONDARY USER
 password status                 NO PASSWORD REQUIRED
 bad password attempts           0
 session started                 1901/01/01 0000
 active users                    0
*******************************************************************
 Declaration section: MU subsytem
 Last updated: 1982/04/26 1308
     Section attributes
-------------------------------------------------------------------
 privileged bu names             MENU
 indicators                      Section contains statistics.
                                 Subsystem module required.
*******************************************************************
```

Figure 3-3 (cont). Edited Listing of Honeywell-Supplied Profiles File

This command creates the $L task group with LISTENUR as the lead task. The $L task group attributes can be modified by first using the line editor to alter GROUP$L.EC. For more information, refer to "Listener Activation" later in this section.

3. From any terminal named in the Terminals file, log in as the system administrator:

   L ADMIN

   The password is ADMIN. Now use the EP utility to register yourself and other users as discussed in Section 4. Users can now log in.

4. If you wish to run in user registration mode on all subsequent boots, put the command EC GROUP$L in your system's >>START_UP.EC file.

GROUPS AND MEMORY POOLS FOR LOGIN TASKS

When you log in as a primary user at a terminal, a task group is spawned. This spawned task group has the terminal as its user-in and user-out files. It has a two-character task group identification (group id) and is assigned to one of the memory pools in the system CLM file. These pools also have two-character ids, but your pool id does not have to be the same as your group id. The selection of your group id and pool id are described under "Listener Activation" and "Login Command" later in this section.

For a discussion of pool creation and memory allocation, refer to the descriptions of the MEMPOOL and SWAPPOOL directives in Section 5.

TAILORING THE TERMINALS FILE

Listener determines which terminals to monitor for system access from information in a Terminals file. The pathname of the Terminals file can be specified in the command that requests the Listener task, or the pathname can be defaulted to >>USER_REG>TERMINALS. Note that the Profiles file must be in the same directory as the Terminals file. A link to the Profiles file will satisfy this requirement (see the LK command in the Commands manual). The Terminals file consists of variable-length G-, T-, and A-type records. Lines beginning with an asterisk (*) are ignored and can be used to insert comments. Use a text editor to modify the file. Arguments within a record are separated by one or more blank characters. Because of concurrency constraints, you cannot modify a Terminals file that is in use by an activated Listener.

If a terminal is to support direct login or Forms Login, the login line must be specified in the T-record for that terminal.

A direct-login terminal is logged in as soon as it is connected; the user never sees the login process. Direct-login terminals do not display the message of the day unless conditional patch KPF #G04675 is in place. For Forms Login, which is implemented by use of direct login, see "Installing Forms Login", later in this section.

If a terminal is to support abbreviations for Login commands, there must be A-records with the desired command lines (see "A-Records", later in this section).

A terminal can be restricted to "secondary login only" and/or "abbrev only" by including "S" and/or "A" in the Terminals file record. However, such restrictions cannot be applied to terminals running Forms Login (see "T-Records", later in this section). The layout of the records of the Terminals file is illustrated in Figure 3-4.

G-Record

There is only one G-record in any Terminals file, in the format:

```
G base_level max_users
```

base_level

Base level, relative to the lowest-numbered (highest priority) level not used by the system group, at which the lead task of a group spawned by Listener for a terminal is to execute (unless a level is specified in the login line).

max_users

Maximum number of concurrent logged-in users allowed on the system. This value does not include task groups created or spawned by commands other than Login. Login attempts that would exceed this limit are terminated and Listener issues the following message:

3915 Maximum number of users already logged in.

T-Records

There is one T-record in the Terminals file for each terminal on which a user can log in, in the format:

```
T[r...] dev name [login line]
```

Where r... is a series of restriction characters from the set A, S, D, and R (defined below). Some examples of T[r...] are T, TA, TR, TSR, TARSD, and TDS. The restriction characters may appear in any combination and any order; repetitions are ignored.

```
G-Record--only one per file
[A-Records--one or more for all terminals]
T-Record--for a specified terminal; one per terminal
[A-Records--one or more for the above terminal]
T-Record--for another specified terminal
[A-Records--one or more for the above terminal]
As many more T-Records and A-Records as required
*-Records (comment)--anywhere as desired
```

Figure 3-4. Arrangement of Records in a Terminals File

The meanings of the restriction characters are:

A [Banner Login terminals only]

Specifies that only abbreviated logins are allowed at this terminal. If a user specifies a nonabbreviated login line at this terminal, Listener issues the following message:

3916 Login must be by abbreviation

S [Banner Login terminals only]

Specifies that only a secondary login is allowed at this terminal. If a user specifies a primary login line at this terminal, Listener issues the following message:

39EA Primary login not allowed

D

Specifies that the terminal is to be initially disabled. When Listener is activated, the terminal's initial state is the same as if it had been disabled with the command SET_LISTEN device_id OFF. Listener does not monitor the terminal, and it is available for direct use by other programs. The terminal can later be enabled with SET_LISTEN device_id -ON.

R

Specifies that the terminal is connected through a rotary connector. Such connectors are typically used for dial-in connections where there are more terminals than the system can accommodate at one time. The connector is linked to a set of telephone lines and a set of channels on the processor. The user dials in on the line and reaches the connector, which searches its channels in a circular order and connects the line to the first free one it finds. It is this channel that the device_id corresponds to and that Listener knows as a terminal. For purposes of the restriction character R, the defining characteristic of a rotary-connected terminal is that the user cannot know in advance which channel he will be connected to, and if there is a line drop, dialing back in will not necessarily connect him to the same channel he was on before. This restriction character enables Listener to take appropriate action on the terminal in case of a line drop.

dev_name

Symbolic device name of the terminal, as specified at configuration.

login_line

The login command line (including the LOGIN or L characters) that specifies the terminal is used for direct login.

A-Records

An A-record contains an abbreviation character and its associated login line. Any number of A-records may follow the G-record ("global A-records") and/or any T-record ("local A-records"). When a user types an abbreviation, Listener scans the A-records following the T-record for that terminal and, if a match is found, uses that login line for logging in. If the abbreviation is not found, Listener scans the A-records following the G-record for a match, and, if a match is found, uses that login line for logging in. If no match is found, Listener issues the message:

390E Abbreviation for terminal not found

The format of an A-record is:

A abbrev login_line

abbrev

A one-character abbreviation that a user can optionally use when logging in on this terminal.

login_line

The login line associated with the abbreviation. For more information, refer to "Login Command" later in this section and the LOGIN command in the Commands manual.

You, the system administrator, can give an abbreviation character multiple definitions as different login lines, locally at different terminals and globally, for the system as a whole. You can define many abbreviations as logins to the same id (e.g., with different sets of arguments), even at a single terminal. See the examples later in this section.

"Exclusion A-records": If you want to make a login abbreviation "almost global" -- functional at most terminals, but not all -- there is an easier way than by making it local to most of the terminals individually. For example, suppose you want S to mean "L SUPERVISOR" at your 80 data-entry terminals, but not at the five executive terminals. Define a global abbreviation "A S L SUPERVISOR". For the executive terminals, define S locally as a different login line. If no particular local definition is desirable, use "A S BYE", which will cause a logical or physical disconnect and a new login banner, or "A S L ?", which will display the login help file. Either of these will effectively exclude the global abbreviation S from the executive terminals.

A-records can be defined for a direct-login or Forms Login terminal, but the abbreviations cannot be used unless the terminal is converted to manual login, either by the Listener because of an error in the login or by the operator with SET_LISTEN -MANUAL. The terminal then cannot be reset to direct or Forms login until the system is rebooted.

LISTENER ACTIVATION

Listener is activated with the Create Group (CG) and Enter Group Request (EGR) operator commands, or with the Spawn Group (SG) operator command, using the arguments shown below. Once activated, Listener can be terminated only if the system shuts down. Once terminated, the $L task group can be recreated as described below. These commands and their arguments are described in the Commands manual. Special Listener arguments are described below.

```
CG $L base_lvl -EFN  LISTENUR  -POOL id

                                           {'path '}
EGR $L user_id -OUT !CONSOLE -ML path -ARG {"path "}  [x]
    ["message"] [Pn XY [...]]

                              or

SG $L user_id base_lvl -EFN  LISTENUR  -POOL id -OUT !CONSOLE
                  {'path '}
   -ML path -ARG  {"path "} [x] ["message"] [Pn XY [...]]
```

The Honeywell-supplied >>GROUP$L.EC file uses LISTEN.SYS_ADMIN as the user id for Listener's task group, $L. The account id portion must not be changed from SYS_ADMIN. The person id can be changed from LISTEN, but any change must be reflected in the access control lists of the >SYS_CTL>PRIV1, >SYS_CTL>PRIV2, and >SYS_CTL>PRIV_GROUP files, or else Listener will trap as a nonprivileged task attempting to execute a privileged instruction.

-ML path

Specifies the pathname of the message library. The default message library is >>ML>MLFILE.EN.

{'path '}
{"path "}

Pathname of the Terminals file described above. The last character in the pathname must be a blank and the entire pathname must be enclosed in either single or double quotes. An omitted (default) pathname must be written as a pair of single or double quotes with one intervening space (' ' or " "), and results in the default pathname >>USER_REG>TERMINALS. The Terminals file and the Profiles file must reside in the same directory.

[x]

The initial first character in the two-character group id for a login task when a Listener-generated default value is used. This argument may be a digit (0 through 9), an uppercase alphabetic (A through Z), or a lowercase alphabetic (a through z). When this argument is omitted, its default value is 0. The second character, from 0 through 9 or A through Z, is appended when a task group is spawned as a result of the Login command.

When 36 group ids have been used, the first character is incremented; for example, after 0Z the next group id is 10. This cycle continues for ten iterations (360 group ids) or until 9Z, ZZ, or zZ is reached; then the series begins again.

When a group id is specified in a Login command, in a profile, or in a login line in a T-record or A-record, Listener uses that as a group id instead of generating a group id. The Login command is described later in this section and in the Commands manual.

["message"]

The message of the day (for terminals using Banner Login), enclosed in double quotes (" ") if it contains embedded blanks. Listener transmits it to all Banner Login terminals for display when the terminal is ready for someone to log on. The message must be 234 or fewer characters long.

[Pn XY [...]]

Specifies the number and name of up to 30 pools among which Listener is to distribute logged in groups that did not specify a pool id in the Login command lines. Default: If the "P1 AB" arguments are deleted from the command lines in GROUP$L.EC, Listener assigns each group to the memory pool specified in the login line, if any (-POOL xx). Logins that do not specify a pool name are assigned the pool name specified by the [x] argument plus 0. If the "x" argument and the "Pn XY [...]" arguments are all omitted, the pool name is 00.

n must be the number of pool ids that follow. XY [...] is a series of n two-character pool-ids separated by blanks. If the system-supplied >>GROUP$L.EC file is used (see Figure 3-1), Listener assigns all user groups to pool AB. To specify that Listener distribute its spawned groups among a number of pools (up to 30), modify Pn XY [...] (the 4th and 5th arguments) of the >>GROUP$L.EC file (see Figure 3-1); for example, P4 AB AC AD AE. These pools must have been allocated via the MEMPOOL and/or SWAPPOOL directives in the system CLM file.

LOGIN COMMAND

The abbreviated description of the Login (L) command that follows shows the type of entries that can be included in the A- and T-records of the Terminals file and the arguments relating to group and pool id selection. For a detailed description of the Login command, see the Commands manual. Note that the length of the Login command line is limited to 252 characters.

The Login command causes either (1) a task group associated with the user's terminal to be spawned, or (2) the terminal to be attached to an existing task group as a secondary terminal. Once the user has access to the system, the user cannot again invoke Login without first issuing a Bye command unless the task group is otherwise terminated.

FORMAT:

$$L \begin{Bmatrix} \text{user\_id} \\ \text{login\_id} \end{Bmatrix} \begin{Bmatrix} \text{[-CPW]} \\ \text{[destination\_id]} \end{Bmatrix} \text{[ctl\_arg]}$$

ARGUMENTS:

user_id

Identifies the user who is attempting to gain access to the system. Provides the user identification for the spawned task group. The user_id argument consists of two or three fields as follows:

person.account

person.account.mode

person

Name of a person who can access the system; can be from 1 to 12 characters. For example, WDSMITH could be the value for the person field.

account

Name of an account under which the user is to work; can be from 1 to 12 characters. For example, JSINVENTORY could be the value for the account field.

mode

Provides a further identification of the user; can be from 1 to 3 characters. For example, VER could be the value for this field. Mode is optional.

login_id

An alternative identification of the user logging in; can be from 1 to 12 characters.

[destination_id]

Optional argument that permits the user to log in as a secondary user of an existing task group. (A request for a secondary user terminal must have been previously issued by that task group.) To log in as a secondary user of a user-created applications program, type the task group id of the task group in which the application is running.

[ctl_arg]

None or any number of Login control arguments can be selected. The allowable arguments are fully described in the Commands manual under the LOGIN command; they are briefly listed here for reference:

```
-{GROUP | GP} group_id
-POOL pool_id
-{HOME_DIR | HD} directory pathname
-EFN bound_unit_pathname
-HOLD
-LRN n
-LFN n
-IRB n
```

```
-TSA n
-{LEVEL | LEV} n
-{LANGUAGE_KEY | LKEY} xx
-ARG a1 {a2 ...}
```

TERMINAL STATE AFTER LISTENER IS ACTIVATED

When it is first activated and again when the session terminates, Listener performs specific operations affecting the state of a terminal.

If a terminal is not ready when Listener is activated, no initial output messages from Listener are displayed when the terminal comes on line. When Listener is activated and a terminal is connected:

1. If the terminal was configured as a direct login terminal, it does not display the message of the day unless the system administrator has applied the conditional patch from KPF #G04675. A task group is spawned for the terminal if the terminal was configured as a primary login terminal. The lead task defined in the login line is executed. The application should display a prompting message to the terminal indicating that it is ready to accept input. When the lead task terminates, the message of the day is or is not displayed and a task group is immediately spawned again.

   If the terminal was configured as a direct login terminal with secondary login, control of the terminal is given to the task group named in the destination field, provided that the task group has an outstanding request for a secondary terminal. The task group may return the terminal to Listener's control by executing a release terminal monitor call. Listener then attempts a secondary login again.

2. A terminal that is waiting for the user to log in to the system displays either a login banner or a login form. A login banner includes a message of the day (if the Listener was activated with one) and the user-login prompting message identifying the system and giving the date and time, as follows:

   ```
   LOGIN terminal_id yyyy/mm/dd hhmm:ss.t
   ```

   The user can then type in the Login command. The Listener then either connects the terminal to an existing tasl group (secondary login) or creates a task group for the terminal, with whatever lead task was specified in the login arguments. When the lead task eventually terminates or the terminal disconnects from the primary task group, the message of the day is displayed followed by the login prompting message.

The following is a login form:

```
LOGIN                                    Tue Dec 06,1986  15:41:29

           Welcome to your Honeywell System.

   ID:                        _________________________________
   PASSWORD:

   To change your password, type C.
   To enter new login arguments, type A.      _
```

The form for a special login is:

```
    LOGIN2                                 Tue Dec 06,1986 15:41:45

Enter one or neither of the following (but not both):
    Destination group for secondary login:
    Group-id for primary login:
Home directory:      ______________________________________________
Lead task pathname:  ______________________________________________
Hold phone line on logout? (Y/N)  N
Memory pool:    ___
Number of LRNs: ___                      Number of LFNs:   ___
Number of IRBs: ___                      Number of TSAs:   ___
Relative priority level:                 Language key:     ___
Arguments to lead task (start on first line):
___________________________________________________________________
___________________________________________________________________
___________________________________________________________________
___________________________________________________________________
```

When a terminal is released different actions result, depending on how the terminal is connected:

1. A terminal connected to the DPS 6 system via a telephone line and with the hangup option is disconnected. The user must dial in again to use the terminal.

2. A terminal connected via a modem bypass or a telephone line, but without the hangup option, displays the message of the day. Either the login prompting message is displayed or, for a direct login, a login task group is spawned.

3. If two successive logins are rejected because of an error or because the user entered BYE, a terminal connected by a telephone line is disconnected even if the hangup option is not supported.

THE LOGIN PROCEDURE AND PASSWORD PARAMETERS

The following subsections describe the login sequence and the password parameters that can be used to modify the sequence.

Logging In

At a Banner Login terminal, the user types a login line or abbreviation (which Listener invisibly expands to a login line). Listener checks the line's validity. If the user's profile requires a password, Listener requests the password; it is not echoed on the screen, and is covered up on hardcopy terminals. If the password is incorrect, Listener gives the user three more tries. If the user still cannot supply the correct password, or if he or she gives up and enters a null line (hits carriage return, EXECUTE, or TRANSMIT), Listener terminates the login attempt, increments the invalid-password counter in the user's profile, and sends a message to the console.

At a Forms Login terminal, the user fills in the form, including the password if any (it is not echoed). An options field allows the user to choose a new password or to specify further login arguments via a second form; some checking is done on the arguments.

The user may be asked to specify a new password for any of several reasons. He or she may be logging in for the first time, or may have used the -CPW argument (in Banner Login) or "C" in the options field (in Forms Login); or the system administrator may require it ("choose new password" modification in EP; see section 4); or the password may have become obsolete, ninety days having passed since it was chosen or last changed. Listener will ask for the new password, concealing it in the same way as the old one, and will then request it a second time for confirmation. The new password must be at least six characters long. If the password is being changed, the new one must be different from the old one.

Password Parameters

Several conditional patches exist to modify the above sequence. They are referred to here by their entry numbers in the Known Problem File (KPF); full details of customizing and applying each patch are given in its Problem Description text there. Values in brackets are in effect when the system is released, without any conditional patches.

KPF #G03022 controls minimum password length [6 characters], number of password retries allowed at login or line drop [3, for a total of 4 tries], and password obsolescence period [90 days]. Setting the password obsolescence period to zero turns off the obsolescence mechanism, so that passwords never become obsolete.

KPF #G04359 causes a profile to be deactivated when the password retries are exhausted (within a single login attempt). The user's subsequent login attempts are rejected as if the user were not registered. The system administrator can reactivate the profile using the Edit Profile facility, EP. KPF #G04677 exempts multi-user profiles from such deactivation. G04677 has no effect if G04359 is not in place.

KPF # G04573 prevents users from changing their own passwords: it causes the -CPW argument, or the corresponding "C" option in Forms Login, to be rejected with the message "Argument not recognized." It has no effect on the initial choice of password, password obsolescence, or the system administrator's forcing a change of password at an individual user's next login. The administrator can prevent an individual user from using -CPW or the "C" option with the EP login trait "Restricted from changing own password".

CHANGING THE LOGIN MESSAGE OF THE DAY

Use the Set Listener (SET_LISTEN) operator command with the -MSG control argument to change the message of the day. For example:

```
SET_LISTEN -MSG "GOOD MORNING!  SYSTEM SHUTDOWN TODAY AT NOON"
```

See the Commands manual for more information.

INSTALLING FORMS LOGIN

Forms Login is a form-driven login procedure that guides the user through the login process with the form known as LOGIN. If the user wishes to do anything more than the default login, he can pass to a second form, LOGIN2, and perform the operation.

The password required by Forms Login consists of from six to eight characters and must not include any of the following:

- Control characters (e.g., TAB character)
- DEL (hex '7F')
- Commercial at (@)
- Leading or embedded blanks (i.e., a blank with another character following it).

Terminal-Specific Steps

Forms Login is available under user registration only and is invoked on a terminal-by-terminal basis. It can only be used with terminals and Line Protocol Handlers (LPHs) that support forms mode. For example, attempting to use Forms Login on a VIP7200 running under the TTY LPH results in an error. A terminal that is using Forms Login does not accept logins by abbreviation and cannot be restricted to secondary logins.

To install Forms Login on a terminal, ensure that the VDAM directive (refer to Section 6) has been specified in the CLM_USER file and that the maximum-terminals parameter is at least the same as the number of terminals that are to use Forms Login. Next, add the Login command line L SWITSWIT to the T-record. Thus, if the T-record of the terminal on which you wish to install Forms Login reads:

```
T ATD01
```

you should modify it so that it reads as follows:

```
T ATD01 L SWITSWIT
```

Ordinarily, T-records can start with T or T plus any combination of the restriction characters (A, S, R, or D). But because Forms Login is not compatible with terminals restricted either to secondary login (S) or login by abbreviation only (A), you have to remove those restriction characters from the T-record of any terminal on which you want to install Forms Login.

Since LISTENUR reserves the Terminals file with shared read access, you cannot directly modify the file while LISTENUR is running, although you can read it. There are two ways to modify the Terminals file:

1. Edit it after booting the system and before bringing up LISTENUR.

2. Rename the file while LISTENUR is running, using the -FORCE option. Read the renamed file into the line editor, and write the edited file to the original name. The new Terminals file is then used the next time LISTENUR is brought up. The following sequence of commands illustrates this second method:

```
CWD >>USER_REG
RN TERMINALS OLDTERM -FORCE
ED
R  OLDTERM
    .
    .   editing commands
    .
W TERMINALS
Q
```

Systemwide Steps

In addition to changes to the Terminals file, Forms Login requires a registered user called SWITSWIT and the operation of the Subsystem Switcher. SWITSWIT is already registered in the system as released.

Include in your system's START_UP.EC file the line:

```
EC >>SPAWN_SS
```

It will activate the Subsystem Switcher, which must be operating in order for Forms Login to function. If you bring up LISTENUR with Forms Login configured before starting up the Subsystem Switcher, the terminals configured for Forms Login repeatedly display "Secondary login not accepted." until the Subsystem Switcher is operating. Then they display the login form, indicating that the terminal is ready to accept logins.

The supplied >>SPAWN_SS.EC file for starting Subsystem Switcher specifies "-LFN 40", which is sufficient for ten Forms Login users. If you expect more than ten simultaneous users of Subsystem Switcher (including Forms Login terminals actually displaying the login form), the SPAWN_SS.EC file should be changed to specify four LFNs per user.

Because large amounts of memory with low activity are required to display the login forms, it is particularly appropriate for the Subsystem Switcher to operate in a swap pool.

You can modify the form displays by using VISION, although you should take care not to change any of the fields. Both LOGIN and LOGIN2 are in the file >FORMS>ECLMENU>ECL_MENU.EN.

EXAMPLES OF LISTENER OPERATION

For these examples, the CLM file includes the following directives:

```
DEVICE ATD01,11,21,X'0580',ATD01
DEVICE ATD02,12,22,X'0600',ATD02
MEMPOOL S,,50000
SWAPPOOL F,AB,60000
MEMPOOL I,A1,40000
```

Example 1:

In this example, assume that >>USER_REG>TERMINALS has been edited to contain the following records:

```
G 1 3
T ATD01
A X L X.X.X
A Y L Y.Y.Y -GP YY -POOL A1 -HD >>SYSLIB2
*BILL'S TERMINAL
T ATD02
A W L W.W.W
```

Assume further that the default login arguments in the user profile for W.W.W do not include a -HD or -HOME_DIR argument to set the home directory.

The following commands initiate Listener and illustrate user logins. Listener is initiated by command 1 typed at the operator terminal in the system group. Command 2 illustrates a login from terminal ATD01. Commands 3 and 4 illustrate logins from terminal ATD02. Note that command 3 does not result in a login.

Command 1 (issued from the operator's console):

```
EC GROUP$L
```

Command 2 (issued from terminal ATD01):

```
Y
```

Command 3 (issued from terminal ATD02):

```
X
```

Command 4 (issued from terminal ATD02):

```
W
```

Command 1 uses the Honeywell-supplied command file GROUP$L.EC (described in this section) to create task group $L with LISTENUR as the lead task. The task group executes at relative level 10 in memory pool AB. The command file activates LISTENUR with a user id of LISTEN.SYS ADMIN, with the console as the in-path and out-path, and with no working directory. The command file passes the following items to LISTENUR: >>USER_REG>TERMINALS as the pathname of the Terminals file, 0 as the initial first character of the generated group ids, "M4-4.0 LISTENUR" as the initial message of the day, and one user distribution memory pool, AB.

When command 2 is executed, Listener spawns the task group YY using memory pool A1, with the command processor as lead task and with >>SYSLIB2 as the working directory. The user at terminal ATD01 can now perform any desired function.

When command 3 is executed, Listener issues the message:

Error in abbrev. Use another or a full login

The message is issued because the A-record for the abbreviation X is under the T-record for ATD01 in the Terminals file; hence, the abbreviation is invisible from ATD02. Listener also writes the following messages on the operator's console:

```
($L) LISTEN:  39 390E (3A)
($L) Abbreviation for terminal not found
($L) ATD02
```

These messages notify the operator of a problem encountered at terminal ATD02; this problem may require a change to the Terminals file.

Command 3 did not result in a login. Therefore, the message of the day and the login prompt message are redisplayed on terminal ATD02. When the user types command 4, Listener spawns task group 00 with a user id of W.W.W and with the command processor as lead task. The command processor then issues message "17 0222 THE WORKING DIRECTORY OR HOME DIRECTORY MUST BE DEFINED" because neither the login line nor the profile's string of default arguments include the -HD control argument. The user must issue a >>SYSL1B2>CWD command to establish a working directory before initiating any application.

Example 2:

This example also illustrates using the Honeywell-supplied command file >>GROUP$L.EC to initiate LISTENUR.

Assume that the following steps have been performed:

- The system administrator copied (for security reasons) the user registration file (>>USER_REG>PROFILES) to a separate volume named USERS.
- The system administrator registered users for the system, storing their profiles in ^USERS>PROFILES. The Profiles file still contains the original registration with login id USER (a multiuser profile).
- The system administrator has created a Terminals file identical to the one in Example 1 (above) with pathname ^USERS>TERMINALS (recall that the Terminals file and the Profiles file must reside in the same directory).
- The following command was issued from the operator's console:

  EC GROUP$L ^USERS>TERMINALS

After the above steps have been performed, you can login from either terminal by typing:

```
L USER
```

Listener does not request a password; it immediately spawns a user task group with group_id 00. If a second user types

```
L USER
```

while the first is still active, Listener spawns a user task group 01 for the second user and records in the Profiles file that two users are logged in using the registration USER.USER.INT.

REMOVE THIS PAGE AND PLACE TAB FOR

TAB 4

USER REGISTRATION

Section 4
USER REGISTRATION

User registration lets a system administrator control user access to a MOD 400 Executive. Appendix J contains definitions of terms associated with user registration and administration. The Edit Profile (EP) administrative utility permits the registration of users and the specification of each user's privileges and constraints.

Once Listener is activated, all attempts to access the system are screened using the Profiles file. The Profiles file contains a user profile for each registered user of the system. A user profile consists of one or more sections. Every profile contains a registration (RE) section and can contain other sections.

The RE section of a user's profile specifies the user's login attributes and can contain statistics describing the user's system usage. A comments (!C) section is an optional section providing a means for the system administrator to associate ASCII text with the user profile (e.g., billing information or messages). Other optional sections, such as the menu processor (MU) section, are associated with a particular subsystem and serve to enhance the user's capabilities when running under that subsystem.

The EP administrative utility initiates an interactive dialog that lets you select any of the EP's functions to create and maintain user profiles.

The List Profile (LP) utility permits an administrator to list the contents of one or more user profiles.

PRELIMINARY SYSTEM ADMINISTRATIVE PROCEDURES

Before you begin using the EP utility, you must tailor your Terminals file to your installation's requirements, and you must
* have Listener executing in the task group $L. For a description of the required steps, refer to "Preparing to Support Listener" in Section 3 of this manual.

EDIT PROFILE (EP) UTILITY

Edit Profile (EP) is the interactive program that lets you create and manage (modify, delete, etc.) user profiles in the Profiles file.

Once invoked, EP prompts you with an easy-to-use dialog, permitting you to perform the 10 functions described in "Edit Profile Functions."

Only those users with the account identifier .SYS_ADMIN can use EP.

The following paragraphs describe the operating features of EP.

Requesting Help

If you don't understand any prompt or question issued by EP, type a question mark character (?) followed by a carriage return (RETURN or RET). EP responds to the question mark with a message describing what is expected in response to the question or prompt.

Exiting EP Functions

You can exit any EP function at any time by pressing the BREAK (or BRK) key and then typing a Program Interrupt (PI) command. EP returns you to the "Enter function:" prompt. The contents of the Profiles file is not changed unless EP issued a message that a change had been made before you pressed the BREAK key.

Answering Yes/No Questions

You can answer any question that requires a yes/no answer by typing Y, YES, N, or NO. There is no default answer for yes/no questions.

Returning to Previous Prompt

You can return to the previous question or prompt by typing a less-than character (<) in response to any question or prompt. Typing two less-than characters (<<) returns you to the first prompt issued by the chosen function.

Accepting the Default Value

Not all questions or prompts have default values. If there is a default value offered, press the RETURN key to accept the default. EP then issues the message "Default accepted."

Edit Profile Functions

EP asks which function you wish to perform by prompting:

```
Enter function:
```

EP functions are:

- REG to register new users (that is, to create a new user profile)
- MOD to modify a section of an existing user profile
- ADD to add a new section to an existing user profile
- DEL to delete a section or sections of a user profile or to delete an entire user profile
- DUP to duplicate (copy) an existing section in one user's profile to another user's profile
- LP to list the information contained in one or more sections of a user's profile or set of user profiles
- STATS to display statistics from a section and optionally reset them to 0
- SEC to list the section ids of a user's profile
- DEC to declare that a subsystem can access the Profiles file and to delete such a declaration
- Q to terminate EP.

SPECIFYING USER IDS

EP functions, once they are invoked, ask for the user id of the user profile in question. Specify user id in the form:

```
PERSON.ACCOUNT.MODE
```

On subsequent functions, you can accept the default user id by pressing the RETURN key. The default user id is the last explicitly entered user id.

SPECIFYING SECTION IDS

Some EP functions also request a section id (when the specified user profile contains more than one section). No default value exists for section id. If EP requests a section id, you must always enter the two-character section id explicitly (for example, RE, !C, MU, etc.).

Registering the Administrator

The Profiles file supplied with your system contains a user profile for ADMIN.SYS_ADMIN.ADM. The password for this user is ADMIN. Log into the system using this profile and invoke EP to register yourself as the system administrator with the user id name.SYS_ADMIN (see "Registering Users (REG)" in this section).

After you register yourself, log out, log in using the profile that you just created, and delete the profile for ADMIN.SYS_ADMIN.ADM using the DEL function of EP. (See "Deleting Sections (DEL)" later in this section).

Only users registered with the user id name.SYS_ADMIN (i.e., administrators) are granted access to the EP utility.

Registering Users (REG)

To register a new user, invoke EP. The system responds with the following registration dialog:

Enter function:

Type REG to register a new user.

User_id:

Type the user id of the registrant in the form PERSON.ACCOUNT[.MODE]. The user id identifies this user's profile for as long as he or she is registered in the Profiles file. The user id must conform to file system standards; that is, each component of the user id must be a valid pathname. (The first character of any name must not be hexadecimal FF (lowercase y with diaeresis) or hexadecimal 2E (period). The underscore must be used to join two words.) The components of a user id are:

PERSON

1- to 12-character name of the individual who can access the system.

ACCOUNT

1- to 12-character name of the account or project under which the user works.

MODE

1- to 3-characters that further identify this user; MODE is optional.

Enter (user_id)'s login_id:

Type the login id for the user. This is an alternate id that the user can specify to log into the system. If the login id is not specified, the user must log in with the above user id, or with a login abbreviation specified in the Terminals file. The login id can be up to 12 unique characters. The period (.) is not a permissible character in a login id. Also, a login id that can be used in a Forms Login must not contain the commercial AT sign (@).

The default is no login id.

Enter (user_id)'s default login arguments:

Type the user's default login line. Choose from the control arguments listed in the login command (see the <u>Commands</u> manual). Enter 80 or fewer characters. The default is to specify <u>no</u> default login line.

EP now displays a list of optional login traits that you can specify for the user.

<u>Optional login traits</u>

(1) Single-user profile
(2) Multiuser profile
(3) Subsystem-Switcher profile
(4) Secondary-user profile
(5) Login with default login line only
(6) Login with abbrev only
(7) Restricted from changing password
(8) Immune to password obsolescence

Enter all desired login trait numbers:

Type the number or numbers corresponding to the traits for this user. Separate numbers with a space or comma; for example, to specify single-user profile (1) and Subsystem-Switcher profile (3) traits, type either

1 3 or 1,3

Any combination of numbers is accepted; EP does not report illogical combinations. All input must be entered on one line. There is no default for this question.

Password required to login?

Answer YES, Y, NO, or N to this question. If you answer Y or YES, the user is required to type a password when logging in. If you answer N or NO, no password is required.

Keep statistics on (user_id)'s use of the system?

Answer YES, Y, NO, N. Answer Y or YES if you wish to collect the following set of system usage statistics for this user:

- Date registered
- Date stats reset
- Last terminal used
- Last session length
- Total session length
- Number of sessions
- Maximum memory allocated
- CPU time
- Communications I-O count
- Disk I-O count
- All other I-O count.

These items are defined in "Optional RE Section Statistics" later in this section.

EP announces that it has created a new user-profile with the following message:

```
(user-id) now registered
```

In the following example, JONES.USER is registered:

```
EP
Functions:  REG (register), MOD (modify), DEL (delete), DUP
            (duplicate), STATS (list statistics), ADD (add a
            section), SEC (list section ids), Q (quit)

Enter function:
REG
Enter the user_id of the new registrant.
JONES.USER
Enter JONES.USER's login id:
JONES
Enter JONES.USER's default login arguments:
-HD >UDD>JONES -LK EN
```

Optional login-traits

(1) Single-user profile
(2) Multiuser profile
(3) Subsystem-Switcher profile
(4) Secondary-user profile
(5) Login with default login line only
(6) Login with abbrev only
(7) Restricted from changing password
(8) Immune to password obsolescence
(9) NONE OF THE ABOVE

Enter all desired login trait numbers:
2,4
Password required to login?
YES
Keep statistics on JONES.USER's system usage?
NO
JONES.USER now registered.

Enter function:

Modifying a User's Profile (MOD)

The Modify function (MOD) permits you to change a section of a user's profile. The system dialog is as follows:

Enter function:

Type MOD to modify a user's profile.

User_id:

Type the user id of the user whose profile you wish to modify or accept the default user id. The default user id is the last explicitly typed user id.

Section_id:

Type the two-character id of the section that you wish to modify. If (user_id)'s profile contains only the RE section, this prompt does not appear.

EP now displays a numbered list of those attributes of the section that can be modified. For example, for the MU (Menu) section, EP displays:

MU Section Menu

(1) Access_roles
(2) Menu_path
(3) First_menu_name
(4) NONE OF THE ABOVE

EP next displays the following prompt:

Selection:

Type the number of the attribute that you wish to modify. You can only modify one attribute at a time. To exit the MOD function, type either the number for the NONE OF THE ABOVE option (4 for the Menu section) or < to back up.

Once you have specified an attribute number, EP asks a number of questions about that attribute. The nature and length of the dialog depend on the attribute being changed. In some cases, EP issues just a prompt requesting the new input; in others, EP asks a series of questions or even prints out another list of options. In all cases, you can use the less-than character (<) to back up to the previous level.

After you have answered all attribute-related questions, EP redisplays the original attribute menu. If you wish to change other attributes in this section, you can do it now. The menu now includes two action keys: A and N; for the MU (Menu) section, the new list is:

MU Section Menu

(1) Access_roles
(2) Menu_path
(3) First_menu_name
(4) NONE OF THE ABOVE
(A) Accept the changes and exit the MOD function
(N) Negate the changes

EP now requests a selection:

Selection:

Type the number (or letter) of the attribute that you wish to modify, or type A or N. If you type A, EP writes the changed section to the user's profile and exits the MOD function. If you type N, EP restores the section to its original condition and redisplays the attribute menu so that you can start again from the beginning.

The list of attributes for the RE section is as follows:

RE Section Menu

(1) Login id
(2) Login default arguments
(3) Login traits
(4) Password status
(5) Statistics maintenance
(6) NONE OF THE ABOVE

If you select (1), the previous contents of the login id field are displayed and you can specify a new login id for the user.

If you select (2), the previous contents of the default login argument's field are displayed and you can specify a new set of default login arguments for the user.

If you select (3), you can choose from the following list of login traits:

- Single-user profile
- Multiuser profile
- Subsystem-Switcher profile
- Secondary-user profile
- Login with default login line only
- Login with abbrev only
- Restricted from changing password
- Immune to password obsolescence

If you select (4), you can now require a password, make the password no longer required, or allow the user to choose a new password.

If you select (5), you can specify starting or stopping the collection of system usage statistics for the user.

The following example modifies the RE section of GRIFFITHS.EXSER to allow him to choose a new password and to no longer collect usage statistics.

```
Enter function:
MOD
User_id:
GRIFFITHS.EXSER
Section_id:
RE

        RE Section Menu
--------------------------------------------------------------
   (1)  login id
   (2)  default login arguments
   (3)  login traits
   (4)  password
   (5)  statistics maintenance
   (6)  NONE OF THE ABOVE

Selection:
4

        Password options
--------------------------------------------------------------
   (1)  Allow user to choose a new password at next login.
   (2)  No longer require a password at login.
```

```
   (3)  No change to current password status.

Select desired option number:
1

        RE Section Menu
-------------------------------------------------------------
   (1)  login id
   (2)  default login arguments
   (3)  login traits
   (4)  password
   (5)  statistics maintenance
   (6)  NONE OF THE ABOVE
   (A)  Accept the changes and exit.
   (N)  Negate the changes.

Selection:
5
Reset current statistics to zero?
NO
Discontinue statistics maintenance on this profile?
YES

        RE Section Menu
-------------------------------------------------------------
   (1)  login id
   (2)  default login arguments
   (3)  login traits
   (4)  password
   (5)  statistics maintenance
   (6)  NONE OF THE ABOVE
   (A)  Accept the changes and exit.
   (N)  Negate the changes.

Selection:
A
RE section modified for GRIFFITHS.EXSER

Enter function:
```

Adding a New Section to a User's Profile (ADD)

The ADD function adds a new section to an existing user profile.

Enter function:

> Type ADD to add a new section to a previously registered user's profile.

User_id:

> Type the user id of the profile to which you wish to add a new section, or accept the default user id.

Section_id:

Type the two-character id of the section that you wish to add.

Once you have specified a section id, EP asks a series of questions about that section. The nature and length of the dialog depend on the section being added. You can use the less-than character (<) to back up to the previous question, if necessary.

After you have answered all of the required questions, EP displays the following message:

```
xx Section added to (user_id)'s profile
```

where xx is the section id of the added section.

Before a subsystem section can be added to any user profile, the subsystem must have been previously declared using the DEC function of EP.

The following example adds MU and !C sections to GRIFFITHS.EXSER's profile.

```
Enter function:
ADD
User_id:
GRIFFITHS.EXSER
Section_id:
MU
Enter pathname of menu catalog (omit language suffix):
>UDD>GRIFFITHS>MENUS
Enter first menu name (maximum of 12 characters):
UTILITIES
Enter access roles (maximum of 10):
C/R
MU section added to GRIFFITHS.EXSER

Enter function:
ADD
User_id:
C/R
Default accepted.
 GRIFFITHS.EXSER
Section_id:
!C

Enter comment line 1:
Your total CPU usage for this billing period was 200 seconds
```

Enter comment line 2:
Your maximum memory usage was six units

Enter comment line 3:
Total cost $28.00

!C section added to GRIFFITHS.EXSER

Enter function:
Q
RDY:

Deleting Sections (DEL)

The DEL function deletes either a specified section from a user's profile or all of the sections of a user's profile. The system dialog is as follows:

Enter function:

Type DEL to delete one or more sections from a user's profile.

User_id:

Type the user id of the user's profile that is to have a section or sections deleted in the format NAME.ACCOUNT[.MODE], or accept the default user id.

NOTE

When deleting user profile records, you can use the star (*) option for either the person or account component of the user id. Profile records with the account id of SYS_ADMIN are not deleted when using the * option.

Section_id:

Type either the appropriate two-character section identifier or an asterisk (*) for all sections. Deleting the registration section deletes all sections. If (user_id)'s profile contains only the RE section, this prompt does not appear.

EP requests confirmation of all deletion requests with one of the following two messages:

Delete section_id from (user_id)?

Delete (user_id)'s entire profile?

Type Y or YES to answer either question in the affirmative and delete the profile or section. Answer N or NO to prevent the deletion.

If you answer either question Y or YES, EP issues the following message to inform you that the deletion process is complete:

```
(section_id) deleted.
```

EP then asks:

```
More deletes? (YES,Y,NO,N)
```

If you answer Y or YES, EP prompts you for the user id as above. If you answer N or NO, the DEL function is exited.

In the following example, the !C and MU sections are deleted from GRIFFITHS.EXSER's profile.

```
Enter function:
DEL
User_id:
GRIFFITHS.EXSER
Section_id:
!C
Delete !C section from GRIFFITHS.EXSER  ?
YES
!C section deleted from GRIFFITHS.EXSER.

More deletes? (YES,Y,NO,N)
YES
User_id:
C/R
Default accepted.
 GRIFFITHS.EXSER
Section_id:
MU
Delete MU section from GRIFFITHS.EXSER  ?
Y
MU section deleted from GRIFFITHS.EXSER.

More deletes? (YES,Y,NO,N)
N

Enter function:
Q
RDY:
```

Duplicating a Section (DUP)

The DUP function of EP allows you to copy a section of one user's profile into another user's profile. You can also use this function to register any number of users with identical attributes. (Of course, different users must have different user ids and login ids). The DUP function dialog is as follows:

Enter function:

Type DUP to copy information from one user's profile to another.

Enter user_id of the original user:

Type the user id of the profile containing the section to be copied in the format NAME.ACCOUNT[.MODE], or accept the default user id.

Section_id:

Type the two-character id of the section that you wish to copy. You can use the star (*) option for this function.

If you specify any value other than RE for section id, EP issues the following message:

Enter receiving user_id:

Type the user id of the profile that is to receive the copied information in the format NAME.ACCOUNT[.MODE]. No default is permitted for this value.

The section is copied into the named user's profile. EP then issues the following two messages:

xx section added to (user_id)'s profile.
Add the same xx section to another profile?

xx is replaced with the section id that you specified earlier.

Type Y or YES to copy the section into another user's profile. Type N or NO to exit the DUP function.

If you specify RE for section id, EP issues the following two messages:

Enter user_id of the new registrant:

Type the user id of the person whom you wish to register. No default is permitted for this value.

Enter (user_id)'s login_id:

Enter the login id of the user whom you wish to register. The default is no login id for the user.

EP then issues the following messages:

(user_id) now registered.
Register another with the same attributes?

Type Y or YES to register another user. Type N or NO to exit the DUP function.

Examples:

The first example illustrates copying the MU section from from GRIFFITHS.EXSER to JONES.USER and A.A.A.

```
Enter function:
DUP
Enter original user_id:
GRIFFITHS.EXSER
Section_id:
MU

Enter receiving user_id:
JONES.USER
MU section added to JONES.USER

Add the same MU section to another user profile?
YES

Enter receiving user_id:
A.A.A

Add the same MU section to another user profile?
NO

Enter function:
Q
RDY:
```

The next example illustrates using DUP to register USER1 and USER2 with the same attributes as GRIFFITHS.EXSER.

```
Enter function:
DUP
Enter original user_id:
GRIFFITHS.EXSER
Section_id:
RE
Enter the user_id of the new registrant.
USER1.TEST.INT
Enter USER1.TEST.INT's login id:
USER1
USER1.TEST.INT now registered.

Register another with the same attributes?
YES
Enter the user_id of the new registrant.
USER2.TEST.INT
Enter USER2.TEST.INT's login id:
USER2
USER2.TEST.INT now registered.

Register another with the same attributes?
N

Enter function:
QT
```

Listing User Profiles (LP)

The LP function lets you list the information contained in one or more sections of one or more user profiles. The dialog is as follows:

Enter function:

> Type LP to specify the List Profile function.

User_id:

> Type the user id in one of the following formats to indicate the user or users whose profile you wish to list, or type DEC to list a subsystem declaration:

User id format:	Lists sections from:
PERSON.ACCOUNT.MODE	A specific user profile
PERSON.*	All user profiles with the specified person id
*.ACCOUNT	All user profiles in the specified account
..*	All user profiles

Type DEC to list a subsystem declaration. Alternatively, accept the default user id.

Section_id:

Type either the two-character abbreviation of the section or sections whose attributes you wish to list; enter * for all sections of the named user profile(s). If you specify more than one section, separate the ids by spaces. For example to list RE, !C, and MU, type

```
RE !C MU
```

EP now displays the requested information on the user-out file. Since the LP function accepts star specifications for user id, there is no default user id for the next EP functions.

The following example lists the information contained in the RE and MU sections of GRIFFITHS.EXSER's profile:

```
Enter function:
LP
User_id:
GRIFFITHS.EXSER
Section_id:
RE MU
```

```
************************************************************************
Registration section:  GRIFFITHS.EXSER.
Last updated: 1982/08/19 1320
    Section attributes
------------------------------------------------------------------------
login id                    GRIFF
default login arguments     -PO * JG -HD >UDD>EXSER>GRIFFITHS -LK EN
current terminal            FC80_COPY
current language key        EN
login traits                SINGLE-USER PROFILE
password status             PASSWORD REQUIRED BUT NOT YET CHOSEN
bad-password attempts       0
session started             1982/08/18 0951
active users                0
************************************************************************

************************************************************************
MU section:  GRIFFITHS.EXSER.
Last updated: 1982/08/19 1346
    Section attributes
------------------------------------------------------------------------
menu catalog pathname       >UDD>GRIFFITHS>MENUS
first menu name             UTILITIES
access roles                AL
user state                  ACTIVE
************************************************************************
```

Displaying and Resetting User Statistics (STATS)

The STATS function lets you display the statistics for a user from a particular section and to reset the statistics to zero, if you wish. After you invoke EP, the system responds with the following dialog:

Enter function:

> Type STATS to display statistics.

User_id:

> Type the user id for which you wish to display or reset statistics in the format NAME.ACCOUNT[.MODE], or accept the default user id. You can reset or list stats for only one user profile at a time.

Section_id:

> Type the appropriate two-character section identifier of the section for which you wish to display statistics.

EP then displays the statistics from the requested section on the error-out file in list form. After displaying the statistics, EP asks:

Reset any statistics to zero?

> Type Y or YES to reset any statistics to zero. Type N or NO to exit the STATS function.

If you answer Y or YES (you wish to reset statistics), EP then displays a list of statistics to reset. The list of statistics depends on the section that you specified. For example, EP lists the following statistics for the RE section:

<u>Choice of Stats to Reset</u>

(1) Last session length
(2) Total session time
(3) Number of sessions
(4) Maximum memory allocated
(5) CPU seconds
(6) Communications I-O count
(7) Disk I-O count
(8) All other I-O count
(9) ALL OF THE ABOVE
(10) NONE OF THE ABOVE

Selection:

> Type one number from the list displayed for the section. If you type the number for ALL OF THE ABOVE (9 for the RE section), all of the statistics are reset to 0. If you type the number for NONE OF THE ABOVE (10 for the RE section), you exit the STATS function without resetting any values.

After you have reset one of the statistics, EP asks:

Reset more statistics?

> Type Y or YES if you wish to reset any other statistics. Type N or NO to exit the STATS function. If you answer Y or YES to this question, EP redisplays the possible choices and prompts for a selection. If you reset at least one of the statistics, the "stats reset" value for this profile is set to the current date.

The following example lists RE section statistics for GLASS.EXSER and resets the "maximum memory allocated" statistic to 0.

```
Enter function:
STATS
User id:
GLASS.EXSER

 RE Section Statistics:  GLASS.EXSER
 Last reset on :  1982/04/08 1215
-------------------------------------------------------------
last session length            0010:25          HHMM:SS
total session time             0319:19          HHMM:SS
number of sessions             16
maximum memory allocated        155
CPU seconds                    8725
communications I-O count       1838
disk I-O count                 30296
all other I-O count            4

Want to reset any statistics to zero?
YES

Choice of stats to reset
-------------------------------------------------------------
   (1)  last session length
   (2)  total session time
   (3)  number of sessions
   (4)  maximum memory allocated
   (5)  CPU seconds
   (6)  communications I-O count
   (7)  disk I-O count
   (8)  all other I-O count
   (9)  ALL OF THE ABOVE
   (10)  NONE OF THE ABOVE

Selection:
4

Any more?  (YES, Y, NO, N)
N

Reset done.
Enter function:
```

Listing the Section Ids of a User's Profile (SEC)

The SEC function lists the sections that are included in a user's profile. The dialog is as follows:

Enter function:

> Type SEC to list the two-character section identifiers of a user's profile.

User_id:

Type the user id for which you wish to list the sections in the format NAME.ACCOUNT[.MODE], or accept the default user id.

EP now lists the section ids included in the specified user profile.

The following example lists the section ids contained in GRIFFITHS.EXSER's profile.

```
Enter function:
SEC
GRIFFITHS.EXSER

Sections in GRIFFITHS.EXSER's profile:  RE, !C, JG, MU

Enter function:
QT
RDY:
```

Declaring a Subsystem

Before you can give users subsystem sections, you must declare the subsystem. You do not have to declare the Menu subsystem (MU) as it is already declared on the Profiles file supplied with your system.

Any subsystem that has been declared can use the facilities of the Profiles file and of EP. If you declare a subsystem called YY, you can then add YY sections to the profiles of users, modify YY sections (using MOD), list and reset statistics kept by YY (using STATS), and delete YY sections (using DEL). The YY subsystem itself can access YY sections using profile monitor calls. For more information on a subsystem's use of the Profiles file, refer to the System Programmer's Guide (Volume I).

To declare a subsystem, first invoke EP. The DEC dialog is as follows:

Enter function:

Type DEC to declare a subsystem.

Enter two character subsystem_id:

Type the two-character identifier of the subsystem being declared. Make sure that this identifer is different from any previously declared identifiers. No default is permitted for this value.

Enter up to 4 bu names for the XX subsystem:

XX is replaced by the subsystem id that you just specified. Type the name or names of one or more bound units that make up the XX subsystem. The bound unit (bu) names must all be typed on one line. The bound units specified here are permitted to use those monitor calls that access the profiles file. Only the first six characters of each name are recorded. Separate names with either spaces or commas. The default is to specify no names (in which case, the subsystem cannot access the profiles file directly).

Does an EP_XX subsystem module exist for Edit_profile's use?

XX is replaced with the previously-specified subsystem id. Answer the question with Y, YES, N, or NO. Answer Y or YES if an EP_XX bound unit exists in a directory that can be reached by the search rules. EP uses the EP_XX subsystem module to process (ADD, MOD, STATS) an XX section. Answer N or NO if XX user profile sections contain only ASCII data and therefore can be maintained by EP without the use of an EP_XX subsystem module. Refer to the System Programmer's Guide (Volume I) for more information on ASCII only type sections.

Does the XX subsystem keep statistics in its user profile sections?

XX is replaced with the previously declared subsystem id. Answer Y, YES, N, or NO. Answer Y or YES if the subsystem collects statistics that can be displayed using EP's STATS function. Answer N or NO otherwise. This question is not asked if subsystem XX is an ASCII only subsystem.

EP now displays

XX subsystem now declared.

Refer to "Deleting a Subsystem (DEC)" for an example of declaring a subsystem.

Deleting a Subsystem (DEC)

The DEC function also permits you to delete the declaration of a previously declared subsystem. Once you have deleted a subsystem, you can no longer refer to it using EP (for example, you can no longer register users of the subsystem using ADD). The deleted subsystem can no longer access the Profiles file. The dialog for DEC is as follows:

Enter function:

Type DEC to delete the declaration of an existing subsystem.

Enter two character subsystem_id:

Type the two-character identifier of the subsystem that you wish to delete. EP checks to see if the specifed subsystem exists. If it does, the dialog continues. If the subsystem does not exist, EP assumes that you want to declare it and initiates the dialog described in "Declaring a Subsystem."

XX subsystem already declared.
Delete XX declaration section?

XX is replaced with the previously specified subsystem id. Answer Y, YES, N, or NO. Answer Y or YES if you wish to delete the declaration of XX. Answer N or NO to exit the function with no changes made.

If you answer Y or YES, EP issues the following message:

XX subsystem declaration section deleted.

The following example illustrates the declaration of two subsystems, XX and WS. WS is an ASCII-only subsystem. The example also illustrates deleting the XX subsystem.

```
Enter function:
DEC
Enter two character subsystem_id:
XX
Enter up to 4 bound_unit names for the XX subsystem:
XXBU1 XXBU2
Does an EP_XX bound unit exist for Edit_profile's use?
YES
Does this subsystem keep statistics?
YES
XX subsystem now registered.

Enter function:
DEC
Enter two character subsystem_id:
WS
Enter up to 4 bound_unit names for the WS subsystem:
WSSUB
Does an EP_WS bound unit exist for Edit_profile's use?
NO
WS subsystem now registered.
```

Enter function:
DEC
Enter two character subsystem_id:
XX
XX subsystem already declared.

Delete XX subsystem declaration section?
YES

XX subsystem declaration section deleted.

Enter function:

Terminating EP (QUIT)

The Quit function permits you to terminate EP. The dialog is as follows:

Enter function:

Type Q or QT to terminate EP.

LIST PROFILE COMMAND

The List Profile (LP) command permits an administrator to list the contents of any user's profile or to list a subsystem declaration. The LP command performs the same function as the LP function of EP. The LP command could be included in a command file, unlike the LP function of EP. Using LP, you can selectively list sections from the profiles of specific users, all users registered in a specific account, or all registered users. You can also use LP to list a subsystem declaration. LP lists the requested information on user-out. LP is not interactive; you must enter all required arguments when you type the command.

FORMAT:

```
LP user_id   [section_id]
             [*]
```

ARGUMENTS:

user_id:

Type the user id in one of the following formats to indicate the user or users whose profile you wish to list, or type DEC to list a subsystem declaration:

User id format:	Lists sections from:
PERSON.ACCOUNT.MODE	A specific user profile
PERSON.*	All user profiles with the specified person id
*.ACCOUNT	All user profiles in the specified account
..*	All user profiles
DEC	List a subsystem declaration

You must specify this argument.

Section_id:

Type either the two-character abbreviation of the section or sections whose attributes you wish to list; enter * for all sections of the named user profile(s). If you specify more than one section, separate the ids by spaces. For example to list RE, !C, and MU, type

```
RE !C MU
```

If you do not specify a value for section id, LP displays the RE section of the specified profile or profiles.

DESCRIPTION:

LP displays the requested information on user-out in the same format as that of the LP function of EP. Refer to "Listing User Profiles (LP)" earlier in this section for an example.

OPTIONAL RE SECTION STATISTICS

When you register a user or when you modify the user's registration, you can specify that a packaged set of statistics be maintained on the user's system usage. These statistics are kept in the user's RE section and can be displayed and individually reset using the STATS function of EP. The LP function of EP and the LP command can also be used to display these statistics if they are maintained for a user.

The package of RE section statistics that you can maintain for a user consists of the following statistics:

- Last session length

  The elapsed time in the format HHHHH:MM:SS (HHHHH is the number of hours; MM is the number of minutes; SS is the number of seconds) of the user's last session with this user profile.

- Total session time

  The total elapsed time in the format HHHHH:MM:SS of all sessions under this user profile since statistics were last reset.

- Number of sessions

  The number of sessions (logins) under this user profile since statistics were last reset.

- Maximum memory allocated

  The largest block of memory ever used by this user since statistics were last reset. This value is displayed in multiples of 1K word blocks.

- CPU seconds

  The total number of seconds of CPU time used by the user logged in under this profile since statistics were last reset.

- Communications I-O count

  The total number of communication device I/O orders since statistics were last reset.

- Disk I-O count

  The total number of disk device I/O orders since statistics were last reset.

- All other I-O count

  The total number of noncommunication, nondisk I/O orders since statistics were last reset.

If you request that the RE statistics be maintained for a user, EP also maintains the following items:

- Stats updated

  The date and time that the statistics were last updated. This is the date and time either when the user's last session ended or when the statistics were last reset using EP, whichever is most recent.

- Last session begun

  The date and time of the beginning of the last complete session under this user profile.

- Last session ended

  The date and time that the user last logged out from this user profile.

- Last terminal used

  The name (as defined in the CLM_USER file) of the last terminal on which the user logged into this user profile.

- Date registered

  The date and time that the collection of the optional set of RE statistics was first specified for this user.

- Stats reset

  The date and time when any one or more of the statistics was last reset using the STATS function of EP.

All date and time values are rounded to the nearest minute. All elapsed time values are rounded to the nearest second and have a maximum value of 50,000 hours.

INFORMATION NOT COLLECTED FOR MULTIUSER PROFILES

Some information is only meaningful for single-user profiles. The system does not maintain or display the following items for multiuser profiles:

- Current terminal
- Last terminal used
- Last session begun
- Last session ended
- Total session time.

Any information that is maintained for a multiuser profile (e.g., maximum memory allocated, CPU seconds, and I-O counts) consists of the totals for all users logged in under the multiuser profile.

PASSWORD OBSOLESCENCE

A password becomes obsolete 90 days after it is chosen or changed. On the first login to a profile whose password has become obsolete, the user must choose a new password. In multiuser profiles, however, passwords do not automatically become obsolete. The system administrator can enforce obsolescence on any profile (including multiuser profiles) with Edit Profile (EP) by editing that profile, entering the MOD function, selecting (4), "Password Status," and specifying option (1), "Allow user to choose a new password at next login." The system administrator should make sure that all users of the profile are aware of the change, which will be made by the next user to login to the profile.

REMOVE THIS PAGE AND PLACE TAB FOR

TAB 5

BASIC CONFIGURATION

Section 5
BASIC CONFIGURATION LOAD MANAGER DIRECTIVES

This section describes the Configuration Load Manager (CLM) directives used to define the basic characteristics of the system. The CLM directives for the optional communications environment of the system are described in Section 6. If you are a first-time user, you should be thoroughly familiar with all the topics discussed at the beginning of this section before you create your CLM directive file.

Appropriate CLM directives (including those for the communications-connected operator's console and any other applicable communications devices) must be placed in the CLM_USER file under directory >>SID on the bootstrap volume before a stage 2 system startup is performed. (The two stages of system startup are described in Section 2.)

Included in appendixes at the back of this manual are several procedures for configuring special functions. These functions and the appendixes that describe the procedures for configuring them are as follows:

To configure:	Refer to:
• Timeslicing	Appendix B
• Power Resumption Facility	Appendix D
• Error Logging	Appendix F
• Display Formatting and Control (VDAM)	Appendix G
• Disk Cache	Appendix K
• Command Accounting	Appendix P
• Terminal Presentation Facility (TPF)	Appendix Q

Table 5-1 summarizes the basic CLM directives, described in alphabetic order later in this section. Briefly, these CLM directives define the following aspects of the system:

- System variables such as power line frequency, scan cycle of real-time clock, Scientific and/or Commercial Instruction Processor options, number of system overlay areas, number of trap save areas, number of intermediate request blocks, and expanded error message code option (SYS directive)

- The legend displayed on the banner page of spooled printed output to identify the installation (INSID directive)

- Characteristics and sizes of memory pools (MEMPOOL and SWAPPOOL directives)

- Physical and logical characteristics of MDC-connected peripheral devices (DEVICE and BOOTCOMP directives)

- System overlays that are to be made permanently resident (RESOLA directive)

- Operating System extensions that are to be made permanently resident (LDBU, ILDBU, and DLDBU directives)

- An informational printout detailing system physical attributes, system structures, device status, user-defined variables, and memory pool specifications (MAP directive)

- Single character expressions that can be used in certain directives instead of constant values (VARIABLE directives)

*

The configuration of the following devices is described in appendixes: Decision Data 8045 Card Reader/Punch (Appendix L), FACIT 4042 Paper Tape Reader/Punch (Appendix M), and Memodyne M-80 Cassette Tape unit (Appendix N).

Table 5-1. Summary of Basic CLM Directives

Directive	Meaning
ASSIGN	Used to explicitly assign a controller to a central processor unit in a multiprocessor environment.
BOOTCOMP	Configures the boot device companion when booting from a fixed/removable device.
CLMIN	Causes subsequent CLM directives to be read from a different device/file.
* (comment)	Used to include a comment in the file of CLM directives.
DLDBU	Identifies a bound unit (Honeywell-supplied or user-written) that is to be made permanently resident in the system area of memory. The bound unit is loaded after the bound units identified in LDBU directives.
DEVICE	Identifies a peripheral device and its characteristics, both physical and logical. *
ILDBU	Identifies a bound unit (Honeywell-supplied or user-written) that is to be made permanently resident in the system area of memory. The bound unit is loaded when the ILDBU directive is encountered.
INSID	Changes the default installation ID.
LDBU	Identifies a bound unit (Honeywell-supplied or user-written) that is to be made permanently resident in the system area of memory. The bound unit is loaded after the QUIT directive is encountered.
MAP	Provides configuration information that allows you to verify the success of certain aspects of the system building process.
MEMPOOL	Defines the system memory pool or one or more independent memory pools.
MLPATH	Specifies a Message Library other than the default one. Also specifies a (national) language key.

Table 5-1 (cont). Summary of Basic CLM Directives

Directive	Meaning
QUIT	Indicates the end of the CLM directives.
REBOOT	Used to configure the Software Reboot facility.
RESOLA	Identifies one or more system overlays that are to be made permanently resident in the system area of memory.
RLOCK	Defines a pool of structures to be used by data management for the Record Locking function.
SWAPPOOL	Defines one or more swap pools to support swapping of user tasks.
SYS	Defines a number of system variables.
SYS2	Defines system variables including those associated with use of a swap pool.
VARIABLE	Allows you to define variables for arguments in certain CLM directive statements.

RULES FOR ARRANGING CLM DIRECTIVES

Only a few rules apply to the order of basic directives in the input file read by the CLM. These rules are listed below. (Rules governing the order of communications-related CLM directives are described in the beginning of Section 6.)

1. Each noncommunications peripheral device in the system must be identified in a separate DEVICE directive. (As described in Section 6, communications devices must be identified in DEVICE directives only if they are to be accessible through the File System.)

* 2. If your configuration requires more than the default number of Trap Save Areas (TSAs) (6 or 7) and/or more than the default number of Intermediate Request Blocks (IRBs) (20), a SYS directive with appropriate TSA and IRB argument values should be placed as close as possible to the beginning of the file of CLM directives. The additional TSAs and/or IRBs are available as soon as the SYS directive is read.

3. If the CLMIN directive is used, it must not precede the DEVICE directive that identifies the device implied by the pathname argument of the CLMIN directive.

4. If the ILDBU directive is used, the device on which the bound unit file resides must have been previously configured with a suitable DEVICE directive.

5. If the ASSIGN directive is used, the directive must precede the DEVICE directive that defines the controller channel number.

6. If LDBU directives are used to identify both a SIP simulator and a CIP simulator, the simulator identified in the second LDBU directive is the first to process a trap to the trap vector shared by the two simulators.

7. A variable must be defined by a VARIABLE directive in the CLM file before it can be used in subsequent directives.

8. QUIT must be the last CLM directive in the input file. Any directives that follow QUIT are not processed.

FORMAT OF CLM DIRECTIVES

A CLM directive consists of a string of up to 140 ASCII characters. The format of a CLM directive is:

```
mnemonic argument_1[,argument_2][,argument_3]...[,argument_n]
```

In the directive descriptions in this section and in Section 6, the following conventions apply:

1. The directive mnemonic is shown in uppercase. It must be specified exactly as shown.

2. The directive mnemonic must begin in the first column. If there is a space in column 1, the directive is ignored.

3. Exactly one space must separate the directive mnemonic from the first argument.

4. In arguments, lowercase words and letters indicate values that must be specified by the user.

5. Arguments within square brackets ([]) are optional.

6. Vertically stacked arguments within braces ({}) represent options, one of which must be chosen when the argument is used.

7. Arguments in a directive are separated by commas. Embedded blanks are not allowed within arguments. Blanks are not allowed between arguments.

8. A blank terminates the list of arguments in a directive. Anything following this blank is considered a comment.

9. All arguments are positional, meaning that they must be specified in the order shown in the directive description. If a given argument is omitted and a following argument is used, the omitted argument must be signified by delimiting commas; for example,

```
DEVICE  LPT02,15,22,X'1380',,,N
```

10. Trailing commas are not required after the last argument.

11. Continuation lines are not allowed.

12. Unless stated otherwise, unsigned positive integers less than or equal to 65,535 (FFFF hexadecimal) can be expressed in decimal or hexadecimal. A decimal integer consists of one or more decimal digits (for example, 1234). A hexadecimal integer consists of the constant X followed by four hexadecimal digits expressed in the format: X'hhhh' (for example, X'FF80').

    If a value exceeds 65,535, it must be expressed in double-word hexadecimal format (that is, D'hhhhhhhh').

13. If an ASCII character string begins with a decimal digit, the string must be enclosed within apostrophes (for example, '1A').

14. The line length specified for a terminal or other input device at system building time overrides the command processor line length of 127 characters. For example, at a terminal configured with a line length of 80 characters, you cannot enter a continuous command line longer than 80 characters unless you type an ampersand (&) at the end of the line and then continue on one or more additional lines. You can alter the terminal's line length by using the Set Terminal Characteristics (STTY) command (refer to the Commands manual) after system initialization or by modifying the STTY directive in your CLM file.

15. For each noncommunication device configured, you have the option of specifying a channel number or allowing the system to dynamically assign or "float" channel assignments at startup time. By assigning floating channel numbers to devices in the CLM file, you need not know the channel numbers at each installation that receive a copy of the system software. To float a channel, you specify the channel number argument in the * appropriate DEVICE directive with a zero (0). The system then assigns appropriate devices to any available channels. Note that if you specify a zero channel and the system cannot locate an appropriate device for it, no error message is issued.

Channel numbers are assigned according to the order in which DEVICE directives with zero (0) channel numbers are specified in the CLM file. To prevent directives with zero channel numbers from inadvertently preempting directives whose channel numbers are explicitly specified, all directives of a given type with explicitly specified channel numbers should precede all directives of that type with zero channel numbers in the CLM file. See Appendix H for more information about channel numbers.

For noncommunication devices, the first available device having the lowest channel number and the correct device identification is assigned to the first available MDC-connected DEVICE directive encountered in the CLM file, and so on.

If a system is configured with directives containing zero channel numbers, you can determine all channel numbers assigned by the system after system initialization. The system operator can issue the command STS -ALL to list each device and its channel number assignment.

When the system identifies the presence of a device adapter, it assumes that the corresponding channel numbers are all valid and usable. In such a case, the system may associate an unusable channel with a physical device. If a system includes device adapters but not associated devices, then explicit channel numbers should be assigned to all devices within those device classes. *

PRELIMINARY CONSIDERATIONS

The following paragraphs describe:

- How to incorporate Executive overlays and user-written bound units into your configuration
- Memory allocation and usage
- Performance considerations
- How to identify your peripheral devices
- How to configure the operator terminal
- File System pathnames in CLM directives.

You should review each of these topics before you create your CLM directive file.

System Overlays

Within the system software, many system functions are implemented as overlays. Each system overlay runs in a 512-word system overlay area in the system area of memory. By default, one 512-word system overlay area is created. If you wish, you may use the "olan" argument of the SYS directive to create additional 512-word system overlay areas (up to a maximum of 99) so that the system loader is not forced to repeatedly load different overlays into a single system overlay area as various system overlay functions are required.

In addition to, or instead of creating multiple system overlay areas, you may wish to make certain frequently used system overlays permanently resident in the system area of memory (increasing its size at the expense of memory available for user tasks). This can be done by means of the RESOLA directive. Each RESOLA directive allows specification of one or more system overlays, each of which is loaded for permanent residency during system startup. The names and approximate sizes of all system overlays are listed in the Software Release Bulletin.

System Extensions

A system extension is an optional, user-written or system-supplied bound unit (consisting of reentrant code) that is identified in either an LDBU, ILDBU or DLDBU directive and loaded, for permanent residency in the system area of memory, during system configuration. The SYS Linker directive must be used when the system extension is linked. Multiple system extensions can be used.

System extensions are most efficient when they are more or less continuously used through symbolic references from multiple application task groups, and this usage does not permit a possible delay in the initial loading of the extension.

System extensions can be used to define systemwide global address symbols. These symbols must have been defined in EDEF Linker directives as each extension bound unit was linked. Later, as the bound unit is loaded during system configuration (by virtue of either an LDBU, ILDBU or DLDBU directive), the systemwide global address symbols are added to the system symbol table, where they can be used (by the system loader) to resolve any unresolved references to them occurring in subsequently loaded bound units.

Any dynamic requests for memory originating from a permanently resident system extension are fulfilled from the memory pool of the task group that is using the extension.

Note that if the system extension code is not concurrently used by multiple applications, it can be individually linked into applicable user-written bound units and not loaded for permanent residency at system startup. In this case, an LDBU directive is not used, and the extension resides in memory only as part of each user-written bound unit with which it has been linked. Appropriate symbols in system extensions used in this way can be identified in XDEF Assembler control statements and thus resolved at link time.

Memory Allocation and Usage

System startup allows main memory to be divided into the following areas (as a maximum number of areas):

- System area
- System memory pool
- Independent pools *
- Swap pools.

The system area contains resident system software, one or more system overlay areas (as specified by the "olan" argument of the SYS directive), any permanently resident system overlays (as specified in RESOLA directives), and any permanently resident system extensions (as specified in either LDBU, ILDBU or DLDBU directives). MEMPOOL and SWAPPOOL directives establish the types * and sizes of one or more memory pools. Except for the system * pool, the MEMPOOL and SWAPPOOL directives also establish the two-character identifier of each memory pool.

Roots of bound units linked as globally sharable are loaded into the system memory pool, whereas roots of bound units linked as nonshareable are loaded into the pool in which the task group is running. Each task group is associated with one and only one memory pool, but more than one task group can be associated with the same memory pool. See the Systems Concepts manual for a more detailed description of memory pools.

Fragmentation

Fragmentation is the development, within a memory pool, of unusable "holes" or "fragments" of memory. It usually results from the dynamic acquisition and release of memory within a pool by a number of concurrently active tasks. Since the system's memory manager satisfies dynamic requests for memory within a pool on a first-come/first-served basis in consecutive multiples of 32-word blocks, it is possible that the order of memory acquisition and release can create a situation wherein a request for memory cannot be satisfied even though sufficient free memory exists.

This situation might occur, for example, when a task has successfully obtained four areas of memory within a pool and the pool's free memory is nearly exhausted. If the task released the second area (for example, 128 words) and the fourth area (for example, 96 words), the task's subsequent request for 192 words might be unsuccessful because that much consecutive memory is not available (even though at least 224 words--the amount released--is available in two nonconsecutive areas).

Note that if tasks obtain and release memory on a last-obtained/first-released basis, the potential for fragmentation is reduced, especially when only a few tasks share a particular memory pool.

The potential for fragmentation increases with the number of tasks sharing a memory pool and with these tasks' dynamic use of memory. Calculations for estimating memory pool size described in the Software Release Bulletin include a factor to allow for fragmentation.

Identifying Peripheral Devices

Each noncommunication peripheral device in the system, except the bootstrap device, must be identified in a separate DEVICE or BOOTCOMP directive. The device can be designated as accessible through the File System or accessible only through physical input/output. (As described in Section 6, communication devices must be identified in DEVICE directives only if they are to be accessible through the File System.)

The DEVICE directive's arguments specify the characteristics of the device:

- Unique device-unit identifier
- Priority level number
- Channel number on Megabus network
- Unique logical resource number (lrn)
- Unique device name for File System references
- Maximum record size
- * Buffered/unbuffered input/output.

Configuring the Operator Terminal

The following paragraphs describe characteristics of the operator terminal and the various configuration options available to you.

OPERATOR TERMINAL CHARACTERISTICS

The operator terminal is the single device designated as the control terminal for the system. It must be assigned to lrn 0 at system configuration time in order to be identified by the system.

The operator terminal is the only terminal from which communication with the system task group $S is possible. Capabilities unique to the operator terminal include the suspension and reactivation of task groups, peripheral device * control, print daemon startup, and the monitoring of system status.

Normally, an operator terminal is configured during a stage 2 system startup. (An operator terminal is always configured automatically in a stage 1 system startup.) The operator terminal is a terminal connected to a communications controller. *

CONFIGURATION OPTIONS

As the system builder, you have three options for configuring the operator terminal:

1. You can configure a terminal that functions as the operator terminal for the life of the system.

2. You can configure a dual-purpose terminal that can function alternately as an operator terminal and as a nonoperator terminal.

3. You can choose not to configure an operator terminal.

You might choose the first option if you wish to always have one terminal reserved for the operator's exclusive use. This terminal would always have the capability to perform those control and administrative functions unique to the operator. Note that the operator terminal cannot support forms processing software.

The second configuration option allows you to assign a dual function and identity to the operator terminal. You assign two unique names to the terminal: a "device name" that identifies the terminal when it is functioning as a nonoperator terminal, and an "operator terminal name" that identifes the terminal when it is functioning as an operator terminal. You might choose this configuration option if operator functions will be infrequently performed and you wish to free the operator terminal for user applications (especially forms processing). The ability to perform forms processing is only available to users when the terminal is functioning as a nonoperator terminal.

Operating procedures for an operator's console configured by either of the above two methods are fully described in the System User's Guide.

The third configuration option does not permit commands and procedures unique to the operator terminal to be performed. You might choose this option if your system configuration will run under a dedicated application.

Configuring an Operator Terminal to Function for the Life of the System

A device configured to function as the operator terminal for the life of the system always allows users the capability to perform those control and administrative functions unique to the operator.

The permanently assigned operator terminal is connected to a communications controller. Both a DEVICE directive and an appropriate communication directive (for example, ATD) are required. A COMM directive is also required. A communications-connected operator terminal is activated only after system startup is complete. The operator terminal is not available to receive any error messages that relate to CLM processing.

The logical resource number on the DEVICE directive must be explicitly specified as 0. Any appropriate level number and channel number can be specified. The record_size argument should normally be set at 140 bytes (characters). Note that the specified record size affects only the File System's use of the operator terminal. Regardless of the value of record_size, the Operator Interface Manager always uses 140 characters as the maximum for its control of input and output operations to the operator terminal. The device name of the operator terminal can be any user-selected name (customarily, CONSOLE is used).

Example:

```
COMM 9
DEVICE ATD00,0,11,X'FF80',CONSOLE,140
ATD 0,11,X'FF80'
```

In this example, the operator terminal is connected to a communications controller. The device unit name is ATD00. The logical resource number is 0 (a requirement for the operator terminal). The priority level is 11. The channel number is FF80. The device name is CONSOLE. The record size for File System access is 140 bytes. In the COMM directive, 9 is the priority level at which the communications controller interrupts the central processor. In the ATD directive, default values have been accepted for modem type, line speed, and device type.

Configuring a Dual-Purpose Operator Terminal

You have the option to configure a dual-purpose operator terminal. A dual-purpose terminal can function alternately as an operator terminal and as a nonoperator terminal. Only one device in your configuration can be configured in this manner. Users control the mode in which the terminal functions (as an operator terminal or nonoperator terminal) by means of the OPER command, described later in this section. *

The dual-purpose terminal is connected to a communications controller. Both a DEVICE directive and an ATD directive are required. A COMM directive is also required. A communications-connected operator/nonoperator terminal is activated only after system startup is complete. The operator terminal is thus not available to receive any error messages that relate to CLM processing. It is recommended that the terminal not be a dialed line.

For a communications-connected device, the ATD and DEVICE directives are of the form:

```
DEVICE ATDnn,lrn,level,X'channel',device_name,rec_size,
       N,,,operator_terminal_name

ATD lrn,level,X'channel',modem,speed,device_type,[del],
    [stop-bit][,parity]
```

The logical resource number on the DEVICE directive must be explicitly specified and must be nonzero. The logical resource number you specify is assigned to the device and the file having the name device_name (fifth argument). The system implicitly assigns logical resource number 0 to the file having the name operator_terminal_name (tenth argument). Logical resource number 0 cannot be assigned to any other device. An appropriate level number and channel number can be specified. The device name (fifth argument) you specify is assigned to the terminal when it is functioning as a nonoperator terminal. An application task group's standard I/O files (for example, command-in, user-in) are assigned to the device name you specify. Only one application task group can be assigned to the device name you specify. You must specify that this device is unbuffered; the seventh argument must be N. *

For a communications-connected terminal, the tenth argument, operator_terminal_name, must be separated from the seventh argument, N, by exactly three commas. The operator terminal name you specify is assigned to users' standard I/O files when the terminal is functioning as an operator terminal. CONSOLE is the customary designation. Any number of application task groups can be assigned to the operator terminal name you specify. *

Example:

```
COMM 8
DEVICE ATD00,9,15,X'FF80',MYVIP,80,N,,,CONSOLE
ATD 9,15,X'FF80',0,9600,'7200'
```

In this example, the operator terminal is connected to a communications controller. The device unit name is ATD00. The logical resource number is 9. The explicitly stated logical resource number value of 9 is assigned to the terminal when it functions as a nonoperator terminal. A logical resource number of 0 is implicitly assigned to the terminal when it functions as an operator terminal. The priority level is 15. The channel number is FF80. In the COMM directive, 8 is the priority level at which the communications controller interrupts the central processor. In the DEVICE directive, the name assigned to the device when it is functioning as a nonoperator terminal is MYVIP. The record size for file system access is 80 bytes. The terminal must be designated as unbuffered (the seventh argument must be N). The name assigned to the terminal when it is functioning as an operator terminal is CONSOLE.

In the ATD directive, the modem type is 0 (direct connect). The line speed is 9600. The device type is 7200. Defaults have been taken for del, stop-bit, and parity.

At system startup, the terminal is running under the system task group and functions as an operator terminal. It will not function as a user terminal until you do one of two procedures:

a. Spawn an application task group whose user-in file is the nonoperator terminal device name.

b. Spawn the Listener task group ($L) with the device name of the nonoperator terminal included in the Listener Terminals file.

Whichever procedure you use, it must be done within an EC file and must observe the following constraints:

1. A Spawn Group command must be preceded by an

   ```
   OPER -OFF  -IM
              -DEF
   ```

   command. These commands are described below.

2. Both commands should occur well after any other commands within the EC file that causes output to be sent to the operator terminal. If output is sent to the operator terminal after the OPER -OFF command has been associated with an active task group, the use of the terminal may be temporarily delayed.

The Spawn Group (SG) command initiates a task group which will be able to access the nonoperator terminal; that is, the logical device named in the DEVICE directive as argument 5. (The CG and EGR commands can be substituted for the SG command.)

The OPER -OFF command transfers control of the terminal from operator control to user control. The -IM control argument (the default) allows all messages processed by the Operator Interface Manager (OIM) to be displayed while the terminal is running under an application task group. The -DEF control argument defers all OIM-processed messages until the terminal is returned to system task group control. However, messages sent to the terminal by two system routines, OPMSG and OPRSP, are never deferred. These messages are displayed immediately to allow the operator or user to respond to volume mount requests or "device not ready" messages. Messages sent to the terminal from other terminals using the MSG command are also not deferred.

Whichever procedure you use to activate the nonoperator terminal (either an application or Listener), you have a choice of when the procedure is executed:

a. Executed automatically at the end of system startup. To have the procedure execute automatically, place the commands at the end of the system startup EC file (>>START_UP.EC).

b. Executed when initiated by the operator. To have the procedure executed by the operator, place the commands in a special EC file whose purpose and location are communicated only to selected users. The operator can execute the EC file when desired (output to the operator terminal should not be pending).

Example 1:

```
OPER -OFF
SG AX USERX.PROJA 0 !TERM00 -WD >UDD>X -POOL AB
```

In this example, the nonoperator terminal is attached to user application group AX. The lead task must be the Command Processor in order for the operator group to remain accessible by using the OPER -ON command. It is helpful if the file >UDD>X>STARTUP.EC causes some output to the group's user-out or error-out file (!TERM00) so that it will be evident when the terminal is ready to be used in nonoperator mode. Once it is ready, you can access the operator group by using the OPER -ON command.

Example 2:

```
OPER -OFF
EC >>GROUP$L
```

In this example, the nonoperator terminal is attached to Listener task group $L. The Honeywell-supplied Listener group activation EC file is executed. The terminal is subsequently ready for use when it displays the login form or banner. From then on, you can only use the terminal by logging in as a user. Once you are logged in, you can access the operator group by using the OPER -ON command.

A System Configured Without an Operator Terminal

A system configured without an operator terminal has the characteristics listed below. In a system configured with an operator terminal, these same characteristics apply within any task group that has used the $CMSUP (Console Message Suppression) macrocall to temporarily suppress messages to the operator terminal.

- A $OPMSG (operator information message) or $OPRSP (operator response message) macrocall to the nonexistent operator terminal results in a 0802 error return (invalid logical resource number) to the issuing application.

- No operator terminal is available to record system error messages and ready device messages and mount volume messages. As a result, error returns to the issuing application occur immediately:

  - An input/output order to a device that is offline or not ready results in a 0105 error return (device not ready) to the application.

  - A GET command or $GTFIL (get file) macrocall to reserve a volume that is not mounted results in a 020C error return (volume not found) to the application.

  - A read or write hardware error results in a 0107 error return (hardware error) to the application.

- No system-supplied message can be issued to signal the termination of system configuration.

File System Pathnames in CLM Directives

Whenever the format of a CLM directive indicates that a File System pathname is to be supplied as an argument, the pathname must be expressed in one of three forms:

1. A full absolute pathname.

   This form of pathname is required to identify a file on a disk volume other than the system (bootstrap) volume.

   A full absolute file pathname begins with a circumflex (^) and a disk volume root directory name (which is the same as the volume id). Each successive element in the pathname is preceded by a greater-than sign (>) and is hierarchically subordinate to the preceding element. Every element in the pathname except the last is a directory name. The last (rightmost) element is a file name. The last element is not followed by any symbol or punctuation. (In non-CLM environments, the last element in a pathname can be either a directory name or a file name, as appropriate to the situation.)

   Example:

   ```
   CLMIN ^USRVOL>DIR_1>FILE_A
   ```

   In this example, a full absolute pathname is used in a CLMIN directive. The pathname indicates a target file that exists on a disk volume other than the system (bootstrap) volume.

2. An absolute pathname.

   This form of pathname is used to identify a file on the system (bootstrap) volume or a nondisk peripheral device.

   For the system volume, an absolute pathname begins with two greater-than signs (>>) and the name of a directory or file that is immediately subordinate to the root directory of the system volume. For a magnetic tape file, the absolute pathname must be !device_unit>volume_id>file_name. The value for device_unit must be specified in the related DEVICE directive. For other nondisk peripheral devices, the absolute pathname must be !device_name. The value for device_name is as specified in the related DEVICE directive. (If device_name is not specified in the DEVICE directive, the value of the device_unit argument is used for device_name.) The rest (if any) of the pathname is similar to the full absolute pathname described above.

Example:

LDBU >>DIR_Y>USRFIL

In this example, an absolute pathname is used in a LDBU directive. This pathname indicates a target file that exists on the system root volume. DIR_Y is a directory immediately subordinate to the root directory of the system volume.

3. A relative pathname.

During execution of the CLM, this form of pathname is used to identify a file in the system task group's current working directory, normally directory >>SID, which is immediately subordinate to the root directory of the system volume. ("Relative" means relative to the working directory of the task group.)

In this situation, the relative pathname is expressed as a simple file name with no preceding or following symbols or punctuation. Other forms of the relative pathname exist, but their use with the CLM is discouraged. The named file must exist in the current CLM working directory (normally >>SID).

Example:

LDBU RBRCIP

In this example, a relative pathname is used in an LDBU directive. This pathname indicates a target file that exists immediately subordinate to the current CLM working directory (normally >>SID on the system volume).

In all types of pathnames, a directory name cannot exceed 12 characters, a file name cannot exceed 12 characters, and the total pathname (including ^ and > characters) cannot exceed 57 characters. In a full absolute pathname, the volume root directory name cannot exceed six characters. The first character of each directory name and each file name must be alphabetic or a dollar sign ($).

BASIC CLM DIRECTIVES SET

The basic CLM directives are described on the following pages, in alphabetical order.

ASSIGN CONTROLLER (ASSIGN)

The ASSIGN directive is used in a multiprocessor environment to explicitly assign a controller to a specific central processor unit (CPU). This is done for the purpose of load balancing. CLM performs load balancing automatically. Therefore, the ASSIGN directive should only be used when fine-tuning the system performance.

FORMAT:

ASSIGN controller,CPU_channel

ARGUMENTS:

controller

The four-digit hexadecimal channel number on the controller to be assigned. The leftmost 10 bits specify the channel number; the rightmost 6 bits must be zero. Channel numbers must be specified in hexadecimal notation; e.g., x'cccc'.

CPU_channel

The four-digit hexadecimal channel number of the CPU to which the controller is to be assigned.

Valid values are:

x'0000' for CPU 0
x'0040' for CPU 1
x'0080' for CPU 2
x'00C0' for CPU 3
*   for boot CPU

DESCRIPTION:

This directive is used in a multiprocessor environment for load balancing. To aid in load balancing, CLM creates a controller table for each controller. When the ASSIGN directive is encountered, CLM updates the controller table for the specified controller to contain the indicated CPU number. The controller table is then marked to indicate that the controller is assigned.

NOTE

> Assignment of a controller is done implicitly by the DEVICE directive. If the controller is to be explicitly assigned by the ASSIGN directive, the ASSIGN directive must precede the appropriate DEVICE directive.

No errors are reported during execution of this directive. However, the following conditions cause the directive to be ignored:

- An invalid CPU channel number
- CPU not present
- Controller not present
- Controller already assigned.

Example 1:

```
ASSIGN x'1200',x'0040'
```

In this example, the controller on channel 1200 is assigned to CPU 1.

Example 2:

```
ASSIGN x'0400',*
```

In this example, the controller on channel 0400 is assigned to the boot CPU.

BOOT COMPANION DEVICE (BOOTCOMP)

The BOOTCOMP directive can be used to configure the bootstrap device companion when booting from one of a fixed/removable device pair.

FORMAT:

BOOTCOMP lrn

ARGUMENT:

lrn

The logical resource number that represents the companion device.

DESCRIPTION:

When the BOOTCOMP directive is issued, the companion to the bootstrap device is immediately configured. Thus, if booting from a removable LARK, the companion is the fixed LARK on the same channel; if booting from a fixed LARK, the companion is the removable LARK on the same channel.

The device type number for both devices is 00 (e.g., RLK00 and FLK00). The logical resource number of the bootstrap device is always 1. The level, which can be changed (refer to the description of the DEVICE directive), is 6. The logical resource number of the companion is the logical resource number value given in the BOOTCOMP directive. The level remains the same as for the bootstrap device.

The BOOTCOMP directive can be issued anywhere in the CLM_USER file. If the boot level is changed, however, the change must be made before the BOOTCOMP directive.

The advantage of the BOOTCOMP directive is that it allows you to create a universal CLM_USER file, as appears in the following example. In a system to be configured with two LARKs and two CMDs, only one CLM_USER file is needed if the BOOTCOMP directive is included. The relevant portion of a sample CLM_USER file is as follows (the logical resource numbers and level numbers are illustrative only):

Example:

```
BOOTCOMP 3
.
.
.
DEVICE RLK01,4,10,0
DEVICE FLK01,5,10,0
DEVICE RLK02,6,11,0
DEVICE FLK02,7,11,0
DEVICE RCM01,8,12,0
DEVICE FCM01,9,12,0
DEVICE RCM02,10,13,0
DEVICE FCM02,11,13,0
```

Note that device number 00 is not specified in any DEVICE directive and that the channel numbers are floated. If the bootstrap device is a fixed or removable LARK, then the devices are configured as device types RLK00, FLK00, RLK01, FLK01, RCM01, FCM01, RCM02, and FCM02. RLK02 and FLK02 are ignored.

The BOOTCOMP directive creates an internal DEVICE directive that can be displayed when an error occurs. For example, if booting from a removable LARK, the following DEVICE directive is created for its fixed companion (assuming a logical resource number argument value of 3 and boot channel X'0480'):

```
DEVICE FLK00,X'0003',X'0006',X'0480'
```

CLM INPUT (CLMIN)

A CLMIN directive changes the source of CLM directives from the current device/file to the one indicated by the pathname argument. All subsequent CLM directives are read from the indicated device/file (unless it too contains a CLMIN directive).

FORMAT:

```
CLMIN pathname[,new_work_dir]
```

ARGUMENTS:

pathname

An ASCII string that identifies the File System pathname from which subsequent CLM directives are read.

[new_work_dir]

An ASCII string that identifies the File System pathname for a new current working directory. >>SID is the initial working directory.

DESCRIPTION:

In a file of CLM directives, a CLMIN directive cannot appear ahead of the DEVICE directive that identifies the device implied by the pathname argument (or the new_work_dir optional argument).

When a CLMIN directive is read by the CLM, the input file is immediately transferred to the beginning of the device/file indicated by the pathname. Subsequent CLM directives are read from the indicated device/file.

If a disk or magnetic tape volume is indicated by pathname, that volume unit must be mounted before system startup begins.

If the new working directory argument is used, the current working directory is changed to the indicated directory before a new user-in file is issued for the pathname argument. If the pathname argument is a simple name, it is relative to the new working directory.

Example 1:

```
CLMIN  ^ABCVOL>BETA1
```

In this example, subsequent CLM directives are read from a file named BETA1 on disk volume ABCVOL. An absolute disk pathname is used. Volume ABCVOL must be mounted on a disk device whose DEVICE directive has already been read by the CLM.

Example 2:

```
CLMIN  !CDR00
```

In this example, subsequent CLM directives are read from the card reader whose device_unit name is CDR00. (Assume that the DEVICE directive for CDR00 did not include a device_name argument; in this case, the device_unit argument value is used for File System references to the device.) The DEVICE directive for CDR00 must already have been read by the Configuration Load Manager.

Example 3:

```
CLMIN  CLM_USER,^VOL2>SID
```

In this example, the working directory is changed to ^VOL2>SID. Subsequent CLM directives are taken from ^VOL2>SID>CLM_USER. All files that are required by CLM must now reside in ^VOL2>SID. The DEVICE directive for ^VOL2 must already have been read by the Configuration Load Manager.

COMMENT (*)

The Comment directive is used to include a comment in the file of CLM directives.

FORMAT:

* comment_text

DESCRIPTION:

When an * (asterisk) is the first character of an entry in the file of CLM directives, the remainder of the entry is treated as a comment and is not processed by the CLM.

Comments can also be included on other CLM directives if they follow a blank after the last argument of the directive.

DEFFERED LOAD BOUND UNIT (DLDBU)

The DLDBU directive causes the indicated bound unit to be added to the end of the bound unit load list. The bound unit's root is then loaded into the system area of memory after all Load Bound Unit (LDBU) directives have been processed and its external symbols are placed in the System Symbol Table. (Since the root of the bound unit is loaded into the system area of memory, the SYS Linker directive must be specified when this bound unit is linked.) Once loaded, the root of the indicated bound unit is permanently resident in the system.

A bound unit loaded by means of the DLDBU directive may contain initialization subroutines. If so, this code is executed immediately after the root is loaded.

FORMAT:

```
DLDBU pathname[,argl,...,argn]
```

ARGUMENTS:

pathname

An ASCII string that identifies the File System pathname of the bound unit. If the pathname is preceded by two asterisks (**) and the bound unit identified by the pathname is not found, no error message is displayed, no halt occurs, and the configuration process continues uninterrupted.

[,argl,...,argn]

One or more arguments that are to be passed to the Initialization Subroutine Table (IST) that executes when the DLDBU directive is loaded during CLM processing.

NOTE

The first argument (argl) in the argument list must be separated from the pathname by a comma, as shown in the format for this directive.

DLDBU

DESCRIPTION:

Application-specific code (usually in the form of subroutines shared among multiple task groups) that is referred to symbolically during application execution can be permanently brought into memory at system startup by DLDBU directives. The pathname in each DLDBU directive is added to the end of the bound unit load list.

The order of DLDBU directives governs the order in which pathnames are added to the load list. If two DLDBU directives specify the same pathname, the second one is ignored. Note that Load Bound Unit (LDBU) directives and DLDBU directives can be interspersed. The deferred bound unit pathnames are separated for later processing.

After the bound units that are specified in LDBU directives have been loaded, the roots of the bound units identified by DLDBU directives are permanently loaded into the system area of memory. The symbol table for each such bound unit is added to the system's resident symbol table list. Once a bound unit has been loaded, any symbol defined therein by an EDEF Linker directive serves to resolve an unresolved reference to that same symbol in a subsequently loaded bound unit. Otherwise, if a bound unit contains an unresolved reference to a symbol, loading stops with an error halt.

Example:

```
DLDBU   **^ABCVOL>ALPHA1
```

In this example, a bound unit named ALPHA1, which is a file immediately subordinate to the root directory of a volume named ABCVOL, is added to the bound unit load list (if not already there). If the file is not found, no error message is displayed and the configuration process continues uninterrupted.

A bound unit loaded by means of the DLDBU directive may contain an Initialization Subroutine Table (IST). The IST defines one or more subroutines of the bound unit that are to be executed once only, immediately after the root is loaded. If a bound unit contains initialization subroutines, the Linker directive IST must be specified when the bound unit is linked. The label in the IST section is the argument to be entered in the IST directive. For a description of the IST section format, refer to the LDBU directive in this section.

DEVICE (DEVICE)

A DEVICE directive is required for each peripheral device in the system except for the bootstrap device and, if the BOOTCOMP directive is used, the companion to the bootstrap device. You can allow the system to dynamically assign channel numbers to appropriate devices on the bus. You must include a special format of the DEVICE directive in your CLM file if your installation uses a dual-purpose operator terminal. Refer to "Configuring a Dual-Purpose Operator Terminal" earlier in this section. *

FORMAT:

```
DEVICE device_unit,lrn,level,X'channel',[device_name],
                        {B}
      [record_size],[{N}]                                  *
```

ARGUMENTS:

device_unit

A string of five ASCII characters. The first three characters identify the type of device; the last two characters (alphanumeric) must be unique within the device type so as to identify one specific device of that type. Table 5-2 indicates the permissible values of device_unit, the device type indicated by each value, and the default physical record size (in bytes) for each device type.

Except where indicated, the default record size can be overridden by use of an explicit record_size argument in the DEVICE directive.

Table 5-2. Unit Value and Default Record Size for Various Devices

Device_Unit Value	Device Type Indicated	Default Physical Record Size in Bytes (Decimal)
* LPTnn	Line Printer	137
SPTnn	Serial Printer	133
CDRnn	Card Reader	80
CRPnn	Card Reader/Punch	80
DSKnn	Diskette	128
* MSMnn	Mass Storage Unit	256
RCMnn	Removable Cartridge Module Disk	256
FCMnn	Fixed Cartridge Module Disk	256
RLKnn	Removable Lark Disk	256
FLKnn	Fixed Lark Disk	256
MT7nn	Magnetic Tape (7-track)	0
MT9nn	Magnetic Tape (9-track)	0
MTCnn	M-80 Cassette Tape	
GENnn	Other Hardware Device	
WINnn	5 1/4 Winchester Disk	
STRnn	Streamer Tape	

* The device type GEN is a special one used to create a structure so that a nonstandard driver or LPH device can be configured to be accessed through the File System (without getting a 1380 error notice). The logical File System structure created is the same as that created for the TTY device type. If the characteristics are not satisfactory, they can be changed with the STTY directive.

Communication devices are described in Section 6.

lrn

The logical resource number by which the device is requested. The value for lrn is a decimal integer from 3 to 252 or 256 to 4002. lrn 0 is used for the operator terminal; lrn 1 is used for the bootstrap device; lrn 2 is used by the system; lrn 253 to 255 and 4003 to 4095 are reserved for system use.

level

The priority level used by the device driver for this device. The value for level is a decimal integer from 7 through 59.

X'channel'

The four-digit hexadecimal channel number of the device. The leftmost 10 bits specify the channel number; the rightmost six bits must be zero.

If you specify the channel number by a single zero, the system automatically assigns this device to an appropriate channel.

[device_name]

For disk and magnetic tape devices, the device_name argument must be omitted or specified as * (asterisk) because the device identity is established by the value of the device_unit argument. (The significance of using the asterisk is described below.)

For devices other than disk or magnetic tape, device_name can be one of the following:

1. A string of up to 12 ASCII characters that establish a unique name by which the device is referred to within the File System. The first character of the string must be alphabetic.

2. Omitted, in which case the value of the device_unit argument is used for File System references to the device.

3. An * (asterisk), which indicates a "private" device to be accessible only through physical input/output.

[record_size]

The length, in bytes, of one physical record. If the device is a line or serial printer, an extra character must be added for the slew character. If record size is not specified, the default record size is as established by the device_unit argument. (The default record size for disk devices and magnetic tapes cannot be modified by the record_size argument.)

$\begin{Bmatrix} B \\ N \end{Bmatrix}$

This argument is meaningful only for devices other than disk and magnetic tape.

* For normally unbuffered devices (for example, CDR, CRP), B indicates that input/output to the device is to be buffered.

For normally buffered devices (for example, LPT, SPT), N indicates that input/output to the device is to be unbuffered.

Input/output to a buffered device can be either asynchronous or synchronous; tabulation characters are expanded into space characters. Input/output to an unbuffered device is always synchronous; tabulation characters are not expanded.

NOTE

All arguments of a DEVICE directive must be unique (that is, not duplicated on another DEVICE directive) with these exceptions: (1) the B/N argument need not be unique, (2) for pairs of removable/fixed cartridge disks, the same level number and channel number must be specified for both platters, (3) multiple communications devices can be configured on the same level, and (4) multiple (polled) VIPs can be configured on the same channel. *

DESCRIPTION: *

Each peripheral device in the system, except for the bootstrap device, must be identified in a separate DEVICE directive. The companion to the bootstrap device can be optionally described using the BOOTCOMP directive instead. The device_unit argument (for example, DSK01) identifies both the type of device (DSK) and one specific device (01) of the indicated type. The lrn, level, and channel of each device must be unique (except in the case of a pair of removable/ fixed cartridge disks, in which case the level and channel for both platters must be the same but the lrn is different).

The first DEVICE directive for each device type causes the appropriate device driver to be loaded as part of the system. Each device driver is reentrant, so that only one copy of each required driver is loaded to service all devices of the same type.

A DEVICE directive for the bootstrap device is optional. If none is specified, the bootstrap device is assigned priority level 6 (the default). If you wish to override the default priority level, you should specify a DEVICE directive that indicates the appropriate device unit, lrn 1, desired priority level, and a channel number of zero (for example, DEVICE DSK00,1,level,X'0'). If the channel number is not zero, the following message is issued: 1345 CMD (DEVICE) SPECIFIES DUPLICATE DEVICE TYPE/UNIT.

Example 1:

```
DEVICE  LPT01,12,20,X'1380'
```

In this example, the line printer whose unit number is 01 is assigned lrn 12 and priority level 20. This device is connected to channel 1380. The following characteristics are established by default:

- The device name used within the File System is the same as the device unit (for example, LPT01).
- The record size is 137 bytes.
- Output to the device is buffered.

Example 2:

```
DEVICE  RCM01,9,10,X'1400'
DEVICE  FCM01,6,10,X'1400'
```

This example illustrates DEVICE directives for a pair of removable/fixed cartridge module disks. The logical resource number for each platter must be unique. The level and channel number for both platters must be the same.

IMMEDIATE LOAD BOUND UNIT (ILDBU)

The ILDBU directive causes the indicated bound unit to be loaded immediately. The bound unit's root is loaded into the system area of memory and its external symbols are placed in the System Symbol Table. (Since the root of the bound unit is loaded into the system area of memory, the SYS Linker directive must be specified when this bound unit is linked.) Once loaded, the root of the indicated bound unit is permanently resident in the system and the external symbols are visible to other bound units that are subsequently loaded.

A bound unit loaded by means of the ILDBU directive may contain initialization subroutines. If so, this code is executed immediately after the root is loaded.

FORMAT:

```
ILDBU pathname[,arg1,...,argn]
```

ARGUMENTS:

pathname

An ASCII string that identifies the File System pathname of the bound unit. If the pathname is preceded by two asterisks (**) and the bound unit identified by the pathname is not found, no error message is displayed, no halt occurs, and the configuration process continues uninterrupted.

[,arg1,...,argn]

One or more arguments that are to be passed to the Initialization Subroutine Table (IST) that executes when the ILDBU directive is loaded during CLM processing.

NOTE

The first argument (arg1) in the argument list must be separated from the pathname by a comma, as shown in the format for this directive.

DESCRIPTION:

Application-specific code (usually in the form of subroutines shared among multiple task groups) that is referred to symbolically during application execution can be permanently brought into memory at system configuration time by ILDBU directives.

NOTE

The device on which the bound unit file resides must have been previously configured with a suitable DEVICE directive.

As soon as the ILDBU directive is encountered, the roots of the identified bound units are permanently loaded into the system area of memory. The symbol table for each such bound unit is added to the system's resident symbol table list. Once a bound unit has been loaded, any symbol defined therein by an EDEF Linker directive serves to resolve an unresolved reference to that same symbol in a subsequently loaded bound unit. Otherwise, if a bound unit contains an unresolved reference to a symbol, loading stops with an error halt.

Example:

```
ILDBU  **^ABCVOL>ALPHA1
```

In this example, a bound unit named ALPHA1, which is a file immediately subordinate to the root directory of a volume named ABCVOL, is immediately loaded. If the file is not found, no error message is displayed and the configuration process continues uninterrupted.

A bound unit loaded by means of the ILDBU directive may contain an Initialization Subroutine Table (IST). The IST defines one or more subroutines of the bound unit that are to be executed once only, immediately after the root is loaded. If a bound unit contains initialization subroutines, the Linker directive IST must be specified when the bound unit is linked. The label in the IST section is the argument to be entered in the IST directive. For a description of the IST section format, refer to the LDBU directive in this section.

INSTALLATION IDENTITY (INSID)

The INSID directive is used to change the default installation ID.

FORMAT:

INSID 'installation id'

ARGUMENT:

'installation id'

A string of up to 30 ASCII characters, representing the installation's id. The single apostrophes are required if the first character is numeric or if a space character is embedded in the string. The apostrophes are not part of the id.

DESCRIPTION:

The INSID directive allows you to create your own installation id. If this directive is not used, the default is: DPS6 INSTALLATION.

For more information on the installation id, refer to the description of the $INSID monitor call in the System Programmer's Guide (Vol. II).

LOAD BOUND UNIT (LDBU)

* The LDBU directive causes the indicated bound unit to be added to the bound unit load list. The bound unit's root is then loaded into the Fixed System Area of memory after all CLM directives have been read and its external symbols are placed in the System Symbol Table. (Since the root of the bound unit is loaded into the Fixed System Area of memory, the SYS Linker directive must be specified when this bound unit is linked.) Once loaded, the root of the indicated bound unit is permanently resident in the system.

If you wish to incorporate an optional system capability into your configuration, such as error logging or defective memory trap handling, you might be required to load one or more bound units using LDBU directives.

A bound unit loaded by means of the LDBU directive may contain initialization subroutines. If so, this code is executed immediately after the root is loaded.

FORMAT:

```
LDBU pathname[,arg1,...,argn]
```

ARGUMENTS:

pathname

An ASCII string that identifies the File System pathname of the bound unit. If the pathname is preceded by two asterisks (**) and the bound unit identified by the pathname is not found, no error message is displayed, no halt occurs, and the configuration process continues uninterrupted.

[,arg1,...,argn]

One or more arguments that are to be passed to the Initialization Subroutine Table (IST) that executes when the LDBU directive is loaded during CLM processing.

NOTE

The first argument (arg1) in the argument list must be separated from the pathname by a comma, as shown in the format for this directive.

DESCRIPTION:

Application-specific code (usually in the form of subroutines shared among multiple task groups) that is referred to symbolically during application execution can be permanently brought into memory at system startup by LDBU directives. The pathname in each LDBU directive is added to a bound unit load list.

The order of LDBU directives governs the order in which pathnames are added to the list. If two LDBU directives specify the same pathname, the second one is ignored.

After the QUIT directive (the last CLM directive) is read by the CLM, the roots of the bound units identified by LDBU directives are permanently loaded into the Fixed System Area of memory. The symbol table for each such bound unit is added to the system's resident symbol table list. Once a bound unit has been loaded, any symbol defined therein by an EDEF Linker directive serves to resolve an unresolved reference to that same symbol in a subsequently loaded bound unit. Otherwise, if a bound unit contains an unresolved reference to a symbol, loading stops with an error halt.

Example:

```
LDBU  **^ABCVOL>ALPHA1
```

In this example, a bound unit named ALPHA1, which is a file immediately subordinate to the root directory of a volume named ABCVOL, is added to the bound unit load list (if not already there). If the file is not found, no error message is displayed and the configuration process continues uninterrupted.

A bound unit loaded by means of the LDBU directive may contain an Initialization Subroutine Table (IST). The IST defines one or more subroutines of the bound unit that are to be executed once only, immediately after the root is loaded. If a bound unit contains initialization subroutines, the Linker directive IST must be specified when the bound unit is linked. The label in the IST section is the argument to be entered in the IST directive. The final displacement of the IST label in the bound unit must be less than 32K.

The format of an IST section is given below:

```
label   DC     0*
        RESV   $AF,0     RFU

        DC     <subl
        DC     0         Arguments to subroutine subl (may
        DC     0         be any 16-bit value)

        DC     <sub2
        DC     0         Arguments to subroutine sub2 (may
        DC     0         be any 16-bit value)
        .
        .
        .
        DC     <subn
        DC     0         Arguments to subroutine subn (may
        DC     0         be any 16-bit value)

        RESV   $AF,0     End of table sentinel

subl    ........
        ........
        ........

sub2    ........
        ........
        ........

subn    ........
        ........
        ........
```

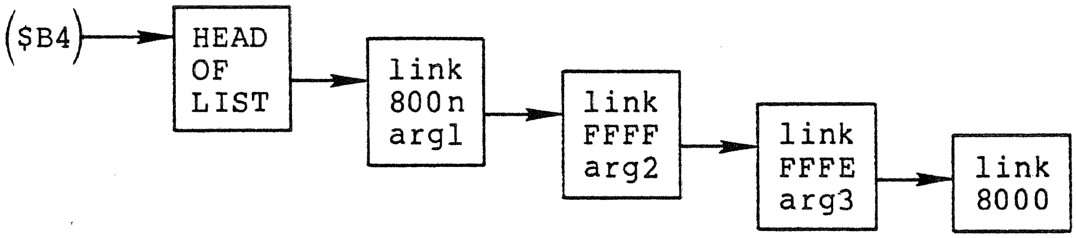
Upon entrance to a subroutine, register B5 contains the return address. Before exit from the subroutine, register R1 must contain the status. A value of zero indicates correct execution. A nonzero value is the error code for this subroutine. If the status is nonzero, the following message is written to the error-out file, and a halt occurs. Execution of CLM cannot continue.

134B INITIALIZATION SUBROUTINE ERROR

(error code returned to register R1)

*If the entry is 0, all memory used by the initialization subroutines is returned to the system after the subroutines have been executed (provided the subroutines have not caused the system to create data structures on their behalf). To retain a subroutine in memory, specify n, where n is the number of words to be retained. In this case, take care not to extend the memory requirement beyond that allocated to the bound unit's root when it was loaded.

The arguments following the pathname in the LDBU directive are passed to the IST code in the following manner: upon entrance to the first subroutine in the IST code, $B4 points to the head of the argument list. The list has the following structure:



where:

800n = argument is an ASCII string of n bytes
FFFF = argument is a 1-word (16-bit) value
FFFE = argument is a 2-word (32-bit) value
8000 = argument is null

The link of the last argument is 0.

MAP (MAP)

The MAP directive provides configuration information that enables you to verify the success of certain aspects of the system building process. The information presented by this directive includes the system physical attributes, system structures, device status, user-defined variables, and memory pool specifications. This information can be used to ease the debugging of system extensions and user-written device handlers. In addition, information (such as system symbols) is presented that would otherwise be unavailable.

FORMAT:

```
MAP [path][,map_form_1][,map_form_2]...
```

ARGUMENT:

[path]

Pathname of the file to which the map listing is to be written. The pathname cannot be that of a communication device. The pathname must be able to be verified when the CLM reads the MAP directive.

If the pathname refers to a device file (for example, !LPT00), that device must have been configured when the MAP directive is read.

If the pathname refers to a disk file, the indicated volume must be mounted on the drive of a device that is already configured and recognized by the File System when the directive is read. If the pathname refers to a disk file that does not exist, the file is created (as long as the directory structure identified in the pathname already exists). If the file already exists, it must be a variable sequential file. In this case, it is opened in renew mode.

The default map path value is >>SID>CLM_MAP.

[map_form_1]...[map_form_n]

Any combination of the map format arguments described below can be entered in any order. If none of these arguments is specified, the entire system map is written to the map file. The complete map presents the following information (in the order shown):

- System physical attributes
- System data structures
- User-defined variables *
- Memory pool attributes
- System extensions
- User-selected permanently resident system overlays and system overlays made resident to serve as device drivers
- System symbol table.

$\left\{\begin{array}{l}\text{-BRIEF}\\\text{-BF}\end{array}\right\}$

Requests display of only the physical attributes, system structures, user-written driver attributes, and user-selected permanently resident overlays.

$\left\{\begin{array}{l}\text{-NDEV}\\\text{-ND}\end{array}\right\}$

Omits from the map the data pertaining to device status (devices accessed through the File System).

$\left\{\begin{array}{l}\text{-NVAR}\\\text{-NV}\end{array}\right\}$

Omits from the map the data pertaining to user-defined variables.

$\left\{\begin{array}{l}\text{-NPOOL}\\\text{-NP}\end{array}\right\}$

Omits from the map the data pertaining to memory pools.

```
{-NRES}
{-NR  }
```

Omits from the map the data pertaining to user-selected permanently resident system overlays and system overlays made resident to serve as device drivers.

```
{-NSYM}
{-NS  }
```

Omits from the map the data pertaining to the system symbol table.

Example 1:

```
MAP
```

The entire map is written to the map file whose pathname is >>SID>CLM_MAP.

Example 2:

```
MAP !LPT00,-NSYM
```

The entire map, except for the data pertaining to the system symbol table, is produced and written to line printer !LPT00. The output in this example is as follows:

```
CLM MAP

PHYSICAL ATTRIBUTES:

     GCOS6 MOD400-R4.0-12/05/1056
     SIP PRESENT
     CIP PRESENT
     MMU PRESENT
     PHYSICAL MEMORY SIZE (HEX): 40000
     TIME BETWEEN REALTIME CLOCK INTERRUPTS:   32
     OPERATOR CONSOLE CONFIGURED
     ERROR MESSAGE LIBRARY IN USE

SYSTEM STRUCTURES: (DECIMAL NUMERIC VALUES)

     120  IRBS,  106  TSAS,  10   SYSTEM OVERLAY AREAS,
     15   RECORD LOCKS CONFIGURED
     0    NUMBER OF SYSTEM IORBS
     5510 SIZE OF SYSTEM IORB (WORDS)
MESSAGE LIBRARY PATHNAME: >>ML>MLFILE.EN
```

```
USER-WRITTEN DRIVERS:


USER-DEFINED CLM VARIABLES:


MEMORY POOLS:

     $S   START=15400   END=3EFFF    SIZE= 29C00   PHYS. START = 15400
          ATTRIBUTES= P         USER'S RING NUMBER= 0
     AB   START=15400   END=3EFFF    SIZE= 29C00   PHYS. START = 15400
          ATTRIBUTES= EP        USER'S RING NUMBER= 0
     V0   START=15400   END=3EFFF    SIZE= 29C00   PHYS. START = 15400
          ATTRIBUTES= EP        USER'S RING NUMBER= 0
     V1   START=15400   END=3EFFF    SIZE= 29C00   PHYS. START = 15400
          ATTRIBUTES= EP        USER'S RING NUMBER= 0
     V2   START=15400   END=3EFFF    SIZE= 29C00   PHYS. START = 15400
          ATTRIBUTES= EP        USER'S RING NUMBER= 0
     V3   START=15400   END=3EFFF    SIZE= 29C00   PHYS. START = 15400
          ATTRIBUTES= EP        USER'S RING NUMBER= 0
     LL   START=40000   END=40FFF    SIZE= 1000    PHYS. START = 3F000
          ATTRIBUTES= EPU       USER'S RING NUMBER= 2

LDBU'S: (SYSTEM EXTENSIONS)

     ZXEDEF        START= 5D47
     ZQTDUM        START= 9F7E
     ZQEXEC        START= BD92
     ZUCSF         START= DD32
     ZNVDAM        START= E200
     ZQPAT6        START= 10941
     ZQPAT1        START= 1196D
     ZQPAT5        START= 11E5B
     ZQPAT4        START= 13199
     ZXDEFM        START= 134E8
     ZGQCDS        START= 13707
     ZNV72F        START= 1375C
     ZNV72C        START= 15036


RESOLA'S: (PERMANENTLY RESIDENT EXECUTIVE OVERLAYS)



     OIOCR    START=9C42    OIODK    START=11C7    OIOLP    START=9D99
     OIOTPE   START=9778    OIX      START=8E24
```

MEMPOOL (MEMPOOL)

* A MEMPOOL directive defines a system memory pool or an independent memory pool. To define a swap pool, refer to the description of the SWAPPOOL directive. Each MEMPOOL directive * describes only one type of memory pool. (You can specify the size of a memory pool symbolically using a VARIABLE directive. Refer to the description of the VARIABLE directive in this section.)

You can perform a default memory allocation by not including any MEMPOOL or SWAPPOOL directives in your CLM_USER file. If no such directives are used, a system pool named $$ of approximately 120,000 bytes is automatically configured. In addition, a *-size swap pool named AB is configured. In the Extended Memory Management (EMMU) environment, the swap pool is allocated all of the remaining memory (less the system pool), up to a total of 16 million bytes. In the Memory Management Unit (MMU) environment, the swap pool is allocated all of the remaining memory (less the system pool), up to a total of 2 million bytes.

FORMAT 1 (System Pool):

```
MEMPOOL S,,size[,S]
```

* FORMAT 2 (Independent Pool):

MEMPOOL I,pool_name,size[,[S] $\left[\begin{Bmatrix} U \\ NU \end{Bmatrix}\right]$]

ARGUMENTS - FORMAT 1 (System Pool):

S

When used as the first argument, the letter S designates the protected system memory pool. The system memory pool must be defined in a separate MEMPOOL directive. If multiple MEMPOOL directives have S as the first argument, only the last one is honored.

size

The size argument is a positive integer that defines the number of words requested for the memory pool. To specify a pool size greater than 65,535 words, use the double-word hexadecimal integer format (for example, D'10A00').

The size of the system pool can also be specified by an asterisk (*), in which case CLM allocates to the system pool all memory remaining after MEMPOOL directives specifying explicit sizes have been processed.

In the EMMU environment there can be only one *-size pool. This pool uses all of the remaining memory (up to 8M words). Any other *-size pool of the same type as the first is renamed to the first one. Any other *-size pool of a different type causes an error.

In the MMU environment, the number of pools with size * depends on the size of memory and the virtual views. A virtual view consists of one of the following combinations of memory pools:

- The system pool, system area, and an independent (I-type) pool

- The system pool, system area, and a swap pool.

The general rule for pools of size * in the MMU environment is: one virtual view per one million (1024 times 1024) words of memory can contain a pool with size *. If the system pool has size *, no other pool can have size * regardless of total memory size. If memory contains one million words or less, there can be only one pool of size *.

If a system memory pool is not defined in an explicit MEMPOOL directive, its size is set at approximately 60,000 words.

The size of pools must be in multiples of a certain number of words and is rounded to the next highest multiple by CLM. For pools that are neither protected nor contained, the size of the multiple is 32 words. For pools that are protected or contained, the size of the multiple is 256 words. (Memory pool protection and/or containment requires that the DPS 6 central processor possess either an MMU or an EMMU.)

[S]

Serial usage of the pool; that is, only one group at a time can use it. An attempt to create a second group using a serial usage pool results in an error return and the second group is not created. If this argument is omitted, the pool can be used by more than one group at a time.

* ARGUMENTS - FORMAT 2 (Independent Pool):

I

This pool is to be an independent pool; such a pool has a view of only itself and the system. It has the Protect Contained (PC) attribute automatically.

pool_name

The two-character ASCII name that uniquely identifies each independent memory pool. The pool name is used in Create Group (CG) and Spawn Group (SG) commands and in $CRGRP and $SPGRP macrocalls.

If the pool name begins with a decimal digit, the pool name argument must be surrounded by apostrophes (for example, '1A').

NOTE

At login you can (1) specify the two-character identification of the group id and memory pool, (2) specify a memory pool identification which is different from the group id (i.e., the -POOL argument), or (3) omit the pool identification and accept a two-character identification assigned by Listener.

If you specify allocation pools with the Pn argument when creating or spawning the Listener group, you must allocate those pools in the CLM_USER file with either the MEMPOOL or SWAPPOOL directive. If you do not, Listener uses L0, which must have been configured.

size

The description of the size argument for format 1 also applies to format 2.

In the MMU environment the system allows multiple *-sized I pools to be assigned, even when their number exceeds the amount of physical memory. Any *-sized I pool that is oversubscribed (i.e., for which there is insufficient or no physical memory) is renamed to the first *-sized I pool that has been successfully allocated. If no *-sized I pool has been successfully allocated, a 1343 error occurs.

[S]

Serial usage of the pool; that is, only one group at a time can use it. An attempt to create a second group using a serial usage pool results in an error return and the second group is not created. If this argument is omitted, the pool can be used by more than one group at a time.

A memory pool that is known by several names (through explicit or implicit renames) should not be designated serial-usage since more than one task group may want to use the memory pool via different pool name assignments.

[U]

Groups using this pool run in the unprivileged mode. Groups running in the unprivileged mode cannot execute privileged instructions and trap if a privileged instruction is attempted. The default is unprivileged.

[NU]

NU specifies that the pool is "not unprivileged"; that is, the pool is privileged.

DESCRIPTION:

Each MEMPOOL directive causes a pool descriptor list to be created. Later, after the memory requirements of the system and its extensions are known (including the impact of any SWAPPOOL directives), each pool set is checked to ascertain whether it can fit into the remaining available space. If any nonoverlapping element or pool set is too large, an error occurs.

The system pool is protected. The swap pool and all independent pools are implicitly protected and contained.

Example 1: *

```
MEMPOOL S,,10000
```

In this example, a system memory pool of 10,000 words is defined. Because a system memory pool is being defined, no pool name argument is specified, but the second comma indicates omission of the argument. By default, the system pool is protected.

* Example 2:

```
MEMPOOL I,IN,D'20000'
```

Pool IN is an independent pool with a size of 128K words. Independent pools are automatically protected and contained.

* Example 3:

```
MEMPOOL I,AB,*
    .
    .
    .
MEMPOOL I,XY,*
```

In the MMU environment these directives assign multiple *-sized I-pools. So long as one I-pool is successfully allocated, the succeeding I-pools that are oversubscribed are renamed to like-allocated *-sized pools in rotation in the order the directives were issued. In the EMMU environment there can be only one *-size pool.

If both I-pools and swap pools are to be oversubscribed, it is wise to issue an I-pool directive followed by a swap pool directive, rather than grouping them together, so as to ensure the successful assignment of the first directive of each type. In the above example, if the first *-sized swap pool directive were assigned after I-pool XY, no swap pool would be assigned and a 1343 error would occur. If, however, the swap pool directive were issued after I-pool AB, then that swap pool would be allocated and subsequent oversubscribed swap pools would be renamed to this one.

I-pool AB can occupy all memory remaining after the pools with explicit sizes have been allocated. In the MMU environment the pool size does not exceed 1024K words minus the size of the system pool and system area. In the EMMU environment the pool size does not exceed 8M words minus the size of the system pool and system area.

MESSAGE LIBRARY PATHNAME (MLPATH)

Specify that a message library other than >>ML>MLFILE.EN is to be used.

FORMAT:

MLPATH name.language_key

ARGUMENT:

name.language_key

name.language_key is the pathname of the desired system global message library. language_key is a two-character system global (national) language key.

DESCRIPTION:

The MLPATH directive allows you to specify a system global message library other than the default one (>>ML>MLFILE.EN). If you do not specify a language key, you receive an error notice (136B).

For information on the message library, refer to the System Messages manual.

QUIT (QUIT)

The QUIT directive must be the last directive in the file of CLM directives.

FORMAT:

*

QUIT

DESCRIPTION:

The QUIT directive causes the CLM to cease reading CLM directives and to begin its loading phase. The following actions occur:

1. Any required final data structures are created.
2. If a communication environment has been defined, it is initialized.
3. Bound units (if any) identified in LDBU and DLDBU directives are loaded into the system area of memory.
4. System overlays (if any) identified in RESOLA directives are loaded into the system area of memory.
5. Memory pool descriptors are created, based on the contents of MEMPOOL directives. Each memory pool set is checked to ascertain whether it can fit into available memory space.
6. The CLM terminates.

Example:

* QUIT

This example causes the Configuration Load Manager to cease
* reading CLM directives and to enter its loading phase.

REBOOT (REBOOT)

The REBOOT directive is used to configure the Software Reboot facility.

FORMAT:

REBOOT [path]

ARGUMENT:

path

The pathname of the file to be used to contain a memory dump taken automatically prior to each software-initiated reboot. The default is for no dump file to be configured and no automatic memory dump to be taken.

DESCRIPTION:

The Reboot facility is configured to enable software-initiated system reinitialization. This facility can later be called from the RBOOT operator command, by internal monitor calls in an application, as a result of a level 2 "halt" condition or from power-fail recovery.

If you specify the path argument (automatic memory dumping), the file must be large enough to contain all installed memory and must be allocated as a single continuous extent. The file can reside on any supported disk medium that contains enough space. If the file is not found, it is immediately created with the correct size for the amount of memory on your system. Sufficient continuous disk space must be available, otherwise CLM halts. The dump file is opened when the system is configured; the volume on which it resides must be mounted at all times. If no pathname is specified, the system is configured without a dump file and automatic dumps are not taken.

NOTE

When the Software Reboot facility is configured, the CLM_USER file is held open after system initialization, and cannot be directly modified while the system is running. To create a new CLM_USER file, you must first use the Rename (RN) command with the -FORCE argument on the old CLM_USER file. Otherwise, you can use the Modify Reboot Parameters (RBPRM) command to dynamically change the current system reboot parameters.

RESIDENT OVERLAY (RESOLA)

The RESOLA directive identifies one or more system overlays that are to be loaded into the Fixed System Area of memory, where they remain resident for the duration of the configured system.

FORMAT:

```
RESOLA overlay_name_1[,overlay_name_2]...
```

ARGUMENT:

overlay_name_i

A string of up to six ASCII characters that identify a system overlay that is to be made resident for the duration of the configured system. The Software Release Bulletin identifies the system overlays that can be specified in this argument and the function(s) provided by each one.

DESCRIPTION:

Each RESOLA directive adds one or more system overlay names to a list of resident system overlays. During the CLM loading phase, each name on the list is compared with the system's table of system overlay names. A match causes the indicated system overlay to be loaded into the Fixed System Area of memory. Making frequently used overlays resident in memory can improve system performance.

If no match occurs, an error is issued. If one overlay name on a RESOLA directive causes an error, all other correctly specified overlays in the same directive can nevertheless be loaded.

The Honeywell-supplied system includes a file named RESOLA in the directory >>SID. >>SID>RESOLA contains RESOLA directives for each of the system overlays. For information about the function of each overlay, refer to the Software Release Bulletin.

You can edit >>SID>RESOLA to suit your installation's needs by removing or commenting out (i.e., inserting * as the first character of a line) any RESOLA directives that specify overlays that you do not wish to make resident. If you modify >>SID>RESOLA, write the edited file to a file with a new pathname.

To use the edited version of >>SID>RESOLA, include the CLM directive,

```
CLMIN pathname
```

(where pathname is the name of the edited file) in your CLM_USER file just before the QUIT directive.

If you want all system overlays resident, include the

```
CLMIN >>SID>RESOLA
```

directive in your CLM_USER file just before the QUIT directive. In this case, you should ensure that >>SID>RESOLA has the correct contents (that is, refers only to those overlays that you wish to have resident).

Example:

```
RESOLA ZZBEF,ZZBIWR,ZZCLPT
```

This example identifies three system overlays (these three are used for processing using record locking/recovery) that are loaded into the system area of memory, where they remain resident for the duration of the configured system. Note that the overlay names must be separated by commas (,) not blanks.

RECORD LOCK (RLOCK)

The RLOCK directive defines a pool of structures to be used by data management when record locking is requested at file reservation time. The size of the pool determines the total number of records that can be locked at one time.

If the RLOCK directive is not specified, then the following system defaults are used:

- Number of initial record lock structures: 37
- Number of record lock structures that can be added at one time: 23
- Total number of record lock structures permitted: unlimited.

One structure is used for each record to be locked. The size of each structure is 10 words. These structures are reusable after records become unlocked. If more than one RLOCK directive is given, the last one takes effect.

FORMAT:

```
RLOCK [init],[inc],[max]
```

ARGUMENTS:

[init]

The number of initial record lock structures. The default is 15.

[inc]

The number of record lock structures that can be added at one time. The default is 5.

[max]

The total number of record lock structures permitted. The default is 30. An entry of 0 signifies that the total number of lock structures is unlimited. If the initial size or the increment size is greater than the maximum size, the following CLM message is generated:

133A RLOCK ARGUMENTS INCONSISTENT WITH EACH OTHER

NOTES

1. The initial record lock pool is allocated in system memory at the time of the first record lock request.

2. The sizes of all three arguments are rounded up, so that the amount of memory allocated is a multiple of 32 words minus 3.

DESCRIPTION:

The Record Lock facility of the File System provides multiuser interference protection for records within shared disk files. For a detailed description of this facility see the Data File Organizations and Formats manual. Because record lock entries are used and surrendered dynamically, the RLOCK pool need contain only enough entries to service the maximum number of records that are locked at the same time. The following procedure shows how to calculate the size of the record lock pool for a system where a number of users perform simultaneous transactions.

Assume that each transaction consists of an update of 10 records. The File System locks the Control Intervals (CIs) in which the 10 records reside.

Assume that 20 users perform transactions but never more than four users access the data base simultaneously. For these assumptions, the initial number of record locks is 40 (4 x 10) and the natural increment is 10.

Example: RLOCK 40,10,200

```
init =  40
 inc =  10
 max = 200
```

SWAPPOOL

SWAPPOOL (SWAPPOOL)

Define a swap pool. You can perform a default memory allocation by not including any MEMPOOL or SWAPPOOL directives in your CLM_USER file. If no such directives are used, a system pool named $$ of approximately 120,000 bytes is automatically configured. In addition, a *-size swap pool named AB is configured. In the Extended Memory Management (EMMU) environment, the swap pool is allocated all of the remaining memory (less the system pool), up to a total of 16 million bytes. In the Memory Management Unit (MMU) environment, the swap pool is allocated all of the remaining memory (less the system pool), up to a total of 2 million bytes.

FORMAT:

```
SWAPPOOL [F] ,pool_name,pool_size,swapfile
         [M]
```

ARGUMENTS:

F

The letter F indicates that this pool is the first pool after the system pool. This swap pool has the special characteristic that the system pool can extend into it (see the SYS2 directive). This swap pool is physically contiguous to the system pool. There can be only one F-type swap pool.

M

The letter M indicates that this is the second or later pool assigned after the initial swap pool, designated F. Systems with an EMMU cannot use M type swap pools.

pool_name

The two-character ASCII name of the swap pool. The name of the swap pool must be unique among the set of all pool names, including all pool names specified in MEMPOOL and other SWAPPOOL directives.

pool_size

A decimal number less than 65K or a hexadecimal number specifying the size of the swap pool in words. Hexadecimal sizes less than 65K are specified in the X'dddd' format; those less than or equal to 536K are specified in the D'dddddddd' format.

The pool size can be specified as *. See the size discussions of the MEMPOOL directive for more information about *-sized pools.

In the MMU environment the system allows multiple *-sized M-type swap pools to be assigned, even when their number exceeds the amount of physical memory. Any *-sized M-type swap pool that is oversubscribed (i.e., for which there is insufficient or no physical memory) is renamed to the first *-sized M-type swap pool that has been successfully allocated. If no *-sized M-type swap pool has been successfully allocated, a 1343 error occurs.

swapfile

The pathname of the swap file. The swap file is used to store segments that are swapped out. If swapfile is not a full pathname, it is assumed to be relative to >>SID or to the current working directory (if changed via CLMIN).

The default swap file pathname is >>SID>SWAPFILE.nn, where nn is the name of the swap pool.

DESCRIPTION:

An implicit LDBU is issued for the bound units SEGMAN and SWAP if you include a SWAPPOOL directive with an F-type swap pool.

There can be only one set of arguments for the SWAPPOOL directive.

Example 1:

```
SWAPPOOL F,QQ,50000
```

QQ is a swap pool and is the first pool after the system pool. QQ has a size of 50,000 words; it is automatically protected and contained. The swap file for QQ has the default pathname >>SID>SWAPFILE.QQ.

Example 2:

```
MEMPOOL S,,60000
SWAPPOOL F,AB,*
SWAPPOOL M,M1,*
SWAPPOOL M,M2,*
SWAPPOOL M,M3,*
SWAPPOOL M,M4,*
SWAPPOOL M,M5,*
SWAPPOOL M,M6,*
SWAPPOOL M,M7,*
SWAPPOOL M,M8,*
SWAPPOOL M,M9,*
SWAPPOOL M,MA,*
SWAPPOOL M,MB,*
SWAPPOOL M,MC,*
```

In the MMU environment these directives assign multiple *-sized swap pools. So long as one swap pool is successfully allocated, the succeeding pools that are oversubscribed are renamed to like-allocated *-sized pools in rotation in the order the directives were issued. In the EMMU environment there can be only one *-size pool.

Although these SWAPPOOL directives result in oversubscription (i.e., the assignment of more *-sized M-type swap pools than there is physical memory), they successfully configure a system of less than one million words up through eight million words. For the one-million-word system, all M-type swap pools are oversubscribed. For the eight-million-word system, swap pools MA, MB, and MC are oversubscribed. For either system, oversubscribed pools are renamed to successfully allocated *-sized swap pools, starting with the first one. If no *-sized M-type swap pool has been allocated, an oversubscribed M-type swap pool is renamed to any size F-type swap pool that has been successfully allocated. If none has, a 1343 error occurs.

If both I-pools and swap pools are to be oversubscribed, refer to the example of oversubscribed I-pools in the MEMPOOL directive for information on mixing I-pools and swap pools.

Swap pools can occupy all memory remaining after the pools with explicit sizes have been allocated. In the MMU environment the pool size does not exceed 1024K words minus the size of the system pool and Executive. In the EMMU environment the pool size does not exceed 8M words minus the size of the system pool and system area.

SYSTEM (SYS)

The SYS directive defines a number of system variables.

FORMAT:

$$\text{SYS [Hz],[scan\_cycle],} \left[\begin{Bmatrix}\text{SSIP}\\ \text{DSIP}\\ \text{CIP}\\ \text{SCIP}\end{Bmatrix}\right] \text{,[olan],[tsa],[irb][,,E]}$$

ARGUMENTS:

Hz

The line frequency at which the system's real-time clock operates. Possible values are 60 (for 60 Hz) and 50 (for 50 Hz). On current DPS 6 models the Hz argument should be null and the default value (60) is used by the operating system.

[scan_cycle]

The time, in milliseconds, between interrupts by the system's real-time clock. The default value is 50 (milliseconds).

The following values (in milliseconds) are possible for scan_cycle.

50 Hz	60 Hz
10	8
20	16
50	25
100	33
	50
	100

Millisecond intervals in cyclic clock request blocks should be expressed (if possible) in multiples of the scan-cycle. This practice ensures long-term accuracy of event synchronization. Otherwise, the full deviation between the specified interval and the nearest scan-cycle multiple is always realized. *

SYS

$$\left[\begin{Bmatrix}\text{SSIP}\\ \text{DSIP}\\ \text{CIP}\\ \text{SCIP}\end{Bmatrix}\right]$$

Scientific instructions and/or commercial instructions are used in applications. On all current DPS 6 models, commercial instructions are processed by the Commercial Instruction Processor (CIP). For compatibility with earlier models, a software simulator, CIPSIM, is supported. Scientific instructions are processed by the Scientific Instruction Processor (SIP) if it is present; otherwise, scientific instructions are processed by the software simulator (SIPSIM).

The commercial extension to the interrupt save area is automatically created for each priority level except for those levels occupied by drivers, regardless of the option specified for this argument. The presence of a hardware SIP automatically creates the commercial and scientific extensions to the interrupt save area for each priority level except for those levels occupied by drivers, regardless of the option specified for this argument.

SSIP indicates that single-precision scientific instructions are to be processed. If the hardware SIP is not present, the single precision SIP simulator (SIPSIM_SP) is automatically loaded during the system startup. If the hardware SIP is present, it is used and the single-precision SIP simulator is not loaded. (SSIP is permissible for use with FORTRAN programs without double precision.)

DSIP indicates that double-precision scientific instructions are to be processed. If the hardware SIP is not present, the double-precision SIP simulator (SIPSIM) is automatically loaded during system startup. If the hardware SIP is present, it is used and the double- precision SIP simulator is not loaded.

CIP indicates that commercial instructions are to be processed. If the required central processor model is not present, you must include an LDBU directive that identifies the commercial simulator (CIPSIM). If the hardware CIP is present, it is used and the commercial simulator is not loaded.

SCIP indicates that both scientific and commercial instructions are to be processed. If the hardware SIP is not present, you must include an LDBU directive that identifies the single-precision SIP simulator (SIPSIM_SP) or the double-precision SIP simulator (SIPSIM). If a commercial central processor model is not present, you must include an LDBU directive that identifies the commercial simulator (CIPSIM). If the hardware SIP and a commercial central processor are present, they are used, and the SIP and commercial simulators should not be identified in LDBU directives.

NOTE

If both SIP and commercial simulators are used, the order of the LDBU directives governs the order in which the simulators process a trap to the single trap vector they share. The simulator identified in the last LDBU directive processes the trap first.

null (no entry)

No entry indicates that no scientific or commercial instructions are to be processed or that any required hardware SIP and/or a commercial central processor model is present and are used. (Thus the corresponding simulator is not used.)

[olan]

The number of 512-word system overlay areas to be created. The value for olan must be a decimal integer from 2 through 99. The default value is 1, indicating one 512-word area to be used for system overlays.

[tsa]

The number of trap save areas to be created in addition to the default number of 6 (system without communications), or 7 (system with communications).

The size of each additional trap save area is the same as the size of each of the default trap save areas (104 words).

You have the option of expressing this argument symbolically by defining it with a VARIABLE directive. (See the description of the VARIABLE directive later in this section.)

[irb]

The number of intermediate request blocks to be created in addition to the default number of 20 for the system. Each intermediate request block is 13 words.

You have the option of expressing this argument symbolically by defining it with a VARIABLE directive. (See the description of the VARIABLE directive.)

[E]

Expanded error messages (text in addition to code) are to be issued. If this argument is omitted, only the error code is issued.

Once this argument is specified, it remains in effect for the duration of the configured system, even if subsequent SYS directives appear with this argument null.

NOTE

Two commas must precede this argument as shown in the format of the directive and in the example below.

DESCRIPTION:

The SYS directive defines a number of system variables. If all of the SYS directive's default values are acceptable, the variable arguments can be omitted. These default values are summarized below.

Hz - 60 Hz line frequency

scan_cycle - 50 milliseconds

SSIP/DSIP/CIP/SCIP - null (no SIP simulator or commercial simulator required)

olan - 1 system overlay area

tsa - 7 trap save areas

irb - 20 intermediate request blocks

E - error code only, no text.

If multiple SYS directives are specified, only the last one is effective. Exceptions are: (1) all trap save areas and intermediate request blocks specified on all SYS directives are added to the system defaults of 7 and 20, respectively, and (2) if E is specified in any SYS directive, expanded error messages (text in addition to code) are issued.

Example:

```
SYS ,25,SSIP,5,,,,E
```

This example defines the system variables as shown below. (Assume that only one SYS directive is used.)

- The system line frequency is 60 Hz (default).
- The scan cycle of the system's real-time clock is 25 milliseconds.
- The single-precision SIP simulator (SIPSIM_SP) is loaded to process scientific instructions, unless the hardware SIP is present.
- Five system overlay areas are created.
- Six trap save areas are available (default if no communications).
- Twenty intermediate request blocks are available (default).
- Error codes are expanded to include message text.

SYSTEM2 (SYS2)

Define several system variables in addition to those defined using SYS.

FORMAT:

```
SYS2 [R],[silent_log_name],,,[sys_pool_max_growth],
     [growth_incr][,sys_pool_threshold]
```

ARGUMENTS:

[R]

Allows optional queuing of requests (e.g., I/O and task requests) by the priority level of the issuing task. The default is no priority queuing.

[silent_log_name]

A 1- to 12-character name that serves as a pseudodevice name to provide the capability of silent writes to the OIM log through File System or standard I/O calls.

[sys_pool_max_growth]

The maximum number of words by which the system pool can expand into the first swap pool. The value is rounded up to the nearest 256 words.

If this value is larger than the size of the swap pool, then the size of the swap pool becomes the maximum growth size.

If you specify 0 for this value, the system pool cannot expand into the swap pool.

If you omit this argument, the system pool expands to the limit of its last large segment, unless this expansion exceeds the size of the swap pool. In the latter case, the size of the swap pool is the maximum growth size.

If the value specified for this argument results in extending into additional large segments, then those segments are reserved for system pool expansion and are not available in any user's virtual view.

[growth_incr]

The number of words by which the system pool can be expanded at one time. A zero or null value means expand by 256 words.

If this value is not a multiple of 256, it is rounded up to the nearest multiple of 256.

[sys_pool_threshold]

This argument represents the percent of the system pool that must be used before it is expanded.

A zero or null value for this argument means 100 percent, and expansion only occurs when a request for system memory cannot be satisfied.

Example:

```
SYS2 R,SILOG,,,D'10000',4096,85
```

Silent logging is enabled, and queuing of all requests by the priority level of the requesting task is requested. The silent log name is SILOG. This name can be referenced later (for example, in an FO command). The maximum growth of the system pool into the swap pool is 64K words, with an increment of 4K words. When 85 percent of the system pool area is used, it can expand into the swap pool.

VARIABLE (VARIABLE)

The VARIABLE directive defines a single-character, symbolic expression (chosen from the letters A through Z) that can be specified in certain CLM directive statements.

Variables can define values for the fields in the SYS directive representing additional trap save areas or intermediate request blocks, or memory pool sizes in a MEMPOOL directive. A variable must be defined by a VARIABLE directive before it can be specified in other CLM directives.

FORMAT:

```
VARIABLE variable=expression
```

ARGUMENTS:

variable

A single character designation, uppercase only, chosen from the letters A through Z.

expression

Any valid expression that defines the value of the variable. Expression can equal any of the following:

- An integer having a value between -32767 and +32767
- CON (represents the total number of MDC-connected terminals finally configured with the CON value) *
- TTY (represents the total number of communication devices finally configured by TTY* directives)
- ATD (represents the total number of devices finally configured by DEVICE ATD directives)
- PVE (represents the total number of communication devices finally configured by PVE directives)
- STD (represents the total number of communication devices finally configured by STD directives)
- ASP (represents the total number of communication devices finally configured by ASP directives).

Expression can take any of the following forms:

- expression $\left\{\begin{matrix} + \\ - \\ * \\ / \end{matrix}\right\}$ expression

  An arithmetic combination of two expressions (chosen from this list) using addition (+), subtraction (-), multiplication (*) or division (/). The value of an expression is always an integer. All fractional values are truncated, not rounded. For example, 5/3 is passed to the system as having a value of one (1).

- (expression1)

  expression1 is to be evaluated before being combined with any other expression term. The expression is always evaluated by the system as an integer; all fractional values are truncated, not rounded. For example, 5/3 is passed to the system as having a value of 1. At each step of the evaluation of an expression, fractional values are truncated; that is, integer arithmetic is performed.

- $\pm$ expression

  Specifies a signed value, plus or minus, for the expression. Note that "+expression" has the same value as "expression".

- x (where x is the name of an already defined variable)

Variables may be used when specifying the number of trap save areas and intermediate request blocks in a SYS directive, or the size of a memory pool in a MEMPOOL directive. Using a VARIABLE directive, you can assign symbolic values for these fields that are subsequently resolved by the system. By specifying variables in directives, you create a generalized CLM file that can be used by several similar but not identical hardware configurations. *

A communication device type is counted once per each unique channel number. For example, in the following:

```
VARIABLE A=TTY*3000
         .
         .
DEVICE STD01,25,25,X'C000'
DEVICE STD02,26,25,X'C000'
DEVICE STD03,27,25,X'C000'
```

* the final value given to variable A is 1, not 3, since there is only one unique channel for the STD device types.

Example 1:

```
            .
            .
            .
VARIABLE A=TTY*3000
MEMPOOL I,T1,A
MEMPOOL S,,*
            .
            .
            .
```

Note that in all cases, a variable must be defined in the CLM file before it can be used in subsequent directives. The variable A has been defined to be equal to the total number of TTY terminals finally configured, multiplied by 3000. In the MEMPOOL directive, A represents the 3000 words of memory that are reserved for each TTY configured. The first MEMPOOL directive defines the name and size of the memory pool that has reserved 3000 words of memory for each TTY terminal. The second MEMPOOL directive allocates all remaining memory to the system memory pool.

By defining memory pools with variables, you can accurately account for as many or as few devices as are actually configured. You need not know the channel number assigned to
* each device, or the number of devices actually configured. The system assigns these characteristics accurately and automatically.

Example 2:

```
            .
            .
            .
VARIABLE B=TTY*3
VARIABLE C=TTY*2
SYS  ,,,,C,B
            .
            .
            .
```

In this example, two variables have been defined for the trap save area and intermediate request block fields in the SYS directive. B is defined as the number of TTY terminals times 3, and C is defined as the number of TTY terminals times 2. The SYS directive will result in the system having the default number of trap save areas plus twice as many trap save areas as there are TTY terminals and the default number of intermediate request block fields plus 3 times as many intermediate request block fields as there are TTY terminals.

Example 3:

Sample CLM File

```
    VARIABLE T=TTY
(1) VARIABLE I=TTY+10
    SYS ,,,,I,T,,E
    DEVICE DSK00,6,10,X'400'
    DEVICE DSK01,7,11,0
    DEVICE FCM00,8,12,0
    DEVICE RCM00,9,12,0
    COMM 13
    DEVICE TTY01,14,14,0
    DEVICE TTY02,15,15,0
    DEVICE TTY03,16,16,0
(2) DEVICE TTY04,17,17,0
    TTY 14,14,0,0,300
    TTY 15,15,0,0,300
    TTY 16,16,0,0,300
    TTY 17,17,0,0,300
(3) VARIABLE Z=TTY*500
    MEMPOOL I,L0,Z
(4) MEMPOOL S,,*
    QUIT
```

The following comments refer to the reference numbers in Example 3:

1. Variables T and I have been defined in VARIABLE directives replacing constant values in the SYS directive. T equals the total number of TTY terminals configured. I equals 10 plus the number of TTY terminals configured. In the SYS directive, T replaces some constant value in the trap save area field; I replaces some constant value in the intermediate request block field. In this context, T specifies one additional trap save area for each TTY configured; I specifies that the number of additional intermediate request blocks equals 10 more than the number of TTY terminals configured.

2. A maximum of four TTY terminals may be configured, depending on the number of available channels (with asynchronous line adapters). All terminals have floating channel assignments and are configured in order. *

3. The variable Z is defined such that 500 words of memory are allocated for each configured TTY terminal. (Z equals 500 times the number of configured TTY terminals.) In the MEMPOOL directive that follows, Z specifies that pool L0 allocates 500 words of memory for each TTY configured. The size of pool L0 could be 500, 1000, 1500, or 2000 words, depending on the final number of TTY terminals actually configured.

4. This MEMPOOL directive allocates all remaining unallocated memory to the system memory pool.

REMOVE THIS PAGE AND PLACE TAB FOR

TAB 6

CLM DIRECTIVES

Section 6
CLM DIRECTIVES FOR A COMMUNICATIONS CONFIGURATION

This section describes the CLM directives used to define the communications environment of a MOD 400 system. Communications CLM directives can be entered from the CLM_USER file or from any file or device specified in a CLMIN directive. Thereafter, system startup can be initiated.

If your installation includes communications devices, you should have the System Programmer's Guide available for reference.

The communications-related CLM directives cause the following functions to be performed:

- Data structures (that is, tables) are established to support the communications hardware on your system.
- The following bound units are loaded into the central processor's main memory:
  - Communications supervisor and online (communications controller) driver
  - One or more line protocol handlers (for example, ATD, STD, BSC, PVE, HASP, or user-written). *

- The following elements are loaded into the memory of one or more communications controllers:

  - Data-set channel (Dial CCP) control program

  - Channel control programs of one or more line protocol handlers.

Table 6-1 summarizes the communications-related CLM directives, which are described in alphabetic order in this section. Table 6-2 provides detailed information on physical devices supported by various directives. The term "station" refers to a physical device on a point-to-point or multipoint communications line or to a logical endpoint on a multiport communications line. The term "line" refers to a physical communications line to which either logical endpoints or physical devices are attached.

Table 6-1. Summary of Communications-Related CLM Directives

Directive (Mnemonic)	Meaning
ACU	Defines an Auto Call Unit (ACU) and associates the ACU channel with a data communications channel. Optionally, provides one or more telephone numbers to be used in the establishment of a connection on the associated data communications channel.
ASP	Identifies an Asynchronous Serial Printer (ASP) station to be supported by the ATD line protocol handler.
ATD	Identifies a station on a line serviced by the ATD line protocol handler.
AXD	Identifies a station on a line to be serviced by the Asynchronous Terminal Device (ATD) line protocol handler as an asynchronous X-ON/X-OFF device.
BPA	Identifies that a receive-only printer is connected to a station on a line serviced by the ATD line protocol handler.
BSC	Identifies a station on a line serviced by the Binary Synchronous Communications (BSC) 2780/3780 line protocol handler.
COMM	Establishes the priority level(s) at which the communications processor interrupts the central processor.

Table 6-1 (cont). Summary of Communications-Related CLM Directives

Directive (Mnemonic)	Meaning
DEVICE	Identifies a communications station to the file system.
DSK	Indicates that a VTS7760/7740 diskette is connected to a station on a line serviced by the Synchronous Terminal Driver (STD) line protocol handler. The line is identified by an STDLN directive and the station is identified by an STD directive.
EQLRN	Permits the multiple definition of logical resource numbers (lrns). This allows for different lrns to reference the same station.
HASP	Identifies a station on a line serviced by the HASP protocol.
H3270	Identifies an emulated station on a line serviced by the BSC 3270 protocol.
I327LN	Identifies an IBM 3270 terminal link serviced by the BSC3270 terminal facility (BTF).
I3270	Defines the 3270 device poll address.
LPHn	Identifies the first (or only) station on a line serviced by a user-written line protocol handler.
LPHDEF	Indicates table extension sizes for channels and stations controlled by a user-written line protocol handler.
MODEM	Defines a nonstandard modem type.
POLIST	Identifies the poll stall interval on a line serviced by the STD line protocol handler. The line is identified by a STDLN directive.
PVE	Identifies a polled VIP emulator station on a line serviced by a driver that supports the PVE protocol.
PVELN	Identifies a line serviced by the PVE line protocol handler.
ROP	Indicates that a receive-only printer is connected to a station on a line serviced by the STD line protocol handler. The line is identified by a STDLN directive and the station is identified by a STD directive.

*

Table 6-1 (cont). Summary of Communications-Related CLM Directives

Directive (Mnemonic)	Meaning
STAPOL	Defines a sequence of station poll addresses to be added to the poll list of a line serviced by the STD line protocol handler. The line is identified by a STDLN directive.
STATION	Identifies the second or subsequent station on a line serviced by a user-written line protocol handler.
STD	Identifies a station on a line serviced by the STD line protocol handler. The line is identified by a STDLN directive.
STDLN	Identifies a line serviced by the STD line protocol handler.
STTY	Specifies the file characteristics of a communications station that is to be serviced as a file system device.
TIMEOUT	Specifies timeout values (in seconds) for general communications line inactivity and for the completion of connect, disconnect, read, and write processing.
TTY	Identifies a station on a line serviced by the ATD line protocol handler.
TTY*	Identifies a station on a line serviced by the teleprinter (TTY) line protocol handler.
TYL	Identifies a station on a line serviced by the TYL line protocol handler.
VDAM	Incorporates the Display Formatting and Control (DFC) software component in the system configuration.

NOTE

ATD will ultimately replace the TTY line protocol handler. The TTY directive now causes ATD and not TTY to be loaded. If you wish to load TTY, you must use the TTY* directive. TTY supports the physical terminal in command (teleprinter) mode only, and does not support block mode for VIP7800 devices nor field mode for VIP7200, VIP7300, or VIP7800 devices. (The ATD directive is required for field or block mode support.) ATD does not provide the transparent I/O or single character mode functions of TTY*.

Table 6-2. CLM Directives and Supported Communications Devices

<table>
<tr><th>Directive
(Mnemonic)</th><th>Device Supported</th></tr>
<tr><td>ATD</td><td>TTY VIP7100 VIP7303 VIP7808 VIP7825
TN0300 VIP7200 VIP7305 VIP7809 HDS 2
TN1200 VIP7201 VIP7307 VIP7813
TWU1001 VIP7205 VIP7801 VIP7814
TWU1003 VIP7207 VIP7802 VIP7815
TWU1005 VIP7301 VIP7803 VIP7824

<u>Receive-only printers</u>:

PRU1004 PRU7076 PRU7176 PRU7211 PRU7271
PRU7007 PRU7077 PRU7177 PRU7212 PRU7272
PRU7070 PRU7170 PRU7200 PRU7260
PRU7071 PRU7171 PRU7201 PRU7261
PRU7072 PRU7172 PRU7202 PRU7262
PRU7075 PRU7175 PRU7210 PRU7270</td></tr>
<tr><td>AXD</td><td>Any asynchronous communication device which supports, or is atleast insensitive to, the X-ON/X-OFF protocol.</td></tr>
<tr><td>BSC</td><td>DPS 6 central processor
IBM central processor</td></tr>
<tr><td>HASP</td><td>IBM central processor</td></tr>
<tr><td>H3270</td><td>IBM central processor</td></tr>
<tr><td>I3270</td><td><u>IBM Controllers</u>:

3271 models 1 and 2
3274 models 1C and 51C
3276 models 2, 3 and 4

<u>Attached devices</u>:

3277 display model 2
3278 display models 2, 3, 4 and 5
3279 display models 2 and 3
3287 printer models 1 and 2</td></tr>
<tr><td>PVE</td><td>DPS 6 central processor
Level 66 host processor</td></tr>
</table>

Table 6-2 (cont). CLM Directives and Supported Communications Devices

Directive (Mnemonic)	Device Supported
STD	PRU1901 VIP7705R VIP7815 VIP7825 VTS7740 TWU1901 VIP7804 VIP7816 VIP7826 VTS7760 VIP7700 VIP7805 VIP7817 VIP7827 VIP7700R VIP7814 VIP7824 VTS7710 Receive-only printers: PRU1003 PRU7070 TN300 PRU1005 PRU7075 TN1200
TTY*	TTY TWU1003 VIP7100 VIP7301 TWU1001 TWU1005 VIP7200 VIP7801

The configuration of the following devices is described in an appendix: Decision Data 8045 Card Reader/Punch (Appendix L), FACIT 4042 Paper Tape Reader/Punch (Appendix M), and Memodyne M-80 Cassette Tape unit (Appendix N).

TOPICS RELATED TO COMMUNICATIONS DIRECTIVES

The following paragraphs describe topics you should review before creating your CLM directive file.

Assigning Logical Resource Numbers

Each communications station, endpoint or device is assigned a unique identifier called a logical resource number (lrn). The system uses the lrn to reference the communications device.

The value for an lrn is an integer from 3 to 252 or 256 to 4002 (decimal). Logical resource numbers 0, 1, 2, 253, 254 255, and 4003 to 4095 are reserved for system use. The communications connected operator's console must be assigned lrn 0.

To specify multiple communications directives that reference the same physical device, use the EQLRN directive.

Assigning Hardware Priority Levels

You cannot arbitrarily assign hardware priority levels--assignments are restricted by the considerations that follow.

Communications software requires at least two sets of hardware priority levels that must not be used for noncommunications devices or tasks. One set of levels is dedicated to the processing of communications device interrupts. Up to four unique (nonshareable) levels can be assigned (in the COMM directive) for processing interrupts. The interrupt levels that can be used are those from 6 to 60 (levels 0, 1, 2, 3, 4, 5, 62, and 63 are required for system use).

The other set of priority levels is dedicated to the processing of communications task requests. Every communications device can have its own unique hardware level (subject to availability), or several devices can share a common priority level. In either case, the task hardware level(s) must have a lower priority (higher numeric value) than any of the levels assigned to service communications interrupts. The task hardware levels that can be used are those from 7 to 61 (levels 0, 1, 2, 3, 4, 5, 62, and 63 are required for system use). Note that if level 7 is assigned to be a communication task level, then level 6 must be a communication interrupt-processing level (at least for devices using level 7 as a communication "task level").

Except for the above restrictions, you can assign any other priority level to any noncommunication device or task.

<u>Assigning Channel Numbers</u>

Channel control programs are loaded into the communications controller on the basis of specified channel numbers. You should observe the following guidelines when assigning channel numbers:

1. Because there is a Channel Control Program (CCP) for each Honeywell-supported line protocol handler, as a general rule no more than two line protocol handlers can be associated with (loaded into) a given MLCP. This limitation is based on the fact that no more than 3072 bytes of MLCP memory are available for the loading of CCPs. Any combination of CCPs can be loaded into the MLCP provided the total does not exceed 3072 bytes.

2. The same restrictions also apply to the number of line protocol handlers that can be loaded into a given MLC-16, with this difference: the number of bytes of MLC-16 memory is 16K. Consequently, the total number of CCPs loaded into the MLC-16 must not exceed 16K bytes.

3. As shown in the following chart, the first six bits of a channel number directly relate to the MLCP or MLC-16 associated with that number:

Channel Numbers	Associated Communications Controllers
FC00 - FF80	first MLCP
F800 - FB80	second MLCP
F400 - F780	third MLCP
F000 - F380	fourth MLCP
EC00 - EF80	fifth MLCP
.	.
.	.
C000 - C780	first MLC-16
C800 - CF80	second MLC-16
D000 - D780	third MLC-16
.	.
.	.

4. Each channel number must be a four-digit hexadecimal number ending in 00 or 80. Channel numbers lower than 0C00 cannot be used by DPS 6 communications devices.

5. Channel numbers can be dynamically assigned or "floated" (with due care) for the TTY*, ATD, and related DEVICE directives. Channel numbers should not be floated when using PVELN, H3270, STDLN, BSC, HASP or LPHn directives.

   For each communications device configured, there is the option of specifying a channel number in the directive or allowing the system to dynamically assign or "float" channel assignments at startup time. To float a channel, you specify the channel number parameter in the directive with a single zero. The system then assigns directives with floating channels to the first available device. Note that if you specify a floating channel and the system cannot locate an appropriate device, no error message is issued; the system ignores the directive.

   You should not use floating channel numbers if there is any chance that the system will associate an LPH with the wrong channel.

   If your system includes device adapters that do not directly support attached devices, it is recommended that explicit channel numbers be assigned to such adapters.

   Channel numbers are assigned as soon as the first floating channel is encountered in the CLM file. To prevent devices with floating channels from inadvertently precluding devices whose channel numbers are explicitly specified, all directives of a given type with explicit channel numbers should precede all directives of the same type with floating channels. All active channels should have higher bus addresses than all inactive channels to prevent the system from inadvertently assigning a device to an inactive channel.

For communications devices, the first available channel (with the highest bus address and appropriate channel adapter) is assigned to the first floatable channel. This process continues with the next highest free channel (of the appropriate type) when the next floatable channel is encountered.

Assigning floatable channels to communications devices is recommended only if all devices are configured identically. For example, a system could specify that all asynchronous devices be direct-connect VIP7200s with identical line speeds, DEL characters, stop bits, and parity. The system cannot distinguish between different types of communications devices.

If a system is configured with devices containing floatable channel numbers, you can determine the channel numbers assigned by the system by issuing the STS -ALL command to list each device and its channel number assignment. *

Selection of Modem Types

Nine standard modem types provide support for direct communications connections and for data sets supporting the DTE/DCE interface standard EIA RS-232C (with and without ring indicator and end-of-number auto call support). Support of nonstandard modems (subject to hardware support of the Data Communications Equipment (DCE) interface) is provided by the MODEM directive.

The nine standard modem types are summarized in Table 6-3. You specify modem type by specifying the appropriate digit (0 through 8) in the appropriate configuration directive.

Terminal Line Speed Selection Capability (Asynchronous Terminals Only)

When you specify an ATD, TTY*, or LPHn directive, you have the option of deferring selection of an asynchronous terminal's line speed (specified by the fifth argument) until the terminal comes online. If you choose this option, an ACU must not be configured for the terminal. To defer selection of a terminal's line speed, specify the fifth parameter in one of the above-mentioned directives as either 'HI' or 'LO'. Choosing this option frees you from specifying an exact line speed for every asynchronous terminal in your configuration. *

Table 6-3. Standard Modem Types

Modem Type	Characteristics
0	Direct connect Connection is initiated by setting Data Terminal Ready (DTR) high (on synchronous adapter an Enable Direct Connect Clock's is set high). Connection requires Data Set Ready (DSR) to be high. Disconnection requires DSR to go low.
1	DSR support; Carrier Detect (CD) support Connection is initiated by setting DTR high. Connection requires DSR and CD to be high. Auto call (ACU) operation is conducted without end-of-number (EON) support. Disconnection requires DSR and CD to be low.
2	DSR support Connection is initiated by setting DTR high. Connection requires DSR to be high. ACU operation is conducted without EON support. Disconnection requires DSR to be low.
3	DSR support; CD support; Ring Indicator (RI) support On auto-answer operations RI must be high before DTR is set high. Connection requires DSR and CD to be high. ACU operation is conducted without EON support. Disconnection requires DSR and CD to be low.

Table 6-3 (cont). Standard Modem Types

Modem Type	Characteristics
4	DSR support; RI support On auto-answer operations RI must be high before DTR is set high. Connection requires DSR to be high. ACU operations are conducted without EON support. Disconnection requires DSR to be low.
5	DSR support; CD support; EON ACU support Same as Modem Type 1 except that the ACU always terminates a dialed number with an EON symbol.
6	DSR support; EON ACU support Same as Modem Type 2 except that ACU use implies EON support.
7	DSR support; CD support; RI support; EON ACU support Same as Modem Type 3 except that ACU use implies EON support.
8	DSR support; RI support; EON ACU support Same as Modem Type 4 except that ACU use implies EON support.

In order to allow the system to determine the terminal's line speed, the operator presses the RETURN key on the terminal's keyboard when the terminal comes online (when DTR goes high). If, after several seconds, normal terminal operations have not begun, the operator should check for one of the following problems:

1. The terminal's line speed may not be within the range specified in the appropriate CLM directive. The operator may be able to adjust the terminal's line speed to fall within the range specified by the directive. In this case, the operator should then press the RETURN key. If, however, the terminal's line speed is fixed, adjustments must be made to the appropriate CLM directive. You may be required to change the designation of the terminal's line speed in the appropriate CLM directive.

2. If the carriage return character was not recognized and the system could not determine the line speed of the terminal, the operator should press the RETURN key again.

3. If the terminal's modem has gone offline, the operator must reestablish the telephone connection.

If the system fails to receive a character transmission within 2 minutes and 30 seconds, the terminal is disconnected.

Self-Configuring Devices

When you specify an ASP, ATD, BPA, or STD directive, you have the option of deferring selection of the device type (specified by the device type argument) until a connect is issued to the device. If the device type argument for one of these directives contains an asterisk (*), the line protocol handler configures the device after the first connect is received for that device. The following restrictions apply to self-configuring devices:

1. For the ASP directive, the device must be either a PRU717x, PRU720x, PRU721x, PRU727x, or a PRU1004.

2. For the ATD directive, the device must be either a VIP781x, VIP782x, VIP730x, HDS 2, or a VIP7201.

3. For the STD directive, VIPs or any combination of CRT and ROP or CRT and DISKETTE are considered self-configuring devices. PVE, POLY21, or EMULATORS should be specified at CLM time and are not supported as self-configuring devices.

4. The device must be configured at a baud rate that is suitable for all devices that may be connected to the communication channel.

5. Self-configuration is performed at connect time by sending an enquiry sequence to the device and setting the configuration based on the response. If a valid response is received but it's not a known device, the device is configured as a PRU1004 when using the ASP directive. For the ATD directive, the device is configured as a VIP7200.

6. Applications that determine the device type by issuing a "get device information" monitor call should be aware that prior to a connect being issued the device type is x"FF". A valid device type is returned only after a satisfactory connection has been established.

8-Bit Data Support

When you specify an ATD, AXD, PVELN or STDLN directive, you have the option of selecting either an 8-bit data path or the appropriate algorithms used when an application uses 8-bit data on a 7-bit communication channel. The number of data bits that can be transmitted on a communications channel is specified by the number-of-data-bits argument of these directives.

If the communication channel does not support 8-bit data, the directive may also specify truncation or shift-in/shift-out. This is accomplished by the compression algorithm argument. If truncation is selected, the transmitted data has the high-order bit removed. If shift-in/shift-out is specified, a shift-in character (hex 0F) precedes the first character to be transmitted that has a high-order bit equal to one. Before the next character in the data stream that has a high-order bit equal to zero is transmitted, a shift-out character (hex 0E) is sent. A shift-in or shift-out character continues to precede the actual data character every time the high-order bit changes. In this way, 8-bit data is sent over a 7-bit communication channel.

General Rules for the Arrangement of Communications Directives

The following rules summarize the required arrangement of communications directives in a CLM file:

1. The COMM directive must precede all other communications-related CLM directives.

2. If an LPHDEF directive is used in association with a user-written line protocol handler, the LPHDEF directive must precede all related LPHn directives.

3. If STATION directives are used in association with an LPHn directive (in cases where a user-written line protocol handler supports more than one station per line), the STATION directives must immediately follow the related LPHn directive.

4. The DEVICE directive should follow related LPHDEF, STATION, and LPHn directives, or related POLIST, STAPOL, ROP, DSK, STDLN, and STD directives.

5. If an ACU directive is used, the station whose lrn appears as an argument in the ACU directive must be defined in the directive that precedes it.

6. When polled VIP emulator (PVE) stations are associated with a communications line, the PVE directives for all the stations on the line must be consecutive.

7. Configuring a line serviced by the STD line protocol handler requires the following directives: STDLN, STD, POLIST, and STAPOL. An STD line can also use an ROP and/or DSK directive. The DSK (diskette) directive is meaningful only with VTS7760 or VTS7740 terminals. The following list illustrates the sequence rules for STD directives. Brackets enclose optional directives. One set of these directives must be specified for each line:

STDLN	Must specify for each line; all related STD, ROP, DSK POLIST and STAPOL directives must follow
STD	One for each station on the line
[ROP]	Optional receive-only printer associated with prior station
[DSK]	Optional 7760/7740 diskette associated with prior station
.	
.	
.	
[STD]	
[ROP]	
[DSK]	
.	
.	
.	
POLIST	Must specify for each line
STAPOL	Must specify one or more for each line; all STAPOL poll addresses should be specified in the order in which stations are to be polled
[DEVICE]	Specifies that the STD or ROP device is accessible to the file system. The DSK device is only accessible through the physical I/O (PIO) interface.

8. The STTY command keyword format, as described in the STTY directive, can be used as an alternative to the STTY directive format in your CLM file. The STTY command format provides additional features not supported by the STTY directive format. For example, if you wish to specify a terminal as being automatically reconnectable following a power failure or line drop, you must configure the terminal using the STTY command format in your CLM file. (For a complete description of restart capability, see Appendix D. For a complete description of the STTY command, refer to the Commands manual.)

9. There are two forms of the ASP directive. One form is used to define stand-alone printers that use the Asynchronous Serial Interface (ASPI). The other form is used to define receive-only printers that are attached to the buffer printer adapter (VAF7821) of a VIP7813, VIP7814, VIP7815, VIP7801, or VIP7808. In the latter instance, the ASP directive (which defines the attached printer) must immediately follow the ATD directive defining the terminal that contains the buffer printer adapter. This form of the ASP directive is being replaced by the BPA directive.

10. Configuring a line serviced by the BTF line protocol handler requires the following directives: I327LN, I3270, and DEVICE. In order to maintain efficient memory usage, the BTF maintains a master-lrn for each line. The following list illustrates the sequence rules for BTF directives. One set of these directives must be specified for each line:

I327LN	Must specify for each line; all related I3270 and DEVICE directives must follow
I3270	One for each station on the line
DEVICE	Specifies that the BTF is accessible to the file system.

11. The BPA directive (if used) must immediately follow the corresponding ATD directive.

Resident Code Requirements for Communications Modules

There are 2944 bytes of RAM memory available in the MLCP for the loading of channel control programs and 16K in the MLC-16. Any combination of the various CCPs can be loaded into the MLCP, provided the total memory does not exceed 2944 bytes. The number of words of resident code required for various communications modules and the memory required in the communications controller for associated Channel Control Programs (CCPs) are listed in the Software Release Bulletin.

* COMMUNICATIONS CLM DIRECTIVES SET

The CLM directives for communications configuration are described on the following pages, in alphabetical order.

AUTO CALL UNIT (ACU)

The ACU directive identifies an Auto Call Unit (ACU) and associates the ACU channel with a data communications channel. The ACU directive must follow the directive describing the station whose lrn is specified in the ACU directive. The ACU directive cannot be associated with any data communications channel that supports the speed select option.

NOTE

> Both the communications adapter that supports the ACU and the communications line to be serviced by the ACU (on whose behalf calls are initiated) must be attached to the same communications controller. For example, an ACU on channel FC00 can only support auto call operations on adapters within the same MLCP or MLC-16; namely, those on channels FD00, FD80, FE00, FE80, FF00, or FF80 for the MLCP and those on channels F880, F900, ... , or FF80 for the MLC-16.

The ACU directive permits the addition of telephone numbers to a list of numbers maintained for a data communications channel. The list of telephone numbers for a data communications channel can be unlimited and has the following format:

Entry 0 empty (initially)

[Entry 1 phone_number 1]

[Entry 2 phone_number 2]

[etc.]

The ACU dials each number in the list three times at 50-second intervals until the list is exhausted or a connection is made.

The first ACU directive that relates an ACU to this data communications channel causes an empty entry 0 to be established. This 0 entry in the list can be loaded and reloaded, as desired, by means of an SDL (set ACU telephone number) command or $SDL macrocall. The first ACU directive also creates an entry in the table (starting with entry 1) for each telephone number (if any) specified in the directive. Any subsequent ACU directive relating the same ACU to the same data communications channel causes one or more additional entries to be added to the list. The additional entries are added to the end of the list in the order in which the telephone numbers appear in the ACU directive.

FORMAT:

```
ACU lrn,[level],X'channel'[,'phone_#1'[,'phone_#2'[,...]]]
```

ARGUMENTS:

lrn

The logical resource number of any station on the data communications channel with which this ACU is associated.

[level]

The priority level of the station whose lrn appears in the lrn argument of this directive. Normally, this argument must be specified. It is optional only with the LLHB directive.

X'channel'

A four-digit hexadecimal number (from X'0400' to X'FF80') specifying the channel number of the ACU. The channel number has the following format:

Bits 0 through 9 - The 10-bit channel address of the send or receive channel on the line.

Bits 10 through 15 - Must be set to zero.

['phone_#n']

A string of 1 to 30 ASCII characters chosen from the set 0 1 2 3 4 5 6 7 8 9 - (hyphen) * (asterisk). The hyphen indicates a request for secondary dial tone before proceeding with the remaining dial operation. A hyphen can be used to gain access to an outside exchange from an internal (PBX) exchange. Certain modems may require an asterisk instead of a hyphen. A telephone number can contain at most one hyphen or asterisk.

DESCRIPTION:

An ACU directive ultimately causes the ACU to initiate a line connection with a remote auto answer data set. When the software issues a connect order and bit 2 of the device specific word is set to 1, the ACU attempts to dial a line using the list of telephone numbers established at configuration time. The ACU dials each number in the list three times at 50-second intervals until a connection is made or the list is exhausted.

Example:

```
ATD 26,8,X'FC00'

ACU 26,8,X'FD00','15552400281'
```

In this example, an ACU on channel FD00 is associated with the data communications channel (FC00) servicing a TTY whose logical resource number is 26 and whose priority level is 8. Since this is the first ACU directive for this ACU, the telephone number in the ACU directive is established as entry 1 in the list of telephone numbers for the indicated data communications channel (FC00). Note that the ACU directive follows the ATD directive describing the station whose lrn appears in the ACU directive.

ASYNCHRONOUS SERIAL PRINTER (ASP)

The ASP directive identifies a serial printer to be serviced by the asynchronous terminal driver line protocol handler. There are two uses of this directive. Each use requires a different format. (See Formats 1 and 2 described below.) The first use of the directive is to define a serial printer which is attached to a VIP7800 type buffer printer adapter (VAF7821). To define a serial printer, an ATD directive must precede the ASP directive; the ATD directive is used to define the terminal that supports the buffer printer adapter. The ATD directive is followed by the ASP directive defining the buffer printer adapter and attached serial printer. This format of the ASP directive is being replaced by the BPA directive. If you are defining a serial printer attached to a buffer printer adapter, it is recomended that you use the BPA directive instead of the ASP directive.

FORMAT 1:

```
ATD lrn,level,X'channel',[modem],[speed],['device_type'],
    [del],[stop_bit][,parity][,HD/FD][,VIPSIM]
    [,data_bits][,compression_algorithm]

ASP lrn,level,X'channel'
```

The lrn, level, and channel numbers must be the same in both directives. Only the lrn, level, and channel number need to be defined in the ASP directive, as the rest of the station specific information is taken from the associated ATD directive.

The second use of the directive is to define a serial printer that is not attached to a buffer printer adapter.

FORMAT 2:

```
ASP lrn,level,X'channel',[modem],[speed],'device_type'
```

The modem and speed arguments are optional. You must specify device-type.

ARGUMENTS:

lrn

The logical resource number associated with the station. The value for lrn is an integer from 3 to 252, or 256 to 4002. A program can use this number to identify the station when it requests an input/output operation to the station.

level

The priority level at which the line protocol handler processes requests for an input/output operation to the station. The value for level is an integer from 7 through 61. It can be the same as the level specified for other communications stations, but it must be a higher number than the communications interrupt level(s) specified in the COMM directive. The level specified for one or more communications stations must not be used for noncommunications devices or tasks.

X'channel'

A four-digit hexadecimal number (from X'080n' to X'FF8n'), specifying the channel number of the station and the interrupt priority level (n) of the line. The channel number has the following format:

Bits 0 through 9 - The 10-bit channel address of the line.

Bits 10 through 13 - Must be zero.

Bits 14 and 15 - Specify n, the priority level at which a communications line interrupts the central processor. n can have a value of 0, 1, 2, or 3 as specified in the COMM directive. (See the description of the COMM directive for further information.)

You can float this channel assignment by specifying a single zero, if you wish.

[modem]

A one-digit number that specifies modem type: what modem signals are set to establish a data-set connection, what signals are required for a connection, and what signals are required for a disconnection. Possible values are: 0, 1, 3, 5, 7, or 9 through 15 (2, 4, 6, and 8 are synchronous only). See Table 6-3 for a description of modem types. The default value is 1.

speed

The data rate in bits per second. Use the following table to determine the proper speed (based on the device configuration):

Device	Allowable Speeds
1004	110, 300, 1200
7007	1200
7070	300, 1200, 9600
7075	300, 1200, 9600
7170	300, 1200, 9600
7175	300, 1200, 9600
7260	300, 1200, 9600
7270	300, 1200, 9600
P7200	300, 1200, 9600

'device_type'

Specifies the type of serial printer that is supported.

There is no default device type for this printer. Possible values are:

Value	Physical Devices Supported
*	A self-configuring device
1004	PRU1004
7007	PRU7007
7070	PRU7070, PRU7071, PRU7072
7075	PRU7075, PRU7076, PRU7077
7170	PRU7170, PRU7171, PRU7172
7175	PRU7175, PRU7176, PRU7177
7260	PRU7260, PRU7261, PRU7262
7270	PRU7270, PRU7271, PRU7272
P7200	PRU7200, PRU7201, PRU7202, PRU7210, PRU7211, PRU7212

*

For more information about self-configuring devices, refer to the discussion under "Topics Related to Communications Directives" at the beginning of this section.

Example:

```
ASP 15,17,x'F800',0,9600,'7070'
```

In this example, a PRU7070 Printer is connected to a line serviced by the ATD Line Protocol Handler. *

ASYNCHRONOUS TERMINAL DEVICE (ATD)

The ATD directive identifies a station on a line serviced by the asynchronous terminal device line protocol handler. This directive is used to configure asynchronous terminals as part of the communications system. The ATD directives must follow the COMM directive in the CLM file.

FORMAT:

```
ATD lrn,level,X'channel',[modem],[speed],['device_type'],
    [del],[stop_bit][,parity][,HD/FD][,VIPSIM]
    [,data_bits][,compression_algorithm]
```

ARGUMENTS:

lrn

The logical resource number associated with the station. The value for lrn is an integer from 3 to 252, or 256 to 4002. A program can use this number to identify the station when it requests an input/output operation to the station.

level

The priority level at which the line protocol handler processes requests for an input/output operation to the station. The value for level is an integer from 7 through 61. It can be the same as the level specified for other communications stations, but it must be a higher number than the communications interrupt level(s) specified in the COMM directive. The level specified for one or more communications stations must not be used for noncommunications devices or tasks.

X'channel'

A four-digit hexadecimal number (from X'080n' to X'FF8n'), specifying the channel number of the station and the interrupt priority level (n) of the line. The channel number has the following format:

Bits 0 through 9 - The 10-bit channel address of the line.

Bits 10 through 13 - Must be set to zero.

Bits 14 and 15 - Specify n, the priority level at which a communications line interrupts the central processor. n may have a value of 0, 1, 2, or 3 as specified in the COMM directive. (See the description of the COMM directive for further information.)

You can float this channel assignment by specifying a single zero, if you wish.

[modem]

A one-digit number that specifies modem type: what modem signals are set to establish a data-set connection, what signals are required for a connection, and what signals are required for a disconnection. Possible values are: 0, 1, 3, 5, 7, or 9 through 15 (2, 4, 6, and 8 are synchronous only). See Table 6-3 for a description of modem types. The default value is 1.

[speed]

The data rate in bits per second.

For an asynchronous line with a communications-pac whose id is hexadecimal 2108, use one of the following values for speed:

	50	300	2400
	75	600	3600
(default)	110	900	4800
	134	1200	7200
	150	1800	9600

If the data rate is 134.5, specify 134.

For an asynchronous line with a communications-pac whose id is hexadecimal 2110, 2118, 2178, 2978, 3518, 3519, 3618 or 3619, use one of the following values for speed:

	50	200	1800
	75	300	2000
(default)	110	600	2400
	134	1050	4800
	150	1200	9600

If the data rate is 134.5, specify 134.

If the adapter is one of the adapter types in the second set, you have the option to defer selection of the line speed until the terminal comes online. You select this option by specifying HI or LO.

'HI'

All terminals associated with this line are permitted to function only at speeds from 1200 through 9600 bits per seconds.

'LO'

All terminals associated with this line are permitted to function only at speeds from 110 through 1200 bits per second.

For more information about the line speed selection capability, refer to the discussion under "Topics Related to Communications Directives."

['device_type']

The type of terminal used. If this argument is not specified, the default is TTY. Possible values are:

Value	Physical Device
*	A self-configuring device
TTY	An asynchronous terminal supported as a teleprinter (TTY) compatible device
7100	VIP7100
7200	VIP7200, VIP7205
7201	VIP7201
7207	VIP7207
7801	VIP7801, VIP7802
7803	VIP7803
7808	VIP7808, VIP7809
7813	VIP7813
7814A	VIP7814, VIP7815 (asynch)
7301	VIP7301
7303	VIP7303

Value	Physical Device
7305	VIP7305
7307	VIP7307
0300	TermiNet 300
7824A	VIP7824, VIP7825
1200	TermiNet 1200
PRU	TWU1001, TWU1003, TWU1005
1001	TWU1001
1003	TWU1003
1005	TWU1005
4100J	4100J printer
4110J	4110J printer
7802J	VIP7802J
7813J	VIP7813J
7814J	VIP7814J
VIP8300	HDS 2

For more information about self-configuring devices, refer to the discussion under "Topics Related to Communications Directives."

[del]

In TTY mode this argument specifies the number of DEL (X'7F') characters to be output at the end of an individual write order when such orders are issued to one of the following terminals: TTY, PRU, 7100, 7200, 7207, 7801, 7808, 7301, 7303, or 7307. The output of trailing DEL characters is independent of any post-order write processing (that is, the optional output of a carriage return and/or a line feed).

This argument allows you to specify from 0 to 32 DEL characters. If this argument is not specified, no trailing DEL characters are output.

[stop_bit]

The number of stop bits that are to follow each character. A value of 1 or 2 can be chosen for each device. Default values are as follows:

Number of Stop Bits

1 (For speeds greater than 110 bits per second)

2 (For a speed of 110 bits per second or less)

[parity]

The type of parity (ODD, EVEN or NONE) to be used. The default is EVEN.

[HD/FD]

Specifies whether this line is half-duplex or full-duplex. If full-duplex, Request-to-Send (RTS) is maintained high at all times. If half-duplex, RTS is only turned on when transmitting.

[VIPSIM]

Specifies that this line simulates a terminal.

[data_bits]

The number of data bits that can be transmitted on this communication link. A value of 7 or 8 can be chosen for each device. The default value is 7. If 8 is selected and the communication controller is either an MLCP or a 6/20, then NONE must be chosen for the parity option.

[compression_algorithm]

Specifies the compression algorithm that is used if a channel that is configured for 7-bit operation receives a byte of data with the high order bit set to one. Possible values are:

TRUNC - Unconditionally reset the high order bit.

SISO - Use shift-in/shift-out technique.

The default is TRUNC.

For more information about 8-bit support, refer to the discussion under "Topics Related to Communications Directives."

Example:

```
ATD 15,17,X'F800',0,9600,'7200',,EVEN,FD,7,TRUNC
```

In this example a VIP7200 terminal is connected to a line serviced by the ATD line protocol handler. The terminal is configured as direct connect and operates at 9600 bits per second. No DEL characters are output ahead of write orders and each character uses one stop bit. The size of a data character is seven bits and parity is even. If a byte is received with the high-order bit on, the bit is reset.

ASYNCHRONOUS X-ON/X-OFF DEVICE (AXD)

The AXD directive identifies a station on a line serviced by the asynchronous terminal device line protocol handler. This directive is used to configure a full duplex asynchronous communications line that supports an X-ON (DC-1)/X-OFF (DC-3) flow control protocol. The AXD line can be used for file transfer operations or to drive a terminal, printer, or any other asynchronous device. The associated DEVICE directive specifies the initial mode of AXD relative to the type of operation is intended for the line. For additional information on the use of AXD, refer to the discussion of "X-ON/X-OFF Mode" in the System Programmer's Guide (Vol. I).

FORMAT:

```
*    AXD lrn,level,X'channel',[modem],[speed],[del],[stop_bit]
         [,parity][,data_bits][,compression_algorithm]
```

ARGUMENTS:

lrn

The logical resource number associated with the station. The value for lrn is an integer from 3 to 252, or 256 to 4002. A program can use this number to identify the station when it requests an input/output operation to the station.

level

The priority level at which the line protocol handler processes requests for an input/output operation to the station. The value for level is an integer from 7 through 61. It can be the same as the level specified for other communications stations, but it must be a higher number than the communications interrupt level(s) specified in the COMM directive. The level specified for one or more communications stations must not be used for noncommunications devices or tasks.

X'channel'

A four-digit hexadecimal number (from X'080n' to X'FF8n'), specifying the channel number of the station and the interrupt priority level (n) of the line. The channel number has the following format:

Bits 0 through 9 - The 10-bit channel address of the line.

Bits 10 through 13 - Must be set to zero.

Bits 14 and 15 - Specify n, the priority level at which a communications line interrupts the central processor. n can have a value of 0, 1, 2, or 3 as specified in the COMM directive. (See the description of the COMM directive for further information.)

You can float this channel assignment by specifying a single zero, if you wish.

[modem]

A one-digit number that specifies modem type: what modem signals are set to establish a data-set connection, what signals are required for a connection, and what signals are required for a disconnection. Possible values are: 0, 1, 3, 5, 7, or 9 through 15 (2, 4, 6, and 8 are synchronous only). See Table 6-3 for a description of modem types. The default value is 1.

[speed]

The data rate in bits per second.

For an asynchronous line with a communications-pac whose id is hexadecimal 2108, use one of the following values for speed:

	50	300	2400
	75	600	3600
(default)	110	900	4800
	134	1200	7200
	150	1800	9600

If data rate is 134.5, specify 134.

For an asynchronous line with a communications-pac whose id is hexadecimal 2110, 2118, 2178, 2978, 3518, 3519, 3618 or 3619, use one of the following values for speed:

	50	200	1800
	75	300	2000
(default)	110	600	2400
	134	1050	4800
	150	1200	9600

If data rate is 134.5, specify 134.

* [del]

This argument specifies the number of DEL (X'7F') characters to be output in front of an individual write order. The output of leading DEL characters takes place before the control byte (if used) is processed.

This argument allows you to specify from 0 to 32 DEL characters. If this argument is not specified, no leading DEL characters are output.

[stop_bit]

The number of stop bits that are to follow each character. A value of 1 or 2 can be chosen for each device. Default values are as follows:

Number of Stop Bits

1 (For speeds greater than 110 bits per second)

2 (For a speed of 110 bits per second or less)

[parity]

The type of parity (ODD, EVEN or NONE) to be used. The default is EVEN.

[data_bits]

The number of data bits that can be transmitted on this communication link. A value of 7 or 8 can be chosen for each device. The default value is 7. If 8 is selected and the communication controller is either an MLCP or a 6/20, then NONE must be chosen for the parity option.

[compression_algorithm]

Specifies the compression algorithm that is used if a channel that is configured for 7-bit operation receives a byte of data with the high order bit set to one. Possible values are:

TRUNC - Unconditionally reset the high order bit.

SISO - Use shift-in/shift-out technique.

The default is TRUNC.

For more information about 8-bit support, refer to the discussion under "Topics Related to Communications Directives."

Example:

```
*        AXD 15,17,X'F800',0,9600,0,1,EVEN,7,TRUNC
*        DEVICE AXDT00,15,17,X'F800',AXD00,256
```

* In this example, a terminal is connected to a line controlled by the X-ON/X-OFF protocol. The terminal is configured as direct connect and operates at 9600 bits per second. No DEL characters are output ahead of write orders and each character uses one stop bit. The size of a data character is seven bits and parity is even. If a byte is received with the high-order bit on, the bit is reset.

The terminal is driven by the mode of AXD, which provides basic teletype support including character-cancel, line-delete, and optional post-order LF/CR on read orders. (Refer to the discussion of the DEVICE directive for more information about AXD device options.)

BUFFERED PRINT ADAPTER (BPA)

The BPA directive is used to define a serial printer attached to a VIP7800 type buffer printer adapter that is serviced by the ATD line protocol handler. An appropriate ATD directive must be paired with the BPA directive and the BPA directive must immediately follow the ATD directive.

FORMAT:

BPA ['device_type']

ARGUMENTS:

['device_type']

Specifies the type of serial printer that is supported. This device type depends on the device specified in the associated ATD directive. Only 780x, 781x, and 782x devices can support BPA.

If the ATD directive specifies a self-configuring device, the BPA device type must also be an asterisk (*) which specifies the printer as self-configuring. For more information about self-configuring devices, refer to the discussion under "Topics Related to Communications Directives" at the beginning of this section.

If the ATD directive specifies a 780x device, the following BPA device types are valid:

Value	Physical Devices Supported
1001	PRU1001
1003	PRU1003
1004	PRU1004
1005	PRU1005
7007	PRU7007

Default device type is 1005 when a 780x device is specified in the ATD directive.

If the ATD directive specifies a 781x or 782x device, the following BPA device types are valid:

Value	Physical Devices Supported
7070	PRU7070, PRU7071, PRU7072
7075	PRU7075, PRU7076, PRU7077
7170	PRU7170, PRU7171, PRU7172
7175	PRU7175, PRU7176, PRU7177
7270	PRU7270, PRU7271, PRU7272
P7200	PRU7200, PRU7201, PRU7202, PRU7210, PRU7211, PRU7212

Default device type is 7070 when a 781x or 782x device is specified in the ATD directive.

Example:

```
ATD 15,17,X;F800',0,9600,'7813'
BPA '7070'
DEVICE ATD00,15,17,X'F800',ATD01
DEVICE ATD00,15,17,X'F800',BPA00,,N,,,,ROP
```

In this example, a VIP7813 terminal with a VAF7821 Buffer Printer Adapter is connected to a line serviced by the ATD Line Protocol Handler. A PRU7070 printer is attached to the buffer printer adapter.

BINARY SYNCHRONOUS COMMUNICATIONS (BSC)

The BSC directive identifies a station serviced by the binary synchronous communications line protocol handler. If the station is to be accessed through the File System interface, an appropriate DEVICE directive must be paired with the BSC directive. (See "DEVICE (DEVICE)" later in this section.) If input/output to the station is asynchronous, the B (buffered) argument must be included in the DEVICE directive.

FORMAT:

```
BSC lrn,level,X'channel',[modem],[primary/secondary],
    [character_set][,multi_block_count]
```

ARGUMENTS:

lrn

The logical resource number associated with the station. The value for lrn is an integer from 3 to 252 or 256 to 4002. A program can use this number to identify the station when it requests an input/output operation to the station.

level

The priority level at which the line protocol handler processes requests for an input/output operation to the station. The value for level is an integer from 7 through 61. It can be the same as the level specified for other communications stations, but it must be a higher number than the communications interrupt level(s) specified in the COMM directive. The level specified for one or more communications stations must not be used for noncommunications devices or tasks.

X'channel'

A four-digit hexadecimal number (from X'080n' to X'FF8n'), specifying the channel number of the station and the interrupt priority level (n) of the line. The channel number has the following format:

Bits 0 through 9 - The 10-bit channel address of the line.

Bits 10 through 13 - Must be set to zero.

Bits 14 and 15 - Specify n, the priority level at which a communications line interrupts the central processor. n can have a value of 0,1,2, or 3 as specified in the COMM directive. See the description of the COMM directive for further information.

This directive does not support floating channel numbers.

[modem]

A one-digit number that specifies modem type: what modem signals is set to establish a data-set connection, what signals are required for a connection, and what signals for a disconnection. Possible values are: 0, 2, 4, 6, 8. For a description of each modem type, see Table 6-3. The default value is 2.

[primary/secondary]

Values can be specified as P or S; indicates whether this is a primary or secondary endpoint of the transmission. A primary endpoint (the default) has priority in contention bidding for use of the communications line.

[character_set]

One of the following can be specified:

- AS ASCII character set (the default)
- EB EBCDIC character set

The user is responsible for using the correct character set. The BSC protocol does not perform character translation.

[multi_block_count]

An integer from 2 to 7 specifying the maximum number of data blocks for a single transmission in multiblock mode. Omitting this parameter indicates no support of multiblock mode (the default).

Example:

```
BSC 29,8,X'FD00',2,S,EB

DEVICE  BSC00,29,8,X'FD00',HOST,,B
```

In this example, line FD00 is used for communications with another computer. Modem type 2 is used for the line. The DPS 6 computer is the secondary endpoint on the line. A File System interface is established for the station by the DEVICE directive. Multiblock mode is not used.

COMMUNICATIONS (COMM)

The COMM directive is mandatory in a system that includes communications. It specifies from one to four priority levels at which communications lines can interrupt the central processor. The COMM directive must precede all other communications-related CLM directives.

FORMAT:

```
COMM level_0 [,[level_1][,[level_2][,[level_3]]]]
```

ARGUMENTS:

level_0
[level_1]
[level_2]
[level_3]

> The four possible priority levels at which communication lines can interrupt the central processor. At least one priority level must be specified. Values for level_0 through level_3 must be in the range of 6 through 60 and must not duplicate any other communications device priority levels. Each value chosen for level_0 through level_3 must be greater or equal (have a lower or equal priority) to the preceding level. The default values for level_1 through level_3 are equal to the value assigned to the preceding priority level (level_(n-1)). For example, if a priority value has not been assigned for level_2 in a COMM directive, the priority value specified for level_1 is assumed.

DESCRIPTION:

The COMM directive specifies from one to four priority levels at which communications lines can interrupt the central processor, as described above.

The interrupt level for a communications line is specified by the values of bits 14 and 15 of the channel number argument in the directive that identifies the line. Bits 14 and 15 correspond to the positions of the level_n arguments specified in the COMM directive.

Example:

```
COMM  10,11
ATD   20,20,X'FF00'...
BSC   21,21,X'FC01'...
BSC   22,22,X'FC81'...
```

In this example, two line protocol handlers are configured. The ATD associated with lrn 20 processes interrupts on level 10; the BSC station with lrn 21 processes interrupts on level 11, as does the BSC station with lrn 22.

DEVICE (DEVICE)

The DEVICE directive is required for a communications station only if it is to be accessed through the File System. In this case, the DEVICE directive must be paired with the appropriate directive (that is, TTY*, ASP, AXD, ATD, I3270, STD, BSC, PVE, or HASP) so that each pair contains the same lrn, level, and channel number. You can float channel numbers, if you wish. The DEVICE directive should follow related LPHDEF, STATION and LPHn directives, or related POLIST, STAPOL, ROP, DSK, STDLN, I3270, and STD directives. The DEVICE directive must not be inserted among related LPHDEF, STATION, and LPHn directives or among related POLIST, STAPOL, ROP, DSK, STDLN, I3270, and STD directives.

You must include a special format of the DEVICE directive in your CLM file if your installation uses a dual-purpose operator terminal. (Refer to "Configuring a Dual-Purpose Operator Terminal" in Section 5.)

FORMAT:

DEVICE device_unit,lrn,level,X'channel',[device_name],
[record_size], $\left[\begin{Bmatrix} B \\ N \end{Bmatrix}\right]$,,,[operator_terminal_name],
[component_id]

ARGUMENTS

device_unit

A string of up to six ASCII characters; the first three or four characters identify the type of station and the last two characters (alphanumeric) identify one specific station of that type. The permissible values of device_unit are as follows:

TTYnn
BSCnn
XBSCnn
PVEnn
ROPnn
ATDnn
ASPnn
STDnn
HASPnn
GENnn
I327nn
AXDTnn
AXDPnn
AXDFnn

The default characteristics of various types of stations are given below. Some of the default characteristics are specified by the default value of the Device Specific Words (DSWs). Certain default characteristics for any station in the configuration can be changed by use of the CLM directive STTY, which is described later in this section.

In addition, once the system has been configured, a user can override (temporarily change) certain default characteristics of a station through the use of system command STTY or the macrocall $STTY.

TTYnn

- Used with TTY*, TTY, ATD, AXD, ASP directives
- Record size: 73 bytes (including control byte)
- Device-specific word for connect/disconnect = 0
  - No autodial used
  - Hang up phone on disconnect
  - Queue abort
- Device-specific word for read/write = 0030 (hexadecimal)
  - Trailing carriage return
  - Trailing line feed
  - Echo input characters
  - Leading control byte
- Detab is ON
- Input is asynchronous
- Output is asynchronous
- Type is bidirectional

BSCnn

- Used with BSC directive
- BSC2780 protocol
- Record size: 137 bytes (including control byte)

DEVICE

- Device-specific word for connect/disconnect = 0

  - No autodial used
  - Hang up phone on disconnect
  - Leading control byte
  - Queue abort
  - Mode is single block

- Device hexadecimal word for read/write = 0

- Detab is OFF

- Input is nonbuffered synchronous

- Output is nonbuffered synchronous

- Type is unidirectional (input only between connects or output only between connects)

XBSCnn

- Used with BSC directive

- BSC3780 protocol

- Record size: 137 bytes (including control byte)

- Device-specific word for connect/disconnect = 0

  - No autodial used
  - Hang up phone on disconnect
  - Leading control byte
  - Queue abort

- Device-specific word for read/write = 0040 (hexadecimal)

- Detab is OFF

- Input is asynchronous

- Output is asynchronous

- Type is bidirectional

PVEnn

- Used with PVE directive

- Record size: 81 bytes (including control byte)

- Device-specific word for connect/disconnect = 0
  - No autodial used
  - Do not save function codes in read IORB
  - Hang up phone on disconnect
  - Queue abort on disconnect
- Device-specific word for read/write = 0
- Detab is ON
- Input is asynchronous
- Output is asynchronous
- Type is bidirectional

ROPnn

- Used with ROP directive
- Record size: 73 bytes (including control byte)
- Device-specific word = 0
  - Trailing carriage return
  - Leading control byte
  - Physical disconnect
  - Queue abort

ATDnn

- Used with TTY and ATD directives *
- Record size: 73 bytes
- Device-specific word for connect/disconnect = 0
  - No autodial used
  - Hang up phone on disconnect
  - Queue abort
- Device-specific word for read/write = 0030 (hexadecimal)
  - Echo input character
  - Trailing carriage return
  - Trailing line feed
  - Leading control byte
- Detab is ON

DEVICE

- Input is asynchronous
- Output is asynchronous
- Type is bidirectional

ASPnn

- Used with ASP or BPA directive
- Record size: 80 bytes (including control byte)
- Device-specific word for connect/disconnect = 0020

  - No autodial used
  - Hang up phone on disconnect
  - Queue abort
  - Connect in ROP mode

- Device-specific word for read/write = 0

  - Trailing carriage return
  - Leading control byte

- Detab is ON
- Input is asynchronous
- Output is asynchronous
- Type is bidirectional

STDnn

- Used with STD directive
- Record size: 81 bytes
- Detab is ON
- Device-specific word for connect/disconnect = 0103 (hexadecimal)

  - No autodial used
  - Home cursor on page overflow
  - Leading control byte
  - Logical poll interval = 1 second
  - No space suppress

- No roll
- Hardware function codes are specified in write requests
- No timeout on read requests
- Send DLE EOT (Data Link Escape; End of Transmission) to VIP7800 series on disconnect
- Hang up phone on disconnect
- Queue abort

* Device-specific word for read/write = 0010 hexadecimal

  - Carriage return at end of message
  - Line feed at end of message
  - Print one copy VIP7800 series

HASPnn

* Used with HASP directive
* Record size: 520 bytes (including control bytes)
* Device-specific word for connect and disconnect = 0

  - Queue abort
  - Hang up phone on disconnect

* Device-specific word for read/write = 0100 hexadecimal (with leading control byte)
* Detab is ON
* Input is synchronous
* Output is synchronous
* Type is bidirectional

GENnn

* Used with LPHn and STATION directives
* Record size: 73 bytes (including control byte)

DEVICE

- Device-specific word for connect/disconnect = 0

  - No autodial used
  - Hang up phone on disconnect
  - Queue abort

- Device-specific word for read/write = 0030 hexadecimal

- Detab is ON

- Input is asynchronous

- Output is asynchronous

- Type is bidirectional

I327nn

- Used with I327LN and I3270 directives

- Record size: 79 bytes

- Device-specific word for connect/disconnect = 0000

- Lrn of station must match lrn specified on an I3270 directive

- Input is asynchronous

- Output is asynchronous

- Type is bidirectional

AXDTnn

- Used with AXD directive (to define a terminal device to be used with the X-ON/X-OFF protocol)

- Default record size: 80 bytes (including control byte)

- Device-specific word for connect/disconnect = 4100 (hexadecimal)

  - Connect in TERMINAL mode of AXD
  - No autodial used
  - Process read orders synchronously
  - Do not solicit initial transfer from sender
  - Do not wait for initial X-ON from receiver
  - Hang up phone on disconnect
  - Queue abort on disconnect

- Device-specific word for read/write = 0230 hexadecimal

  - Echo input characters
  - Trailing carriage return on reads and writes
  - Trailing line feed on reads
  - Leading control byte on writes
  - No leading DELs on writes
  - No trailing EOF character on writes
  - Suppression of trailing spaces on writes
  - Breakable writes
  - No editing of control characters on writes

AXDPnn

- Used with AXD directive (to define a receive-only printer device which supports the X-ON/X-OFF protocol)

- Default record size: 80 bytes (including control byte)

- Device-specific word for connect/disconnect = 4400 (hexadecimal)

  - Connect using PRINTER mode of AXD
  - No autodial used
  - Process read orders synchronously
  - Do not solicit initial transfer
  - Do not require initial X-ON from printer
  - Hang up phone on disconnect
  - Queue abort on disconnect

- Device-specific word for writes = 0

  - Trailing carriage return
  - No trailing line feed
  - Leading control byte
  - No leading DELs
  - No trailing EOF character
  - No supression of trailing spaces
  - Breakable writes
  - No editing of control characters on writes

AXDFnn

- Used with AXD directive (to define a communication line over which bidirectional file transfer can be run)

- Default record size: 80 bytes

- Device-specific word for connect/disconnect = 4384 (hexadecimal)

  - Connect in FILETRAN mode of AXD
  - No autodial used
  - Process read orders synchronously
  - Solicit initial transfer from sender with X-ON
  - Do not wait for initial X-ON from receiver
  - Send an EOF character prior to disconnect
  - Hang up phone on disconnect
  - Queue abort on disconnect

- Device-specific word for read/write = 0800 hexadecimal

  - No echo of input characters
  - Post order End-of-Record character on writes
  - No trailing line feed on read or write
  - No leading control byte on writes
  - No leading DELs
  - No trailing EOF character
  - No supression of trailing blanks
  - Breakable writes

lrn

The logical resource number of the station identified by the AXD, ASP, ATD, I3270, STD, TTY*, BSC, PVE, HASP, LPHn, or STATION directive with which this DEVICE directive is paired. The value of lrn is an integer from 3 to 252, or 256 to 4002.

level

The priority level of the station identified by the AXD, ASP, ATD, I3270, STD, TTY*, BSC, PVE, HASP, or LPHn directive with which this DEVICE directive is paired.

The value for level is an integer from 7 through 61. It can be the same as the level specified for other communications stations, but it must be a higher number than what is specified in the COMM directive. The level specified for one or more communications stations may not be used for noncommunications devices or tasks.

X'channel'

The channel number of the station identified by the ASP, AXD, LPHn, ATD, I3270, STD, TTY*, BSC, PVE, or HASP directive with which this DEVICE directive is paired. Bits 10 through 15 of the channel number specified in a DEVICE directive should equal 0 even though they may not equal 0 in the paired directive. Bits 14 and 15 of the channel number specified in a DEVICE directive do not indicate the interrupt level of the communications line. You can float this channel assignment by specifying a single zero, if you wish.

[device_name]

A string of 1 to 12 ASCII characters, the first of which must be alphabetic. This device name is a unique name by which the station can be referenced within the File System. If a device name is not specified, the device_unit argument is used as the device name.

[record_size]

The length, in bytes, of one physical record. If record size is not specified, the default record size is as established by the device_unit argument.

$\left[\begin{Bmatrix} B \\ N \end{Bmatrix}\right]$

For normally unbuffered stations (for example, BSC or HASP), B indicates that input/output to the station is to be buffered.

For normally buffered stations (for example, ATD), N indicates that input/output to the station is to be unbuffered. *

For TTY*, PVE, ATD, LPHn, and STD stations, the default is buffered. For HASP, BSC, and XBSC stations, the default is unbuffered.

Input/output to a buffered station can be asynchronous or synchronous. Input/output to an unbuffered station is always synchronous; tabulation characters are not expanded.

For a BSC3780 (XBSC) station, input/output must be asynchronous. Therefore, this argument must not be specified as N for an XBSC station.

It is recomended that this argument be specified as N for receive-only printers on an ASP or BPA station.

Tabulation characters are normally expanded into space characters for the following station types:

- TTY*
- PVE
- ROP
- ATD
- STD
- HASP

Tabulation characters are not expanded if N is specified for these station types. Tabulation characters are never expanded for BSC or XBSC station types.

[operator_terminal_name]

Applies only if the device unit is ATDnn and if a dual-purpose operator terminal is being configured. The dual-purpose operator terminal name can be up to 12 characters.

[component_id]

Permits selected devices to share the same lrn.

To use this feature, the lrn, level, and channel arguments of the second multiplexed device must be the same as the first one. The eleventh argument of the DEVICE directive is called the "component_id." It must be chosen from one of the following:

- CRT
- ROP
- DSK
- BGR

The default value is CRT. The component id is used in conjunction with the ROP or ASP communications directive.

DESCRIPTION:

If a communications station is to be accessed through the File System interface, the appropriate communications directive must be paired with a DEVICE directive. The lrn, level, and channel numbers for each pair of directives must be identical.

Multiple DEVICE directives that specify the same lrn and level are invalid, in most cases, unless each DEVICE directive specifies a unique component id. The EQLRN directive (described later in this section) allows you to equate multiple lrns to a device.

Example 1:

```
ATD 21,8,X'FF80',,300
DEVICE ATD00,21,8,X'FF80',ATDFILE
```

In this example, a terminal is to be accessed through the File System. The DEVICE directive specifies the same lrn, level, and channel number as the ATD directive. The default characteristics of the station are shown under the device_unit argument above. The default record size (73 bytes) is also to be used since no record_size argument is specified in the DEVICE directive. The device name ATDFILE is to be used for references to the station within the File System.

Example 2:

```
ASP 15,17,X'F800',0,9600,'7270'
DEVICE ASP00,15,17,X'F800',ASPFILE
```

In this example, a serial printer is to be accessed through the File System. The printer is an ASPI-38 with an RS-422 interface. The DEVICE directive specifies the same lrn, level, and channel number as the ASP directive. The device name ASPFILE is to be used for references within the File System.

Example 3:

```
.
.
.
STDLN 10,X'FF00',2,2400,W4
STD 20,0,,V7805
ROP ROSY26
STD 21,1,,V7804
STD 22,2,,V7760
POLIST 2
STAPOL 0,1,0,2
DEVICE STD00,20,10,X'FF00',V7805,80,B
DEVICE STD01,20,10,X'FF00',ROSY,80,N,,,,ROP
DEVICE STD02,21,10,X'FF00',V7804,80,B
DEVICE STD03,22,10,X'FF00',V7700,80,B
.
.
.
```

In this example, four synchronous devices are to be accessed through the File System interface. DEVICE STD00, DEVICE STD02, and DEVICE STD03 directives are paired with the first, second, and third STD directives in the CLM file. DEVICE STD01 is paired with the ROP directive in the same CLM file.

Note that each DEVICE directive has the same lrn, level, and channel number as specified in its paired STD directive. Also note, that the device directives STD00 and STD01 have the same lrn, level, and channel number. The directives are distinct in that STD00 has a default component id of CRT and that STD01 has an explicit component id of ROP.

Example 4:

```
AXD 16,14,X'C000',1,1200
DEVICE AXDF00,16,14,X'C000',TRANLINE,1024
STTY TRANLINE -DSW1 5100 -DSW2 0820
```

In this example, a communication line controlled by the X-ON/X-OFF protocol is to be accessed through the file system. It is a full duplex line connected to a modem with a speed of 1200 bits per second. The line is used by an application program using file system calls to transfer files to or from another DPS 6 over the telephone line. The DEVICE directive specifies the same lrn, level, and channel number as the AXD directive. The record size is 1024 bytes. The device name TRANSLINE is to be used for references to the line within the file system.

The STTY directive in this example specifies that AXD supports multiple asynchronous read orders issued by the application and that received characters are echoed back to the other side for verification.

VTS7760/7740 DISKETTE (DSK)

The DSK directive specifies that a VTS7760/7740 diskette is connected to a station on a line serviced by the STD line protocol handler. See the STDLN directive later in this section for information on how to specify DSK directives in a CLM file.

FORMAT:

```
DSK
```

EQUATE LRN (EQLRN)

The EQLRN directive allows you to specify multiple logical resource numbers (lrns) for the same physical device. Although each device in a communications configuration must be assigned a unique lrn, the EQLRN directive permits you to "equate" two or more lrns to reference the same physical device.

FORMAT:

```
EQLRN lrn,secondary_lrn_1[,secondary_lrn_2,...]
```

ARGUMENTS:

lrn

The primary lrn associated with the physical device. This argument must be specified.

secondary_lrn_i

The secondary lrn(s) associated with the device being referenced by lrn.

DESCRIPTION:

The following example illustrates one possible application of the EQLRN directive. A communications device can be accessed through the File System as two independent files with different characteristics (such as different buffer sizes and different types of buffering). Each file is referenced by a unique file name. This requires two separate communications DEVICE directives, each specifying a unique file name. The device characteristics are specified by STTY directives.

The following set of CLM directives illustrates those directives that must be included at configuration time:

```
    .
    .
    .
COMM 9
LPH3 32,32,X'FF80',,4800
DEVICE ATD05,32,32,X'FF80',CDR00,80
STTY CDR00,,X'0C00',,S
DEVICE ATD06,33,32,X'FF80',CRP00,80
STTY CRP00,,X'0800',OFF
EQLRN 32,33
    .
    .
```

HASP (HASP)

The HASP directive identifies an IBM workstation on a line serviced by the HASP line protocol handler. If the system is to be accessed through the File System interface, an appropriate DEVICE directive must be paired with the HASP directive. If input/output to the station is to be asynchronous, the B (buffered) argument must be included in the DEVICE directive.

FORMAT:

```
HASP lrn,level,X'channel',[modem]
```

ARGUMENTS:

lrn

The logical resource number associated with the station. The value for lrn is an integer from 3 through 252. A program can use this number to identify the station when it requests an input/output operation to the station.

level

The priority level at which the line protocol handler processes requests for an input/output operation to the station. The value for level is an integer from 7 through 61. It can be the same as the level specified for other communications stations, but it must be a higher number than the communications interrupt level(s) specified in the COMM directive. The level specified for one or more communications stations must not be used for noncommunications devices or tasks.

X'channel'

A four-digit hexadecimal number (from X'080n' to X'FF8n'), specifying the channel number of the station and the interrupt priority level (n) of the line. The channel number has the following format:

Bits 0 through 9 - The 10-bit channel address of the line.

Bits 10 through 13 - Must be set to zero.

Bits 14 and 15 - Specify n, the priority level at which a communications line interrupts the central processor. n can have a value of 0, 1, 2, or 3 as specified in the COMM directive. (See the description of the COMM directive for further information.)

This directive does not support floating channel numbers.

[modem]

A one-digit number that specifies modem type: what modem signals are set to establish a data-set connection, what signals are required for a connection, and what signals for a disconnection. Possible values are: 0, 2, 4, 6, 8. The default value is 2. For a description of modem types, see Table 6-3.

H3270 (H3270)

The H3270 directive identifies an emulated station on a line serviced by the BSC3270 line protocol handler. Only one station can be configured on a line.

FORMAT:

```
H3270 lrn,level,X'channel',[modem],X'poll_address',
      X'select_address'
```

ARGUMENTS:

lrn

The logical resource number associated with the station. The value for lrn is an integer from 3 through 252. A program can use this number to identify the station when it requests an input/output operation to the station.

level

The priority level at which the line protocol handler processes requests for an input/output operation to the station. The value for level is an integer from 7 through 61. It can be the same as the level specified for other communications stations, but it must be a higher number than the communications interrupt level(s) specified in the COMM directive. The level specified for one or more communications stations must not be used for noncommunications devices or tasks.

X'channel'

A four-digit hexadecimal number (from X'080n' to X'FF8n'), specifying the channel number of the station and the interrupt priority level (n) of the line. The channel number has the following format:

Bits 0 through 9 - The 10-bit channel address of the line.

Bits 10 through 13 - Must be set to zero.

Bits 14 and 15 - Specify n, the priority level at which a communications line interrupts the central processor. n can have a value of 0, 1, 2, or 3 as specified in the COMM directive. (See the description of the COMM directive for further information.)

This directive does not support floating channel numbers.

[modem]

A one-digit number that specifies modem type: what modem signals are set to establish a data set connection, what signals are required for a connection, and what signals for a disconnection. Possible values are: 0, 2, 4, 6, 8. The default value is 2. For a description of modem types, see Table 6-3.

X'poll_address'

A two-digit hexadecimal number (from X'00' to X'FF') specifying the poll address of the 3270 control unit.

X'select_address'

A two-digit hexadecimal number (from X'00' to X'FF') specifying the select address of the 3270 control unit.

Example:

```
H3270 25,20,X'FC00',2,X'40',X'60'
```

In this example, the host uses an address of X'40' to poll the station and an address of X'60' to select the station.

I327LN (I327LN)

The I327LN directive identifies an IBM 3270 terminal link serviced by the BSC3270 terminal facility (BTF).

FORMAT:

```
I327LN master_lrn,level,X'channel',[modem],translation
       option,[size of dedicated buffer]
```

ARGUMENTS:

master_lrn

The logical resource number of the master lrn. There are many devices on one line and the master lrn controls the whole line. The value for the master lrn is an integer from 3 to 252.

level

The message interrupt priority level at which the 3270 line protocol handler processes requests for an input/output operation to the station. The value for level is an integer from 7 through 55. It can be the same as the level specified for other communications stations, but it must be a higher number than the communications interrupt level(s) specified in the COMM directive. The level specified for one or more communications stations must not be used for noncommunications devices or tasks.

X'channel'

A four-digit hexadecimal number (from X'080n' to X'FF8n'), specifying the channel number of the station and the interrupt priority level (n) of the line. The channel number has the following format:

Bits 0 through 9 - The 10-bit channel address of the line.

Bits 10 through 13 - Must be set to zero.

Bits 14 and 15 - Specify n, the priority level at which a communications line interrupts the central processor. n can have a value of 0, 1, 2, or 3 as specified in the COMM directive. (See the description of the COMM directive for further information.)

[modem]

A one-digit number that specifies modem type: what modem signals are set to establish a data set connection, what signals are required for a connection, and what signals for a disconnection. Possible values are: 0, 2, 4, 6, 8, or 9 through 15. The default value is 2. For a description of modem types, see Table 6-3.

translation option

Valid values are:

AS - All data except exempt types is translated to/from EBCDIC from/to ASCII by the channel control program.

EB - All data is passed through untranslated, i.e., the user's application sends/receives EBCDIC data.

AE - Data is mixed ASCII and EBCDIC, determined on a per order basis. AE is required for the DSA/SNA Gateway.

There is no default for this parameter. Overall efficiency is reduced when AE is selected.

[size of dedicated buffer]

Size of channel table dedicated buffer in characters. The default and minimum value is 2048.

Example:

```
I327LN 192,14,X'F680',2,AE,2049
I3270 193,X'40',X'40','3278'
I3270 194,X'40',X'C1','3278'
I3270 195,X'40',X'C2','3287'
DEVICE I3270,193,14,X'F680',,,N
DEVICE I3271,194,14,X'F680',,,N
DEVICE I3272,195,14,X'F680',,,N
```

In this example, there are two CRTs and one printer configured on the 3270 host link. Data is mixed ASCII and EBCDIC on a per order basis.

I3270 (I3270)

The I3270 directive configures the supported devices on the IBM 3270 host link.

FORMAT:

```
I3270 lrn,control_unit,device_address,'device_type'[,rows,cols]
```

ARGUMENTS:

lrn

The logical resource number associated with the station. The value for lrn is an integer from 3 to 252. A program can use this number to identify the station when it requests an input/output operation to the station.

control_unit

The Polling address of the destination IBM control unit.

device_address

The address of the device on the IBM control unit.

'device_type'

The device type of the station. Valid values are:

3277 - 3277 CRT
3278 - 3278 CRT
3279 - 3279 CRT
3287 - 3287 printer.

There is no default for this parameter, it must be specified.

[rows,cols]

The number of rows and columns on the CRT. The valid values for the rows are:

12 (3277, 3278 only)	Indicates 3277/8 model 1
24 (3277, 3278, 3279 only)	Indicates 3277/8/9 model 2
27 (3278 only)	Indicates 3278 model 5
32 (3278, 3279 only)	Indicates 3278 model 3 or 3279 model 2
43 (3278 only)	Indicates 3278 model 4.

The default value for rows is 24.

The valid values for the columns are:

40 (3277 only)	Indicates 3277/8 model 1
80 (3277, 3278, 3279 only)	Indicates 3277 model 2, 3279, or 3278 models 1, 2, 3, or 4
132 (3278, 3287 only)	Indicates 3278 model 5 or 3287.

The default value for columns is 80 for the 3277, 3278, and 3279 CRTs. For the 3287 CRT, the columns default value is 132.

Example:

```
I327LN 148,14,X'F600',2,AS,2048
I3270 149,X'40',X'40','3278'
I3270 140,X'40',X'C1','3278'
I3270 141,X'40',X'C2','3287'
DEVICE I32720,149,14,X'F600',,,N
DEVICE I32721,140,14,X'F600',,,N
DEVICE I32722,141,14,X'F600',,,N
```

In this example, there are two 3278 CRTs and one 3287 printer configured on the 3270 host link.

LINE PROTOCOL HANDLER (LPHn)

The LPHn directive identifies the first (or only) station on a line serviced by a user-written line protocol handler.

FORMAT:

```
LPHn lrn,level,X'channel',[modem],[speed],
     [FDX/HDX][,lph_specific_word]
```

In the directive name LPHn, n is an integer from 0 through 3 or from 32 through 47 and identifies a specific line protocol handler. If an LPHDEF directive is used in association with this line protocol handler, the value of the LPH argument in the LPHDEF directive must match n.

ARGUMENTS:

lrn

The logical resource number associated with the station. The value for lrn is an integer from 3 to 252, or 256 to 4002. A program can use this number to identify the station when it requests an input/output operation to the station.

level

The priority level at which the line protocol handler processes requests for an input/output operation to the station. The value for level is an integer from 7 through 61. It can be the same as the level specified for other communications stations, but it must be a higher number than the communications interrupt level(s) specified in the COMM directive. The level specified for one or more communications stations must not be used for noncommunications devices or tasks.

X'channel'

A four-digit hexadecimal number (from X'080n' to X'FF8n') specifying the channel number of the station and the interrupt priority level (n) of the line. The channel number has the following format:

Bits 0 through 9 - The 10-bit channel address of the line.

Bits 10 through 13 - Must be set to zero.

Bits 14 and 15 - Specify n, the priority level at which a communications line interrupts the central processor. n can have a value of 0,1,2, or 3 as specified in the COMM directive. (See the description of the COMM directive for further information.)

This directive does not support floating channel numbers.

[modem]

A one-digit number that specifies modem type: what modem signals are set to establish a data-set connection, what signals are required for a connection, and what signals for a disconnection. Possible values are: 0, 1, 2, 3, 4, 5, 6, 7, 8 or 9 through 15. The default value is 2. For a description of modem types, see Table 6-3.

[speed]

The data rate in bits per second.

For an asynchronous line with a communications-pac whose id is hexadecimal 2108, use one of the following values for speed:

	50	300	2400
	75	600	3600
(default)	110	900	4800
	134	1200	7200
	150	1800	9600

If the data rate is 134.5, specify 134.

For an asynchronous line with a communications-pac whose id is hexadecimal 2110, 2118, 2178, 2978, 3518, 3519, 3618 or 3619, use one of the following values for speed:

	50	200	1800
	75	300	2000
(default)	110	600	2400
	134	1050	4800
	150	1200	9600

If the data rate is 134.5, specify 134.

If the adapter is one of the adapter types in the second set, you have the option to defer selection of the line speed until the terminal comes online. You select this option by specifying HI or LO.

'HI'

All terminals associated with this line are permitted to function only at speeds from 1200 through 9600 bits per second.

'LO'

All terminals associated with this line are permitted to function only at speeds from 110 through 1200 bits per second.

For more information about the line speed selection capability, refer to the discussion under "Topics Related to CLM Directives" at the beginning of this section.

For a synchronous line, this parameter can be omitted; the communications adapter directly determines the line speed from the data set or from a timing source within the communications controller.

[FDX/HDX]

Specifies if the line is full- or half-duplex (HDX). If it is full-duplex (FDX), two channel tables are assigned. The default value is HDX.

[lph_specific_word]

A word containing user-defined information passed to the line protocol handler through the station table at offset ZQSSTS. The default is zero.

DESCRIPTION:

The LPHn directive must be included once for each line on which there are communication stations to be driven by a user-written line protocol handler. An LDBU directive (described in Section 5) must be included among the CLM directives so that the CLM loads the user-written line protocol handler bound unit and executes its initialization code. If the sizes of the channel and station tables are different from the default sizes for these tables, an LPHDEF directive must be included before the related LPHn directive(s). The values specified in the LPHDEF directive apply only to the LPHn and STATION directives that immediately follow the LPHDEF directive.

If there is more than one station on a line driven by the user-written line protocol handler, the additional stations on the line must be identified by STATION directives that immediately follow the LPHn directive.

Example:

```
LPH0 27,8,X'FD80',,,FDX
STATION 28,1
```

In this example, there are two stations on a synchronous, full-duplex line controlled by a user-written line protocol handler.

LINE PROTOCOL HANDLER DEFINITION (LPHDEF)

For each user-written line protocol handler, you can include an LPHDEF directive to define the extension sizes of the tables used for the channels and stations controlled by the line protocol handler. If the LPHDEF directive is not included, the default channel table and station table sizes are used.

The values specified for channel_table_ext and station_table_ext apply only to the LPHn and STATION directives that immediately follow the LPHDEF directive.

FORMAT:

```
LPHDEF lph,[channel_table_ext] [,station_table_ext]
```

ARGUMENTS:

lph

An integer from 0 through 3, or from 32 through 47 that associates this LPHDEF directive with a line protocol handler identified in an LPHn directive.

[channel_table_ext]

Specifies the number of words required for an extension to the basic channel table. The default value is zero, to indicate that the channel table is of standard length with no extension.

[station_table_ext]

Specifies the number of words required for an extension to the basic station table. The default value is zero, to indicate that the station table is of standard length with no extension.

Example:

```
LPHDEF 0,30
LPHO 27,8,X'FD80',,,FDX
STATION 28,1
```

In this example, line FD80 has two stations on a synchronous, full-duplex line controlled by a user-written line protocol handler. Each of the two channel tables for the line has an extension of 30 words, as defined by the channel_table_ext argument in the LPHDEF directive. The default value is used for station_table_ext.

MODEM (MODEM)

The MODEM directive defines a nonstandard modem type. The information provided in this directive is used to establish criteria for the initiation of a communications connection and the tests for the determination of a communications connection or disconnection.

FORMAT:

```
MODEM type_number,connection_AND_mask,connection_XOR_mask,
      disconnection_AND_mask,disconnection_XOR_mask,
      data_set_control
```

ARGUMENTS:

type_number

An integer from 9 to 15 that is assigned to this modem definition and can then be used in a communications station directive (that is, TTY*, ATD, STD, BSC, HASP, PVE, and LPHn directives). *

connection_AND_mask

A two-digit hexadecimal number whose value governs which bits (that is, from 0 through 3) of line register 5 (LR5) are examined when a connect request is processed.

connection_XOR_mask

A two-digit hexadecimal number whose value governs which bits (from 0 through 3) of LR5 must be ON (that is, set to 1) for a connection.

disconnection_AND_mask

A two-digit hexadecimal number whose value governs which bits (that is, from 0 through 3) of LR5 are examined when a disconnect request is processed or when a test for the occurrence of a disconnect is made.

disconnection_XOR_mask

A two-digit hexadecimal number whose value governs which bits (from 0 through 3) must be ON (that is, set to 1) for a disconnection.

data_set_control

A two-digit hexadecimal number. Bits 0 through 4 and bits 6 and 7 are loaded unconditionally into the respective bits of Line Control Table (LCT) 20. The contents of LCT byte 20 are loaded into line register 2 of the communications-pac when a line is to be connected. Standard modem types 3, 4, 7, and 8 have a connect feature for some European data sets that require data-terminal-ready to remain low during auto answer until the ring indicator is turned on. This feature is activated by setting bit 7 of the data_set_control argument. Further, on auto call operations, the setting of bit 6 causes the ACU to terminate a dialed number with an End Of Number (EON) symbol.

NOTES

1. To test for a successful connection, the contents of LR5 are first ANDed against the (user-supplied) connection_AND_mask. Then a logical exclusive OR operation is performed on the result of the first operation, against the (user-supplied) connection_XOR_mask. If the result is zero, a connection has been established.

2. To test for a disconnect, the same operations are carried out using the analogous disconnection masks. A zero result indicates a disconnection.

3. The following shows the mask and data set control values for the standard CLM-recognized modem types:

Adapter Type	Modem Type	CONNECT AND	Masks XOR	DISCONNECT AND	Masks XOR	Data Set Control
ASYNC	0	X'80'	X'80'	X'80'	X'00'	X'80'
SYNC	0	X'80'	X'80'	X'80'	X'00'	X'88'
ASYNC	1	X'A0'	X'A0'	X'A0'	X'00'	X'80'
SYNC	2	X'80'	X'80'	X'80'	X'00'	X'80'
ASYNC	3	X'A0'	X'A0'	X'A0'	X'00'	X'81'
SYNC	4	X'80'	X'80'	X'80'	X'00'	X'81'
ASYNC	5	X'A0'	X'A0'	X'A0'	X'00'	X'82'
SYNC	6	X'80'	X'80'	X'80'	X'00'	X'82'
ASYNC	7	X'A0'	X'A0'	X'A0'	X'00'	X'83'
SYNC	8	X'80'	X'80'	X'80'	X'00'	X'83'

4. The following shows line register 5 and LCT byte 20. RSU stands for "reserved for system use."

Line Register 5:

0	1	2	3	4	5	6	7
DATA SET STATUS			COMMUNICATIONS-PAC STATUS				
DATA SET READY	RSU	CARRIER DETECT INDICATOR	RING INDICATOR	RSU	RSU	RSU	RSU

LCT Byte 20:

0	1	2	3	4	5	6	7
DATA SET STATUS				COMMUNICATIONS-PAC STATUS			
DATA TERMINAL READY	RSU	RSU	RSU	SYNCHRONOUS DIRECT CONNECT	RSU	ACU EON SUPPORT	MONITOR FOR RING INDICATOR

Example:

```
MODEM 9,X'20',X'20',X'20',X'00',X'88'
```

In this example, a modem type requiring only the carrier-detect signal for a connection and absence of this signal for a disconnection is defined. Note also that a connection request causes both Data Terminal Ready (DTR) and the synchronous direct connect signal to be set high.

POLL LIST (POLIST)

POLIST specifies the time interval between successive scans of the poll list. The POLIST directive must precede the STAPOL directive.

FORMAT:

POLIST [poll_cycle_delay]

ARGUMENT:

[poll_cycle_delay]

Defines the time interval in seconds between successive scans of the poll list. After completing a scan of the poll list, the LPH waits the specified time interval before rescanning the poll list. If this argument is specified, it must be in the range of 1 to 10 seconds. If not specified, the default is 1 second.

Refer to the description of the STDLN directive for examples of the POLIST directive.

POLLED VIP EMULATOR STATION (PVE)

The PVE directive identifies a VIP emulated station.

FORMAT:

```
PVE lrn,poll_address[,poll_response][,controller_poll_address]
```

ARGUMENTS:

lrn

The logical resource number associated with the station. The value for lrn is an integer from 3 to 252. A program can use this number to identify the station when it requests an input/output operation to the station.

poll_address

The poll address of the station on the line. The poll address argument is an integer from 0 through 31. Each station on the line must have a unique poll address.

[poll_response]

The type of response that PVE generates if a select and/or poll is received for a station that is not connected. The possible values are:

QA - If an unconnected station is selected to receive data, PVE responds to the subsequent poll with a positive acknowledgement (ACK).

If an unconnected station is simply polled, PVE responds with a quiescent (Q) frame.

QN - If an unconnected station is selected to receive data, PVE responds to the subsequent poll with a negative acknowledgement (NAK).

If an unconnected station is simply polled, PVE responds with a quiescent (Q) frame.

The default is no response to a poll or select.

[controller_poll_address]

The controller poll address associated with this station on the line. The address is equivalent to the poll address of the VIP7760 controller to which this station is connected. The station emulates a VIP7760 station having the same controller poll address. Controller poll address must be an integer in the range 0 through 7. Several stations on the line can have the same controller poll address.

The default is to assign no controller poll address to this station.

DESCRIPTION:

PVE directives are used in configuring the system where polled VIP terminal emulation is to be performed. The system at the other end of the communications line must be configured to support the polled VIP emulated devices.

A single communications line can have up to 32 polled VIP stations. Up to 32 stations can be combined in groups of eight when emulating a VIP7760 on a line. All stations in a group must specify the same controller poll address.

If the station is accessed through the File System interface, an appropriate DEVICE directive must be paired with the PVE directive.

Example:

```
PVELN 12,X'FD80',0,,8
PVE 30,0,QA
PVE 31,1,QA
PVE 32,2,QA
```

In this example, three polled VIP emulated stations are defined for the communications line FD80. Each station has a unique lrn and poll address. The modem type 2 is used.

POLLED VIP EMULATOR LINE (PVELN)

The PVELN directive identifies a line serviced by the PVE line protocol.

FORMAT:

```
PVELN level,X'channel',[modem][,buffer_size][,data_bits]
```

ARGUMENTS:

level

The priority level at which the line protocol handler processes requests for an input/output operation to the station. The value for level is an integer from 7 through 61. It can be the same as the level specified for other communications stations, but it must be a higher number than the communications interrupt level(s) specified in the COMM directive. The level specified for one or more communication stations must not be used for noncommunication devices or tasks.

X'channel'

A four-digit hexadecimal number (from X'080n' to X'FF8n'), specifying the channel number of the station and the interrupt priority level (n) of the line. The channel number has the following format:

Bits 0 through 9 - The 10-bit channel address of the line.

Bits 10 through 13 - Must be set to zero.

Bits 14 and 15 - Specify n, the priority level at which a communications line interrupts the central processor. n can have a value of 0, 1, 2, or 3 as specified in the COMM directive. (See the description of the COMM directive for further information.)

This directive does not support floating channel numbers.

PVELN

[modem]

A one-digit number that specifies modem type: what modem signals are set to establish a data set connection, what signals are required for a connection, and what signals for a disconnection. Possible values are: 0, 2, 4, 6, 8, or 9 through 15. The default value is 2. For a description of modem types, see Table 6-3.

[buffer_size]

The size of the dedicated buffer in bytes. The PVE line protocol handler uses 3 dedicated buffers to process incoming messages. This parameter determines the size of each of the dedicated buffers. For optimum performance, a value should be selected which is no less than one third of the size of the largest record it is expected to read. Excessively large values do not improve performance. If this value is specified for more than one PVE directive for a single channel, the largest specified value is used. The default value is 200 bytes.

[data_bits]

The number of data bits that can be transmitted on this communication link. A value of 7 or 8 can be chosen for each device. The default value is 7.

If a channel that is configured for 7-bit operation receives a byte of data with the high-order bit set to one, the high-order bit is unconditionally reset to zero.

DESCRIPTION:

Refer to the description of the PVE directive for an example of the PVELN directive.

RECEIVE-ONLY PRINTER (ROP)

The ROP directive specifies that an ROP is connected to a station on a line serviced by the STD line protocol handler. See the STDLN directive for information on how to specify ROP directives in a CLM file.

FORMAT:

ROP 'rop_type'

ARGUMENT:

'rop_type'

The type of ROP device that is connected to the station. Possible values for this argument are:

rop type	Corresponding Device Type
1003	PRU1003
1005	PRU1005
7070	PRU7070, PRU7071
7075	PRU7075, PRU7076
TN300	TermiNet 300
TN1200	TermiNet 1200

One value for rop_type must be specified.

Refer to the description of the STDLN directive for examples of the ROP directive.

STATION POLL (STAPOL)

The STAPOL directive defines the order in which stations are polled on a line serviced by the STD line protocol handler. The order in which stations are polled is determined by the positions of arguments in the STAPOL directive.

FORMAT:

```
STAPOL station_poll_address_1,[,station_poll_address_2]...
     [,station_poll_address_15]
```

ARGUMENT:

station_poll_address_i

Specifies the poll address of a station on this line. Up to 15 station poll addresses can be specified per STAPOL directive. Multiple STAPOL directives can be used for the same STD line, for example:

```
.
.
.
STDLN 10,X'FC00',2,4800
STD 35,0,,'V7700'
STD 36,1,,'V7804'
STD 37,2,,'V7804'
STD 38,3,,'V7814S'
POLIST 1
STAPOL 0,1
STAPOL 2,3
.
.
.
```

The station poll addresses must have been previously specified in an STD directive associated with this line. The value of this argument must be in the range 0 through 31. The poll address can be specified as many times as is necessary to create the required polling priority on the line.

Refer to the description of the STDLN directive for examples of the STAPOL directive.

STATION (STATION)

The STATION directive identifies the second or subsequent station(s) on a line controlled by a user-written line protocol handler that supports multiple stations on a line. One station on the line must be identified by a LPHn directive. Additional stations are identified by STATION directives, one per station, immediately following the related LPHn directive. STATION directives and LPHn directives must immediately follow the LPHDEF directive in the CLM file.

Note that the priority level, channel number, modem type, line speed, and line type (FDX/HDX) parameters are obtained from the LPHn directive that precedes the STATION directive.

FORMAT:

```
STATION lrn[,lph_specific_word]
```

ARGUMENTS:

lrn

The logical resource number associated with the station. The value for lrn is an integer from 3 to 252, or 256 to 4002. A program can use this number to identify the station when it requests an input/output operation to the station.

[lph_specific_word]

Specifies a word containing user-defined information that is to be passed to the line protocol handler through the station table, at offset ZQSSTS. The default is 0.

Refer to the description of the LPHn directive for an example of the STATION directive.

SYNCHRONOUS TERMINAL DEVICE (STD)

The STD directive identifies a synchronous terminal or station on a line serviced by the Synchronous Terminal Driver (STD) line protocol handler.

FORMAT:

```
STD lrn,station_poll_address[,,'device_type'][,PB][,PVE]
```

ARGUMENTS:

lrn

The logical resource number associated with the station. The value for lrn is an integer from 3 to 252, or 256 to 4002. A program can use this number to identify the station when it requests an input/output operation to the station.

station_poll_address

The poll address of this station. The poll address is an integer from 0 through 31. Each station on the line must have a unique station poll address.

['device_type']

The type of supported terminal. Possible values for device_type are:

device_type	Corresponding Terminal Type
*	A self-configuring device
7814S	VIP7814/VIP7815 (synchronous protocol)
7824S	VIP7824/VIP7825 (synchronous protocol)
7816S	VIP7816/VIP7817 (synchronous protocol)
7826S	VIP7826/VIP7827 (synchronous protocol)
771626	VIP7826/VIP7827 in VIP7700 mode
V7804	VIP7804/VIP7805
V7700	VIP7700/VIP7700R/VIP7705R
V7760	VIP7760
V7740	VTS7740
V7710	VTS7710
* 1901	TWU1901

If this argument is not specified, the default device type is V7700.

For more information about self-configuring devices, refer to the discussion under "Topics Related to Communications Directives" at the beginning of this section.

[PB]

Process Before Verify: allows for a buffer length greater than 1024. This argument applies to the VIP7800 family of synchronous terminals. The default is Verify Before Process. The buffer range on writes is limited to 1023 bytes.

[PVE]

Indicates that the station addressed by this address is not a real terminal, but an emulated station serviced by the PVE line protocol handler.

DESCRIPTION:

When using the Synchronous Terminal Driver (STD) line protocol handler, you must observe the following guidelines:

- The master lrn station must be the last station to be disconnected.
- An STD directive must be specified for every station that is supported on the line.
- All station poll addresses must be unique.
- Device types can be mixed on a single line.

Refer to the description of the STDLN directive for examples of the STD directive.

SYNCHRONOUS TERMINAL DEVICE LINE (STDLN)

The STDLN directive identifies a line serviced by the STD line protocol handler.

FORMAT:

```
STDLN level,X'channel',[modem],[speed][,2/4_wire]
      [,data_bits]
```

ARGUMENTS:

level

The priority level at which the line protocol handler processes requests for an input/output operation to the station. The value for level is an integer from 7 through 61. It can be the same as the level specified for other communications stations, but it must be a higher number than the communications interrupt level(s) specified in the COMM directive. The level specified for one or more communications stations must not be used for noncommunications devices or tasks.

X'channel'

A four-digit hexadecimal number (from X'080n' to X'FF8n'), specifying the channel number of the station and the interrupt priority level (n) of the line. The channel number has the following format:

Bits 0 through 9 - The 10-bit channel address of the line.

Bits 10 through 13 - Must be set to zero.

Bits 14 and 15 - Specify n, the priority level at which a communications line interrupts the central processor. n can have a value of 0, 1, 2, or 3 as specified in the COMM directive. (See the description of the COMM directive for further information.)

This directive does not support floating channel numbers.

[modem]

A one-digit number that specifies modem type: what modem signals are set to establish a data-set connection, what signals are required for a connection, and what signals for a disconnection. Possible values are: 0, 2, 4, 6, 8. The default value is 2. For a description of modem types, see Table 6-3.

[speed]

The data rate in bits per second. The default value is 2000. Other possible values for speed include:

1200
2000
2400
4800
9600

[2/4_wire]

A two- or four-wire connection. Possible values are:

(default)	W2	to specify two-wire connection
	W4	to specify four-wire connection

[data_bits]

The number of data bits that can be transmitted on this communication link. A value of 7 or 8 can be chosen for each device. The default value is 7.

If a channel that is configured for 7-bit operation receives a byte of data with the high-order bit set to one, the high-order bit is unconditionally reset to zero.

DESCRIPTION:

The STDLN directive defines a line serviced by the STD line protocol handler; it precedes all other directives that characterize this line. Configuring a line serviced by the STD line protocol handler requires that at least four different directives be specified for the line and all stations on the line. These directives must be specified according to the following guidelines:

<u>Required directives</u>

STDLN (one for each line)
STD (one for each station on the line)
POLIST (one for each line)
STAPOL (one or more as needed)

Optional directives

ROP (one per STD line)
DSK (one per STD line, only with 7760 or 7740 terminals)

Configuring a line serviced by the STD line protocol handler requires at least the directives STDLN, STD, POLIST, and STAPOL. An STD line can also use an ROP and/or DSK directive. The DSK (diskette) directive is meaningful only with 7760 or 7740 terminals. The following list illustrates the sequence rules for STD directives. Brackets enclose optional directives. One set of these directives must be specified for each line:

STDLN — Must specify for each line; all related STD, ROP, DSK, POLIST, and STAPOL directives must follow

STD — One for each station on the line

[ROP] — Optional receive-only printer associated with prior station

[DSK] — Optional 7760/7740 diskette associated with prior station

.
.
.
STD
[ROP]
[DSK]
.
.
.

POLIST — Must specify for each line

STAPOL — Must specify one or more for each line; all STAPOL directives should be specified in the order in which stations are to be polled

[DEVICE] — Specifies that the STD or ROP device can be accessed by the File System. The DSK device can only be accessed through the physical I/O (PI/O) interface.

Example 1:

```
.
.
.
STDLN 20,X'FC00',,2400,W4
STD 20,1,,'V7804'
ROP  'TN300'
STD 21,2,,'V7804'
ROP '1005'
STD 22,3,,'V7804'
POLIST 5
STAPOL 1,2,1,3
.
.
.
```

In this example, an STD line has been configured with three stations. As specified in the STDLN directive, this line's request level is 20, the channel number is FC00, the default modem used is type 2, the line speed is 2400 bits per second, and the line connection is four-wire. All three stations support VIP7804 terminals. The first station has an lrn of 20, a poll address of 1, and includes a TN300 ROP. The second station has an lrn of 21, a poll address of 2 and includes a PRU1005. The third station has an lrn of 22 and a poll address of 3. The POLIST directive specifies a 5-second delay between successive poll list scans. The STAPOL directive specifies that the station with poll address 1 (lrn 20) is polled twice as often as the other two stations, implying that this station has higher priority.

Example 2:

```
.
.
.
STDLN 10,X'FF00',2,2400,W4
STD 20,0,,'V7805'
ROP '1005'
STD 21,1,,'V7804'
STD 22,2
POLIST 2
STAPOL 0,1,0,2
DEVICE STD00,20,10,X'FF00',V7805,80,B
DEVICE STD02,20,10,X'FF00',ROP,80,N,,,,ROP
DEVICE STD03,21,10,X'FF00',V7804,80,B
DEVICE STD04,22,10,X'FF00',V7700,80,B
.
.
.
```

The above example describes an STD line connected to channel FF00, with a request level of 10. The line uses a type 2 modem at 2400 baud and is a four-wire connection. There are three VIPs on the line. The terminals are polled in the following sequence for data: 0, 1, 0, 2; after which there is a 2-second delay before they are polled for data again. Address 0 is polled twice as frequently as the other two terminals. All of the devices are configured for the File System, as shown in the DEVICE directives.

SET TERMINAL CHARACTERISTICS (STTY)

The STTY directive specifies the file characteristics of a nondisk device. The characteristics specified by this directive override characteristics specified in a DEVICE directive or any characteristics established by a previous STTY directive.

The format for this directive is the keyword format described in the Commands manual for the STTY command. The keyword format supports all of the functionality of STTY including the arguments required to identify a device as automatically reconnectable whenever a power resumption or line drop recovery occurs.

TIMEOUT (TIMEOUT)

The TIMEOUT directive permits the assignment of certain communications timeout values on a line basis.

FORMAT:

TIMEOUT [inactivity][,[connect][,disconnect]][,[read][,write]]

ARGUMENTS:

[inactivity]

The maximum number of seconds that can elapse without any communications line activity before a line disconnect situation is forced on the associated line. Permissible values are * (never), or $1 \leq$ inactivity ≤ 32767.

[connect]

The maximum number of seconds that can elapse before the post of a connect request as unsuccessful (0110 return status). Permissible values are * (never) or $1 \leq$ connect ≤ 32767. For use with RS-232C connected line.

[disconnect]

The maximum number of seconds that can elapse prior to the post of a disconnect-with-hangup request as unsuccessful (0000 return status). Permissible values are $1 \leq$ disconnect ≤ 255. For use with RS-232C connected line.

[read]

The maximum number of seconds that can elapse before the post of a read with a 0106 error. The default is 300 seconds.

[write]

The maximum number of seconds that can elapse before the post of a write with a 010B error. The default is 15 seconds for ATD ROP mode.

NOTE

Read and write TIMEOUTs are effective for ATD only.

DESCRIPTION:

A TIMEOUT directive of the form "TIMEOUT *,300,3" is implied by the presence of the COMM CLM directive. Additional TIMEOUT directives are optional. The parameters of a given TIMEOUT directive apply to the communications terminal directives that follow it. The absence of a particular parameter in a TIMEOUT directive indicates that the value of the corresponding argument in the preceding TIMEOUT directive still applies.

Example:

	Inactivity Timeout	Connect Timeout	Disconnect Timeout	Read Timeout	Write Timeout
COMM 10					
BSC 32,32,X'FC00',2	*	300	3	300	--
TIMEOUT 1800					
ATD 33,33,X'FF00',0	1800	300	3	300	--
TIMEOUT ,*,5,30,15					
ASP 34,34,X'FF00',0	1800	*	5	30	15

The effect of the default TIMEOUT directive (associated with the COMM directive) and associated secondary TIMEOUT directives are summarized in this table of TIMEOUT values.

ASYNCHRONOUS TERMINAL DEVICE (TTY)

The TTY directive identifies a station on a line serviced by the Asynchronous Terminal Driver line protocol handler. This directive is used to configure asynchronous terminals as part of the communications system. The TTY directives must follow the COMM directive in the CLM file. See the ATD directive for more information.

The old TTY directive has been renamed to TTY* to facilitate the eventual phase-out of the TTY line protocol handler. If the TTY directive is specified, the Asynchronous Terminal Driver (ATD) line protocol handler is configured into the system. See the ATD directive for further information.

TELEPRINTER (TTY*)

The TTY* directive identifies a station on a line serviced by the TTY (teleprinter) line protocol handler. The TTY* directive supports asynchronous terminal devices as teleprinter-compatible devices.

The old TTY directive has been renamed to TTY* to facilitate the eventual phase-out of the TTY line protocol handler. If the TTY directive is specified, the Asynchronous Terminal Driver (ATD) line protocol handler is configured into the system. A brief description of each of the line protocol handlers is given in the description section of the ATD directive.

FORMAT:

```
TTY* lrn,level,X'channel',[modem],[speed],['device_type']
     [,HD/FD]
```

ARGUMENTS:

lrn

The logical resource number associated with the station. The value for lrn is an integer from 3 to 252, or 256 to 4002. A program can use this number to identify the station when it requests an input/output operation to the station.

level

The priority level at which the line protocol handler processes requests for an input/output operation to the station. The value for level is an integer from 7 through 61. It can be the same as the level specified for other communications stations, but it must be a higher number than the communications interrupt level(s) specified in the COMM directive. The level specified for one or more communications stations must not be used for noncommunications devices or tasks.

X'channel'

A four-digit hexadecimal number (from X'080n' to X'FF8n'), specifying the channel number of the station and the interrupt priority level (n) of the line. The channel number has the following format:

Bits 0 through 9- The 10-bit channel address of the line.

Bits 10 through 13 - Must be set to zero.

Bits 14 and 15 - Specify n, the priority level at which a communications line interrupts the central processor. n can have a value of 0, 1, 2, or 3 as specified in the COMM directive. (See the description of the COMM directive for further information.)

[modem]

A one-digit number that specifies modem type: what modem signals are set to establish a data set connection, what signals are required for a connection, and what signals are required for a disconnection. Possible values are: 0, 1, 3, 5, 7, or 9 through 15. The default value is 1. For a description of modem types, see Table 6-3.

[speed]

The data rate in bits per second.

For an asynchronous line with a communications-pac whose id is hexadecimal 2108, use one of the following values for speed:

	50	300	2400
	75	600	3600
(default)	110	900	4800
	134	1200	7200
	150	1800	9600

If the data rate is 134.5, specify 134.

For an asynchronous line with a communications-pac whose id is hexadecimal 2110, 2118, 2178, 2978, 3518, 3519, 3618, or 3619, use one of the following values for speed:

	50	200	1800
	75	300	2000
(default)	110	600	2400
	134	1050	4800
	150	1200	9600

If the data rate is 134.5, specify 134.

If the adapter is one of the adapter types in the second set, you have the option to defer selection of the line speed until the terminal comes online. You select this option by specifying HI or LO.

'HI'

All terminals associated with this line are permitted to function only at speeds from 1200 through 9600 bits per second.

'LO'

All terminals associated with this line are permitted to function only at speeds from 110 through 1200 bits per second.

For more information about the line speed selection capability, refer to the discussion under "Topics Related to Communications Directives" at the beginning of this section.

['device_type']

Specifies the type of terminal used. If this argument is not specified, the default is TTY. Possible values are:

Value	Physical devices supported
7200	VIP7200
7801	VIP7801, VIP7802
PRU	TWU1001, TWU1003, TWU1005
TTY	An asynchronous terminal that is supported as teleprinter compatible.

[FD/HD]

Specifies if the line is full- or half-duplex (HD). If it is full-duplex (FD), two channel tables are assigned. The default value is HD.

DESCRIPTION:

The TTY* directive and the ATD directive support asynchronous terminal devices. The line protocol handler called by the TTY* directive supports the terminal only in command mode. The TTY* line protocol handler does not provide block mode support for VIP7800 devices; neither does it provide support for VIP7200, VIP7300, and VIP7800 devices in forms mode. (For these functions the ATD directive is required.) The ATD line protocol handler does not provide for transparent-read or single-character input operations as does the TTY* line protocol handler.

Example:

```
TTY* 22,8,X'FF00',0,1200,'7801'
DEVICE TTY02,22,8,X'FF00',TTY2
```

In this example, the 7801 terminal is connected by a direct cable connection and operates at 1200 bits per second. The terminal can be accessed through the File System.

TYL LPH (TYL)

The TYL directive identifies a station on a line serviced by the TYL line protocol handler. The TYL directive supports a high performance asynchronous communications driver.

FORMAT:

```
TYL lrn,level,X'channel',[modem],[speed]
```

ARGUMENTS:

lrn

The logical resource number associated with the station. The value for lrn is an integer from 3 to 252, or 256 to 4002.

level

The priority level at which the line protocol handler processes requests for an input/output operation to the station. The value for level is an integer from 7 through 61. It can be the same as the level specified for other communications stations, but it must be a higher number than the communications interrupt level(s) specified in the COMM directive. The level specified for one or more communications stations must not be used for noncommunications devices or tasks.

X'channel'

A four-digit hexadecimal number (from X'080n' to X'FF8n'), specifying the channel number of the station and the interrupt priority level (n) of the line. n can have a value of 0, 1, 2, or 3 as specified in the COMM directive. You can float this channel assignment by specifying a single zero, if you wish.

[modem]

A one-digit number that specifies modem type: what modem signals are set to establish a data set connection, what signals are required for a connection, and what signals are required for a disconnection. Possible values are: 0, 1, 3, 5, 7, or 9 through 15. The default value is 1. For a description of modem types, see Table 6-3.

[speed]

The data rate in bits per second.

For an asynchronous line with a communications-pac whose id is hexadecimal 2108, use one of the following values for speed:

	50	300	2400
	75	600	3600
(default)	110	900	4800
	134	1200	7200
	150	1800	9600

If the data rate is 134.5, specify 134.

For an asynchronous line with a communications-pac whose id is hexadecimal 2110, 2118, 2178, 2978, 3518, 3519, 3618 or 3619, use one of the following values for speed:

	50	200	1800
	75	300	2000
(default)	110	600	2400
	134	1050	4800
	150	1200	9600

VDAM (VDAM)

The VDAM directive must be specified if the Display Formatting and Control (DFC) software is used. This directive causes the DFC software component to be incorporated in the configuration. The VDAM directive must be used if you intend to support DEF-II, OASF, or the Menu Subsystem at your installation. Appendix G contains more information about DFC software.

FORMAT:

VDAM [lrn,level][,maximum_terminals]

ARGUMENTS

lrn

The logical resource number associated with the Display Formatting and Control task. The value for lrn is an integer from 3 to 252, or 256 to 4002. This value must be specified if asynchronous Display Formatting and Control processing is desired. Two free lrns are automatically reserved for Display Formatting and Control processing.

level

The priority level at which Display Formatting and Control processing operates. The value for level must be an integer from 6 through 61. This value must be specified if asynchronous processing is desired.

Either both lrn and level must be specified, or neither argument should be specified. If neither one is specified, the default is that no asynchronous orders are issued by Display Formatting and Control.

[maximum_terminals]

The maximum number of terminals that can use Display Formatting and Control processing in one task group. The default value is 10.

REMOVE THIS PAGE AND PLACE TAB FOR

TAB 7

APPENDIXES

Appendix A
STARTUP HALTS

Startup halts are classified in three categories:

- Halts related to bootstrap operation
- Error halts related to Configuration Load Manager (CLM)
- Error halts related to other aspects of system initialization.

A halt related to the bootstrap operation can have been intentionally requested, or it can reflect an error condition. A bootstrap halt is intentionally requested by setting ON bit 13 of the 16-bit (four hexadecimal digits) bootstrap channel number (see Table 2-2). In the event of this type of bootstrap halt, the following register contents are significant:

- R1 register contains 1602.
- R7 register contains bootstrap channel number.

Error halts during bootstrap result in a 16nn value in the R1 register. See the System Messages manual for a description of the 1611, 1612, and 1616 error halts.

NOTE

> If a bootstrap halt occurs with a 1616 error condition, there is a possibility that the D7 register contains no error status. To obtain status in such circumstances, select D7 prior to retrying the operation and observe D7 during bootstrap processing.

Error halts related to the Configuration Load Manager result in a 0Bnn or a 13nn value in the R1 register; usually additional information relative to the halt is available in, or through, other registers. CLM error messages are described in the System Messages manual.

Error halts related to other aspects of system initialization result in a 99nn value in the R1 register; in some cases, additional information relative to the halt is available in, or through, other registers. See the System Messages manual.

Appendix B
CONFIGURING TIMESLICING

An LDBU is automatically issued to load the time slice bound unit TSLICE. (See the Systems Concepts manual for further information concerning timeslicing.) If you want time-slicing with other than the default values, use the LDBU CLM directive as follows:

FORMAT:

```
LDBU TSLICE[,length,time,levels,base]
```

ARGUMENTS:

length

The length of a timeslice in milliseconds. A timeslice is the length of time that a task at the head of a task queue is permitted to execute before it is interrupted and demoted to the end of the queue. The default length is 100 milliseconds, with a default scan cycle of 50 milliseconds. (Refer to the description of the SYS directive in Section 5.)

time

The maximum time on level; that is, the number of consecutive time slices that a task can remain CP bound before the task is demoted to the next priority level (unless the current level is 62). The default value is 4.

levels

The number of levels that a task can be demoted; levels must be greater than or equal to zero. If levels + base is greater than 62, then levels is forced to (62 - base). The default value is 62 - base.

base

A two-digit integer specifying the base level for timeslicing, that is, the highest priority (lowest numbered level) that is subjected to timeslicing. The base level is relative to the first available user application level. For example, if you specify 0 for base, you get the first available user level. In this case, all levels from base through 62 (inclusive) are subject to timeslicing. The default is 0 (the first available user level).

Example 1:

```
LDBU TSLICE,50,1
```

In this example, timeslicing is configured for a multiuser environment such as OAS where it is used to make the system more interactive for each individual user. The length parameter assumes a scan cycle of 25 milliseconds. (Refer to the SYS directive in Section 5). Defaults are taken for the levels and base parameters.

Example 2:

```
LDBU TSLICE
```

In this example, timeslicing is configured for an environment in which it can be expected that CPU time will be evenly distributed among the tasks and that only occasionally will a task use large amounts of CPU time and thus require the timeslicing function. Defaults are taken for all parameters.

NOTE

A system in which there are few competing tasks, such as a single-user system, does not significantly benefit from the timeslicing function.

Appendix C
MINIMUM HARDWARE REQUIREMENTS

This appendix contains guidelines for the minimum requirements of a MOD 400 system. Minimum system guidelines provide base configurations from which the system builder can calculate what additional memory and peripheral devices (if any) are necessary for efficient and effective operations. These guidelines will remain in effect until the next major release of MOD 400. For a complete list of the hardware supported by MOD 400, see the Software Release Bulletin.

This release of MOD 400 is a LAF-only (Long Address Form) release. There is no SAF (Short Address Form) counterpart to this release. Either the hardware Memory Management Unit (MMU) or the Extended Memory Management Unit (EMMU) is required for this release of MOD 400.

MEMORY CONSIDERATIONS

Generally, 512K bytes is the smallest amount of memory that a MOD 400 system can be configured in that permits utilization of features such as the Menu Subsystem, User Registration, a swap memory pool, Subsystem Switcher, and other components. Depending on the system features configured and the nature of the work being done, one or possibly several users can perform application program development in 512K bytes of memory.

With the configuration of independent (type I) memory pools, and optionally a swap pool, up to 8 million words of physical memory can be configured and utilized by the system if the physical memory is present. As more memory is configured, more features can be utilized concurrently and more users can run concurrently.

DISK CONSIDERATIONS

The boot device must be a disk, not a diskette.

The amount of disk space required for the released operating system is between 6.5 and 7 megabytes; optional components (for example, DEF-II or COBOLA) require additional disk space. Therefore, users running with 5-megabyte devices (Lark 1 or cartridge disks) should plan their disk utilization thoughtfully. Two options are available to such users:

1. Divide their required Honeywell-supplied bound units and files between two devices and use the separate volume concept (system root and user root) described in the System Concepts manual.

2. Delete unused Honeywell-supplied files so that the system volume requirements drop below the boot volume capacity.

Appendix D
THE POWER RESUMPTION FACILITY

The power resumption facility allows the system execution environment to be restarted after a power interruption. To do this, the central processor must have the memory save and autorestart unit. This unit can preserve the memory image through a power failure lasting up to two hours, but it cannot preserve the state of the I/O controllers, ensure that no operational changes have been made to the mounted volumes, or preserve the contents of the communication controllers. Restoration of this system information is accomplished through the power resumption facility.

If fewer than two hours have elapsed when power is returned to the central processor, the power resumption facility performs the following actions:

- Reinitializes the I/O controllers
- Reconnects terminal devices
- Reestablishes the integrity of mounted volumes
- Restarts the communications subsystem
- Restarts the application tasks that were active at the time of the failure

* Signals a power-resumption software trap to tasks that support this trap

* Restarts Display Formatting and Control processing.

If the power remains off for more than two hours, the memory image is destroyed and the power resumption facility is disabled. The memory-on indicator contained in the Memory Save and Autorestart Unit is illuminated. The operator must manually reset this indicator before rebootstrapping the system to perform a restart. See the System User's Guide for procedures used in restarting after a system failure.

POWER RESUMPTION CONFIGURATION REQUIREMENTS

To implement the power resumption capability, you must configure the following:

* Memory Save and Autorestart Unit
* Power resumption facility
* Automatic terminal reconnect.

Note that peripheral devices (those devices not attached to a communications controller) are designated automatically reconnectable when they are configured at system building. These devices are reconnected when power is restored.

Configuring Memory Save and Autorestart Unit

For systems with a full control panel, four small rocker switches located behind the full control panel on the control panel circuit board supply configuration information to the central processor. The switch on the extreme left (when the panel is open) is the volatile memory switch. It must be set off (pushed down) to inform the system that the Memory Save and Autorestart Unit is present.

NOTE

On some control panels, the volatile memory switch is on the extreme right.

For systems equipped with SCF, the rocker switches are located on the SCF circuit board.

Configuring the Power Resumption Facility

You add the power resumption facility to your system by including the following LDBU directive in the Configuration Load Manager (CLM) file:

```
LDBU ZXPFR
```

See Section 5 for a description of the LDBU directive.

Configuring Automatic Terminal Reconnect

Each communication device (each terminal connected to a communications controller) that is to be automatically reconnected by the power resumption facility must be so designated using the STTY command with the -RECONNECT argument in your CLM file. See the description of STTY in the Commands manual.

The STTY command can be used as a CLM directive. The terminal characteristics established at CLM time become the baseline characteristics of the terminal. These baseline characteristics can be changed at run time with an STTY command used as a GCOS 6 command.

ACTIONS FOLLOWING POWER RESUMPTION

When the power resumption facility restarts the execution environment, it automatically performs a number of system functions:

The power resumption facility automatically performs the following functions:

- Restarts the device drivers, clock, and communications subsystem
- Reconnects peripheral devices
- Resets the system date and time. (The date/time clock has a separate memory save power supply; it is supported through the real-time adapter.)
- Reloads the Memory Management Unit
- Restarts all tasks that were active when the power failure occurred.

When the operator and system users are notified that power resumption has occurred, they may be required to perform certain actions. These power resumption procedures are described in the System User's Guide.

Appendix E
CONTROLLING USER ACCESS IN USER REGISTRATION SYSTEMS

The system administrator (any user whose user-id account is SYS_ADMIN) is responsible for determining the level of control exercised over both the configuration and use of the installation's MOD 400 user registration system. For example, requiring all users to supply passwords when they log in represents more control over the system than not requiring passwords.

This appendix discusses some methods that the system administrator can use to control user access to a user registration system.

TERMINALS FILE

You can specify Access Control Lists (ACLs) and/or Common Access Control Lists (CACLs) for the Terminals file. If you specify ACLs or CACLs, you must give read/write access to the account SYS_ADMIN.

You can specify ACLs and/or CACLs for any directories superior to the Terminals file.

You can restrict access to the system from a Listener-monitored terminal on a per terminal basis as follows:

- Specify a login line in the T-record for the terminal.
- Limit logins at the terminal to abbreviations.
- Limit logins at the terminal to secondary logins.

All of these limitations are imposed by including appropriate records in the Terminals file.

PROFILES FILE

You can specify ACLs and/or CACLs for the Profiles file. If you specify ACLs or CACLs, you must give the account SYS_ADMIN read/write access.

You can specify ACLs and/or CACLs for any directories superior to the Profiles file.

The generic user profiles supplied with the Honeywell-supplied Profiles file represent a window into the system for an unregistered user. These user profiles should be deleted once the actual system administrator is registered.

You can specify permissions or constraints for specific individuals who are registered with Edit Profile, such as:

- Multiuser or single-user profile
- Default login only (allows change of password with -CPW or "C" option
- Abbreviated login only
- Secondary login only ·
- Password required
- Login to Subsystem Switcher required
- Restricted from changing password.

If you restrict users by one of the above constraints to access MOD 400 through the menu subsystem, you can limit them even further by specifying menu subsystem access roles.

You can restrict a user to logging in at just one terminal, or a set of terminals, by (1) restricting his or her profile to login by abbreviation only and (2) defining appropriate abbreviations locally at the terminal or terminals desired. If you want to restrict a user in this way but still allow him or her several different logins (e.g., with different login arguments, or different account_ids representing different projects), you can define a different abbreviation character for each login.

You can specify ACLs and/or CACLs for the backup of the Profiles file (and any directories that contain it). Since it is difficult to create an initial Profiles file, you should maintain a backup version of it. Honeywell suggests that you maintain a skeletal backup version of the Profiles file named:

>>USER_REG>BASE>PROFILES

Any backup Profiles file should be in a different directory from that of the in-use Profiles file because of copy considerations for this file with alternate indices.

ACTIVATION OF LISTENUR

LISTENUR can be directed to use the Profiles file and Terminals file in a directory on a volume other than the system boot volume.

The Spawn Group or Create Group/Enter Group Request commands used to activate LISTENUR can be typed in by the system operator, or stored in a command file invoked by the system operator, or placed in the $S START_UP.EC. The less operator intervention required, the greater the administrator's control.

ACLs and/or CACLs can be specified for the command file (if any) used to activate LISTENUR.

ACLs and/or CACLs can be specified for any directories containing the command file used to activate LISTENUR.

Because of the concurrency constraints imposed by LISTENUR on the Terminals and Profiles files, the sooner LISTENUR is activated, the sooner access to these files is restricted.

OTHER CONSIDERATIONS

There can be one or several people registered as system administrators.

You can specify ACLs and or CACLs for the bound units EP, LP, and LISTENUR in the directory >>SYSLIB2.

The memory pool assignments for LISTENUR and LISTENUR-spawned groups can be the swap pool or an independent (I) memory pool.

You can increase system control by physically controlling access to some of its elements. For example, you might want to control physical access to:

- The system console, from which one can execute in group $S and be unaffected by ACL and CACL specifications
- The storage medium that contains the Profiles and Terminals files.

Once the system is executing, the person monitoring the system console can exercise supervisory control. LISTENUR reports to the console apparent attempts to breach security. These reports (along with all reports to the console) can be logged in a disk file and later printed. (Refer to the System User's Guide (CZ04) for information on the OIM log.) This log can also be made to accept "silent" entries; that is, entries that are written to it without appearing on the console. If the OIM log is active and is accepting silent entries, LISTENUR silently logs all logins and logouts. LISTENUR also keeps a count in each user's profile of the number of password failures. The system administrator must decide how to respond to reports of password failure and must arrange to be informed of them.

For system security purposes, Subsystem Switcher users should operate from terminals for which there is a "-MODES RECONNECT" attribute with the STTY directive in the CLM_USER file. Otherwise, a disconnection that would normally log a user out only returns them to the Subsystem Switcher.

Appendix F
ERROR LOGGING

An LDBU is automatically issued to load the error logging bound unit ZERRST. Error logging is a feature that allows the operator to collect memory or hardware-related error statistics for selected peripheral devices. Error logging is intended primarily as a preventive maintenance tool, allowing for early detection and correction of potential memory or hardware failures. The error statistics collected can be used by the manufacturer's field engineering personnel to monitor memory and peripheral device performance to determine if corrective actions are required. *

The system operator has the option of activating error logging through commands or macrocalls. Error logging commands and operating procedures are described in the System Maintenance Facility Administrator's Guide and the System User's Guide.

CONFIGURATION REQUIREMENTS

Error logging is incorporated automatically into your configuration by the CLM.

HARDWARE REQUIREMENTS

If you wish to support cumulative file processing, your system must have a Commercial Processor (or a Commercial Simulator).

MEMORY REQUIREMENTS

* Error logging can be run in the system task group ($S) or in a user group (for example, $H). However, the SET_ELOG command must always be run in the system task group ($S), never in a user group. Since the memory area necessary to run error logging can be quite large, error logging should not be executed from the $S group. The operator should initially create a group in a user memory pool large enough to accommodate the error logging session. (If the operator wishes to create and maintain error logging disk files, additional file space is required as well.)

Appendix G
CONFIGURING DISPLAY FORMATTING AND CONTROL

This appendix describes the requirements for configuring the Display Formatting and Control software. Unless you configure this software for your installation, forms processing is not available to users. If your installation uses the Data Entry Facility-II (DEF-II), the Office Automation System Facility (OASF), or the Menu Facility, you must configure Display Formatting and Control software. During configuration, you should have the Display Formatting and Control manual available for reference.

CONFIGURATION REQUIREMENTS

The following hardware and software requirements are necessary to configure the Display Formatting and Control software.

Hardware Requirements

To perform forms processing, your installation requires one or more of the following terminals:

- Character mode terminals supported by the ATD LPH:
  - VIP7200, VIP7205, VIP7207
  - VIP7801, VIP7802, VIP7803
  - VIP7301, VIP7307, HDS 2
  - VIP7303, VIP7305, VIP7808, VIP7813, VIP7814, VIP7824

- Block mode terminals supported in forms mode by the STD LPH:

  - VIP7804, VIP7805
  - VIP7700, VIP7705, VIP7760
  - VTS7710, VTS7740.

Software Requirements

The CLM directives required to configure the Display Formatting and Control software with communications terminals are as follows: (The directives mentioned below are fully described in Section 6.)

VDAM

> This directive must be specified if the Display Formatting and Control software is to be incorporated in the configuration. There should be only one VDAM directive in any CLM file.

COMM

> This directive is required for a system that includes communications devices. It must precede all other communications-related directives.

The terminal device directives that you specify depend on whether the device is asynchronous or synchronous as follows:

- Asynchronous devices

```
ATD lrn,level,X'channel',[modem],[speed],['Device_type'],
    [del],[stop_bit][,parity][,HD/FD][,VIPSIM]
    [,data_bits][,compression_algorithm]

DEVICE ATDnn,lrn,level,X'channel',[device_name],
       2000,N,,,[operator_terminal_name],[component_id]
```

  Each asynchronous terminal that supports forms processing must be described with an ATD directive paired with a DEVICE directive. It is recommended that asynchronous terminals be configured with a minimum line speed of 1200 baud if they support forms processing.

- Synchronous devices

```
STDLN lrn,X'channel',[modem],[speed],[2/4_wire]
STD   lrn,station_poll_address,[,,device_type]
POLIST [poll_cycle_delay]
STAPOL station_poll_address1[,station_poll_address2,...]
DEVICE STDnn,lrn,level,X'channel',[device_name],2000,N
```

Each synchronous terminal that supports forms processing must be described with an STD directive paired with a DEVICE directive. The STDLN, POLIST, and STAPOL directives define the line to which a polled terminal is attached.

You should include only one VDAM directive in any CLM file. The location of the VDAM directive in the CLM file determines which types of terminals support forms processing. The VDAM directive must follow the CLM directives that define the terminal types requiring Display Processing and Control support. Display Formatting and Control support modules are loaded only for those terminal types that are described in CLM directives that precede the VDAM directive. If no support code is loaded for a particular terminal type, no terminals of that type can support forms processing.

The following example illustrates how the placement of the VDAM directive influences forms processing support for terminals defined for system use. The following example is a partial CLM file. Asterisks (*) denote omitted material. Numbers in parentheses and arrows (---->) are for reference only (see below); they are not CLM directives.

```
(6)---->
   *
   ATD 29,8,X'FE00',0,9600,'7200'
   DEVICE ATD01,29,8,X'FE00',ATD01,2000,N
   *
(1)---->
   *
   ATD 30,8,X'FE80',0,9600,'7801'
   DEVICE ATD02,30,8,X'FE80',ATD02,2000,N
   *
(2)---->
   *
   ATD 31,8,X'FD00',0,9600,'7200'
   DEVICE ATD03,31,8,X'FD00',ADT03,2000,N
   *
(3)---->
   *
   ATD 32,8,X'FD00',0,9600,'7301'
   DEVICE ATD04,32,8,X'FD00',ADT04,2000,N
   *
(4)---->
   STDLN 33,X'FC00',0,4800,W2
   STD 33,10,,V7804,PB
   POLIST 1
   STAPOL 0
   DEVICE STD01,33,10,X'FC80',STD01,2000,N
   *
(5)---->
```

If you place the VDAM directive at position (1), forms processing support is provided for the VIP7200 devices that are configured (in this case, the devices named ATD01 and ATD03 support forms processing). If you place the VDAM directive at position (2), all VIP7200 and all VIP7801 terminals configured into the system support forms processing. If you place the VDAM directive at position (3), the result is the same as placing it at position (2). If you place the VDAM directive at position (4), all terminals configured as VIP7200s, VIP7801s, and VIP7301s support forms processing. If you place the VDAM directive at position (5), all of the device types configured here (VIP7200, VIP7801, VIP7301 and VIP7804) support forms processing.

If you place VDAM before any terminal directives at position (6), forms processing is not supported on any terminal types.

CONFIGURATION OPTIONS

In addition to the configuration requirements described above, you have the option of configuring the operator terminal to support forms processing. The device configured as a standard operator terminal in your configuration does not support forms processing. If, however, the operator terminal is configured as a dual-purpose terminal that runs alternately under operator control and user control, users can utilize the forms processing capability when the terminal is under user control (that is, when it is not functioning as an operator terminal). In this manner, it is possible to run forms processing in a one-terminal configuration. Refer to Section 6 for information on configuring a dual-purpose operator terminal.

CONFIGURATION OPTIONS

In addition to the configuration requirements described above, you have two options to consider when configuring forms processing for your installation:

1. Activating the Listener capability for terminals in your configuration
2. Configuring the operator terminal to support forms processing.

Listener performs specific operations affecting the state of a terminal. Whether you wish to activate Listener or not depends entirely on your installation's processing requirements. Refer to Section 3 for a complete description of the Listener capability.

The device configured as a standard operator terminal in your configuration does not support forms processing. If, however, the operator terminal is configured as a dual-purpose terminal that runs alternately under operator control and user control, users can utilize the forms processing capability when the terminal is under user control (that is, when it is not functioning as an operator terminal). In this manner, it is possible to run forms processing in a one-terminal configuration. Refer to Section 6 for information on configuring a dual-purpose operator terminal.

Appendix H
ASSIGNING CHANNEL NUMBERS

At system building time, you must assign channel numbers to all I/O components in your system. Channel numbers establish paths of communications by which the system requests I/O devices to perform read or write operations. Channel numbers also establish paths by which the requested data is transferred to and from main memory.

To configure peripheral and communication devices, you must know the physical channel numbers of the peripheral and communication devices that are physically attached to the system. You must specify in the CLM_USER file the device channels to be serviced by the MOD 400 operating system. You create the CLM_USER file using a text editor. This appendix provides information to help system builders to determine appropriate device channel numbers. *

PERIPHERAL DEVICE CHANNEL NUMBERS

Peripheral devices (also called noncommunications devices) are those I/O devices connected to a Mass Storage Controller (MSC), Magnetic Tape Unit (MTU), or Multiple Device Controller (MDC).

Specifying Channel Numbers

The Computer Configuration Sheet lists the channel numbers formulated by the manufacturer for your peripheral devices. It accompanies every system shipped by the manufacturer, and is intended to be kept by the customer as a permanent record. *

The channel numbers appear on the software channel assignment list of the Configuration Sheet. They are four-digit hexadecimal numbers ending in 00 or 80. You can enter channel numbers into your CLM_USER file simply by reading them from the Locator Card on the Configuration Sheet.

Floating Channel Numbers

For the peripheral devices in your system other than the operator terminal and bootstrap device, you can either specify channel numbers or have the system supply or "float" channel numbers for you. You can float channel numbers when entering CLM directives by means of the Editor.

* When floating channel numbers, you list a blank channel number for the peripheral devices that you have indicated as belonging to your system. A blank channel number appears in the CLM_USER file as 0000. At each startup, the system reads the CLM_USER file. When it encounters a blank channel number for a listed device, it locates the board and adapter that supports the type of device listed. From these coordinates, it derives the device's channel number.

The chief advantage of having the system float channel numbers is that it relieves your needing to know the channel numbers at other installations that will receive a copy of your CLM_USER file. At another installation, the system fills in the blank values in your CLM_USER file just as your own system did. You need only be certain that the other system does not include a component-type not listed in your CLM_USER file, or a greater number of some component common to both systems.

In some cases, floating channel numbers do not produce the desired results. For example, when the system locates devices in order to determine their channel number, it cannot make fine distinctions between device-types. Although, for example, the system can "recognize" a line printer, it cannot distinguish between a high-speed line printer and a low-speed line printer.

Suppose that your system includes both types of printers, each intended by you for a certain type of job. The manufacturer has assigned to the high-speed printer the channel number 1280 and to the low-speed printer the channel number 1200; the printers have been installed accordingly. If the high-speed printer that you have designated LPT01 is to receive the tasks dispatched to it, the system must associate the predetermined channel number 1280 with printer LPT01. If, however, you float channel numbers, the system is as likely to associate LPT01 with 1200 (the low-speed printer's predetermined number) as with 1280. Instead of floating channel numbers in this case, you should find out from the Configuration Sheet the high-speed printer's channel number and specify it in the CLM_USER file. *

COMMUNICATIONS DEVICE CHANNEL NUMBERS

A communications device is any device connected to a communications controller.

Floating Channel Numbers

When you create a CLM_USER file using a text editor, you can float channel numbers for selected communications devices. Guidelines for the assignment of floating communications channel numbers associated with asynchronous terminal devices are given in Section 6 of this manual. *

Specifying Channel Numbers

Channel numbers are assigned to the devices in your system before it is installed. The Customer Service Engineer who supervised the installation of your system can inform you of the channel numbers assigned to your communications devices, if you need to specify the numbers in the CLM_USER file. The Configuration Sheet also provides the information necessary for specifying communications device channel numbers. *

If you know where the cable from a specific device attaches to the chassis of your system, you can derive the device's channel number. The remainder of this appendix explains the procedure for deriving channel numbers by inspection.

DERIVING CHANNEL NUMBERS

A valid channel number consists of four hexadecimal characters ending in 00 or 80. A channel number refers to a specific port on a specific circuit board.

Circuit boards are plugged into slots on the Megabus network that runs down the back of the chassis. In this appendix, boards supporting peripheral devices and boards supporting communications devices are called controller boards and communications controllers, respectively.

On both kinds of boards, there are four ports that accept cables from devices. On a controller board, each port accepts a cable from one device; therefore, four devices and channel numbers can be associated with one board. (Because some devices require more than one port, a controller board does not always support as many as four devices.)

Communications controllers differ from controller boards in this respect: each port on an MLCP can accept two communications lines and each port on an MLC-16 can accept four communications lines. Thus, an MLCP can support up to eight communications devices and their associated channel numbers, and an MLC-16 can support up to 16.

Every board on the Megabus network has a unique address consisting of four hexadecimal characters. This hexadecimal address must not be confused with the decimal address of the Megabus slot into which the board is plugged. Slot numbers run consecutively from the top of the chassis to the bottom; slot number 4 is located physically below slot number 3. A board's hexadecimal address, however, does not necessarily correspond to its physical position on the Megabus network.

A board's hexadecimal address is established when the board is installed by two rotary switches which define the base channel number of the board.

The ports on a board are numbered in ascending order from right to left as one faces the front (the control panel side) of the cabinet. The first port on a board and a device attached to that port carries a channel number made up of the board's address followed by 00. For example, a controller board's address, indicated by its rotary switches, is hexadecimal 12. The channel address of the device attached to the board's first port is hexadecimal 1200. The channel number of each successive port is the next highest, four-character hexadecimal number ending in 00 or 80. Thus, the channel number of the second port and the device attached to it is 1280; port number 3, 1300; port number 4, 1380.

The procedure for deriving channel numbers for communications devices is similar. As mentioned above, however, each port on an MLCP accepts two communications lines and each port on an MLC-16 accepts four communications lines. Consequently, there are two devices with associated channel numbers for each MLCP port and four devices and channel numbers for each MLC-16 port. Table H-1 lists the channel numbers of the communications lines for both communications controllers. (In this example, the address of the MLCP is FC00 and that of the MLC-16 is C000.)

In summary, to derive the channel number of a peripheral device, you must be able to locate on a board the specific port to which the device is attached, and you must know the board's hexadecimal address. To derive the channel number of a communications device, you may have to know the device's line number in addition to its port location and board address.

The address of a board can be determined in several ways. Usually, the address is labeled on the board itself. The number of the board's Megabus slot is also labeled. The Configuration Sheet shows both the slot number and hexadecimal address of all boards in your system. Thus, if you know into what slot the board is plugged, you can look up the board's address on the Configuration Sheet. Finally, you can inspect the setting of a board's rotary switches. To do so, however, you must remove the board from the chassis.

Table H-1. Communications Controller Channel Numbers

Communications Controller	Port	Line Number	Channel Number
Multiline Communications Processor (MLCP)	1	0	FC00
		1	FC80
	2	2	FD00
		3	FD80
	3	4	FE00
		5	FE80
	4	6	FF00
		7	FF80
MLC-16	1	0	C000
		1	C080
		2	C100
		3	C180
	2	4	C200
		5	C280
		6	C300
		7	C380
	3	8	C400
		9	C480
		10	C500
		11	C580
	4	12	C600
		13	C680
		14	C700
		15	C780

Appendix J
GLOSSARY OF USER REGISTRATION TERMS

This appendix contains definitions of terms associated with user registration. The terms defined here are primarily of interest to system administrators. For definitions of general terms, see the System Concepts manual.

abbreviation (abbrev)

A single character that a user can type in instead of a Login command line. Listener expands the abbreviation to a Login line by consulting its Terminals file. A login abbreviation can apply to one terminal only ("local abbreviation") or to all terminals in the system ("global abbreviation"). A global abbreviation can be made near-global by using exclusion abbreviations, which locally redefine the abbreviation character as a useless login line at those terminals where the global abbreviation is not to function.

abbrev-only login

A login trait that limits the user to logging in with an abbreviation. The user's actual login line is formed by combining the arguments in an abbrev line (A-record) in the Terminals file and the default login arguments in the user's profile.

Banner Login

A means of logging in that requires the user to type in a Login command line or an abbreviation. "Banner" refers to the banner message (consisting of the message of the day and the login prompt) displayed on the screen of a terminal that is ready to accept a login line.

Banner Login terminal

A terminal on which Banner Login has been installed. See Banner Login.

comments section

An optional section of a user's profile with section id !C. The comments section allows the system administrator to associate ASCII text with an individual user.

default login arguments

The set of login arguments stored in the RE section of a user's profile. The default login arguments are combined with arguments from the Terminals File and/or arguments entered manually at login time to form the actual login line. Users can be limited to logging in with default login arguments only. If a user is permitted to specify arguments manually, any manual arguments override corresponding default arguments.

defaults-only login

A login trait that prevents the user from entering any arguments except -CPW (change password) at login time. The user's default login arguments in his/her profile form the actual login line.

direct login

A login that Listener performs on a terminal as soon as that terminal comes online. The login is controlled by the terminal's T-record in the Terminals file. Compare manual login.

Edit Profile (EP)

An interactive program used by a system administrator to register users on a user registration system and to perform other administrative functions involving the Profiles file.

EP

See Edit Profile.

EP subsystem module

A bound unit that contains code and data enabling Edit Profile to create and modify a particular type of subsystem section. A subsystem module is named EP_id, where id is the section id of the associated subsystem. For example, when the MU section of a user's profile is being processed, Edit Profile uses the EP_MU subsystem module.

Forms Login

A means of logging in under user registration that guides the user through the login process by means of forms displayed on the screen.

Forms Login terminal

A terminal on which Forms Login has been installed. See Forms Login.

language key

A two-ASCII-character identifier used as a file name suffix to provide multiple national language support. The system default language key is specified at CLM time with the system default message library pathname. For the default message library pathname, the language key is EN (English). A primary or secondary login can specify a language key using the -LKEY argument (except that multiuser profiles can specify -LKEY only as a default login argument). A user who does not specify otherwise is given the system language key.

Listener

A system component that monitors all terminals that are configured at CLM time and are listed in its Terminals file.

LISTENUR

The bound unit name of Listener. LISTENUR supports user registration and requires a Profiles file.

List Profile (LP)

A program that lists sections of a user's profile in an edited form on the user-out file. A system administrator can use LP to list any user profile. A registered user can use LP to list his or her own profile only.

login identification (login_id)

An alternate form of identification to user id; a registered user can use either the login id or the full user id when logging in to the system.

login traits

The set of characteristics specified in the RE section of a user's profile that define how the user can log in to the system.

LP

See List Profile.

LP subsystem module

A bound unit attached to List Profile containing code and data that enables List Profile to list a particular type of subsystem section. The naming convention is LP_id, where id is the section id of the associated subsystem.

manual login

A login performed by the user who types in a login line or abbreviation or fills in a login form. Compare direct login.

multiuser profile

A login trait specifying that any number of users can be logged in to the system using this profile's user id.

password

A string of ASCII characters that a user may be required to enter in order to confirm his or her identity when logging in or after a line drop. A password may be up to eight characters long; when the system is released, its minimum length is six characters.

password obsolescence

A mechanism which forces a user to choose a new password after a set period has elapsed. When the system is released, this period is ninety days.

primary login

The form of login that requests Listener to spawn a task group that has the terminal from which the login originated as its primary system file. (That is, the terminal is the initial assignment for the group's user-in, command-in, error-out, and user-out files.)

profile section

A portion of a user profile identified by a two-character section id (for example, RE, !C, MU) and serving a specific purpose. See also registration section, comments section, and subsystem section.

Profiles file

A relative file with two alternate indices that contains user profiles and subsystem declarations. The Profiles file is a fundamental part of user registration. It (or a link to it via the LK command) must be in the same directory as the Terminals file. Its name (PROFILES) and those of its two alternate indices (PROFILES_K1 and PROFILES_K2) must not be changed.

registered user

A person who can access a user registration system from a Listener-monitored terminal because a user profile bearing his or her user id exists in the Profiles file.

registration section

A required section of a user's profile. The registration section is created at registration time; it has section id RE. The RE section determines the user's login privileges and/or constraints. It can optionally contain statistics on the user's system usage.

secondary login

The form of login that requests Listener to transfer control of the user terminal to a specified task group. The task group must already exist and have an outstanding Request Terminal ($RQTML) or Request Specific Terminal ($RQSPT) monitor call that the secondary login satisfies.

secondary user

A user logged into a subsystem by means of a secondary login.

section attributes

The elements of a profile section that define the privileges, constraints, and resources assigned to the user when executing in a particular processing environment.

section identification (section_id)

Two ASCII characters that identify a profile section type.

single-user profile

A login trait specifying that only one user can be logged in to the system using this profile's user_id.

subsystem

A general purpose application-oriented facility that provides interactive users with their capabilities and view of the system. A subsystem is generally identified directly with the lead task of a task group. A subsystem can either be primary-user oriented (supporting one interactive user per task group) or secondary-user oriented (supporting multiple interactive secondary users per task group).

subsystem declaration

An entry in the Profiles file that reserves a section_id, associates the section id with a subsystem, and authorizes the subsystem to access user profile subsystem sections identified by that section id.

subsystem section

A profile section defined by a subsystem that contains user-specific subsystem information. For example, an MU (Menu) section in a user's profile contains permissions, constraints, and statistics relative to the user's use of the Menu Subsystem.

Subsystem Switcher

A menu-oriented component of the User Productivity Facility (UPF) that allows a logged-in user to switch from one subsystem to another without having to log out and log back in again. These logouts and logins are invisible to the user.

Subsystem Switcher user

A login trait that specifies that the user can only log in through the Subsystem Switcher.

system administrator

A user with account SYS_ADMIN who is responsible for managing a user registration system. Such a user controls individual user access to the system by maintaining user profiles with the Edit Profile utility.

Terminals file

A sequential file that names the terminals monitored by Listener, defines terminal-specific access constraints and direct-login lines, and defines systemwide and terminal-specific abbreviations for login lines. As the system is released, its pathname is >>USER_REG>TERMINALS, but other directory or filenames can be used; see Profiles File.

user profile

The set of sections of the Profiles file associated with a specific user_id.

user registration

A MOD 400 functionality that maintains a file of registered users that specifies their login defaults and individual access rights.

*

Appendix K
CONFIGURING THE DISK CACHE

An LDBU is automatically issued to load the disk cache bound unit ZXDCSH on the DPS 6/75 and DPS 6/95 systems that have the disk cache processor. The disk cache is a feature that allows disk I/O read requests to be processed at memory access speeds, avoiding the time constraints posed by physical disk devices. For the DPS 6/75, no memory configuration activity is required with CLM_USER, since all memory above one megaword is automatically configured as the disk cache buffer. On the DPS 6/95, you may either designate a portion of memory to be utilized as the cache buffer or take the one megaword default. If the one megaword default is not desired, use the following LDBU CLM directive as the last LDBU in your CLM_USER file:

FORMAT:

```
LDBU **ZXDCSH,D'xxxxxx'
```

ARGUMENT:

D'xxxxxx'

The hexadecimal number of words to be allocated from memory as the cache buffer. The number must be in increments of megawords.

Example:

```
LDBU **ZXDCSH,D'200000'
```

In this example, 2 megawords of memory are allocated as the disk cache buffer. The specified amount of memory becomes the disk cache buffer, starting with the upper memory limit and decrementing the specified amount.

Appendix L
CONFIGURING THE DECISION DATA 8045 CARD READER/PUNCH

If your system includes a Decision Data 8045 Card Reader/Punch, you must include the following CLM directives, with appropriate arguments, in your CLM file:

- LPHDEF
- LPHn
- LDBU
- DEVICE
- STTY.

The required and optional arguments associated with each of the above directives are given in the following subsections.

LPHDEF DIRECTIVE

The LPHDEF defines the channel and station table requirements of the Line Protocol Handler (LPH). It must be included in your CLM file exactly as shown below:

```
LPHDEF 0,100,200
```

For a complete description of the LPHDEF directive, see Section 6.

LPHn DIRECTIVE

The LPHn directive identifies the reader and punch stations on the line serviced by the LPH. Include LPH0 directives in your CLM file as follows:

For the card reader:

```
LPH0 lrn1,level,X'channel',0,speed,HDX
```

For the card punch:

```
LPH0 lrn2,level,X'channel',,speed,HDX
```

ARGUMENTS:

lrn1

The logical resource number associated with the card reader. The value for lrn1 is an integer from 2 through 252. A program can use this number to identify the card reader when requesting an input/output operation to the card reader.

lrn2

The logical resource number associated with the card punch. Normally, this number is one greater than the lrn used for the card reader; lrn2 cannot be the same value as lrn1.

level

The priority level at which the communications supervisor processes requests for an input/output operation to the station. The value for level is an integer from 7 through 58. It can be the same as the level specified for other communication stations, but it must be a higher number than the communications interrupt level(s) specified in the COMM directive. The level specified for one or more communication stations cannot also be used for noncommunication devices or tasks.

X'channel'

A four-digit hexadecimal number (from X'040n' to X'FF8n'), specifying the channel number of the station.

speed

The data rate in bits per second.

For a complete description of the LPHn directive, see Section 6.

LDBU DIRECTIVE

The LDBU directive loads the bound unit required by the Decision Data 8045 Card Reader/Punch. Include the following LDBU directive in the CLM file:

```
LDBU ZQLCRP
```

For a complete description of the LDBU directive, see Section 5.

DEVICE DIRECTIVE

DEVICE directives are required for each Decision Data 8045 Card Reader/Punch in the system in the following format:

```
DEVICE TTYxx,lrn,level,X'channel',RDRH,,N
```

ARGUMENTS:

TTYxx

The device_unit TTYxx identifies the type of device.

lrn

The logical resource number by which the device is requested. The value for lrn is a decimal integer from 2 through 252.

level

The priority level used by the device driver for the device. The value for level is a decimal integer from 7 through 58.

X'channel'

The four-digit hexadecimal channel number of the device.

RDRH

The unique name used by the File System to refer to the device.

N

Indicates that input/output to the card reader/punch is unbuffered.

STTY DIRECTIVE

The STTY directive specifies the file characteristics of a device that is not a disk device or a private device. The characteristics specified by this directive override the default characteristics established by the associated DEVICE directive (or the characteristics established by a previous STTY directive).

FORMAT:

```
STTY  device_name,[length],[X'dsw']
```

ARGUMENTS:

device_name

    This value identifies the device and must be the same as the one specified in the previous DEVICE directive.

[length]

    The length in bytes of one physical record.

[dsw]

    The reader/punch's device-specific word for connect/disconnect and read/write requests. The value of dsw is a 4-character hexadecimal number enclosed in apostrophes. (For detailed information, see the STTY directive in Section 6.)

SAMPLE CARD READER/PUNCH CONFIGURATION

The following directives configure a card/reader punch. The reader is associated with lrn 22 and the punch with lrn 23.

```
*  CARD READER - PUNCH
*
LPHDEF 0,100,200
LPHO 22,22,X'CA01',0,9600,HDX
LPHO 23,22,X'CA81',,9600,HDX
LDBU ZQLCRP
DEVICE TTY03,22,22,X'CA01',RDRH,,N
STTY RDRH,80,X'0001'
DEVICE TTY04,22,22,X'CA01',RDRB,,N
*
```

```
STTY RDRB,120,X'1001'
DEVICE TTY05,22,22,X'CA01',RDRS,,N
STTY RDRS,160,X'1201'
*
DEVICE TTY01,23,22,X'CA81',PUNH,,N
STTY PUNH,80,X'8001'
DEVICE TTY02,23,22,X'CA81',PUNB,,N
STTY PUNB,120,X'9001'
DEVICE TTY06,23,22,X'CA81',PUNS,,N
STTY PUNS,160,X'9201'
```

Appendix M
CONFIGURING THE FACIT 4042 PAPER TAPE READER/PUNCH

If your system includes a FACIT 4042 Paper Tape Reader/Punch, you must include the following CLM directives with appropriate arguments in your CLM file:

- MODEM
- LPHDEF
- LPHn
- LDBU
- DEVICE
- STTY.

This appendix describes the required and optional arguments associated with each of these CLM directives.

MODEM DIRECTIVE

The MODEM directive defines the characteristics of the direct connect interface required by FACIT 4042 Paper Tape Reader/Punch. It must be included in your CLM file exactly as shown below:

```
MODEM 5,X'20',X'20',X'20',X'00',X'20'
```

For a complete description of the MODEM directive, see Section 6.

LPHDEF DIRECTIVE

The LPHDEF directive defines the channel and station table requirements of the Line Protocol Handler (LPH). It must be included in your CLM file exactly as shown below:

```
LPHDEF 2,100,200
```

For a complete description of the LPHDEF directive, see Section 6.

LPHn DIRECTIVE

The LPHn directive identifies the first (or only) station on a line serviced by a user-written line protocol handler. It must be included in your CLM file as follows:

```
LPH2 lrn,level,X'channel',5,speed
```

ARGUMENTS:

lrn

The logical resource number associated with the station. The value for lrn is an integer from 2 through 252. A program can use this number to identify the station when requesting an input/output operation to the station.

level

The priority level at which the communications supervisor processes requests for an input/output operation to the station. The value for level is an integer from 7 through 58. It can be the same as the level specified for other communications stations, but it must be a higher number than the communications interrupt level(s) speci- fied in the COMM directive. The level specified for one or more communications stations also cannot be used for noncommunications devices or tasks.

X'channel'

A four-digit hexadecimal number (from X'040n' to X'FF8n') specifying the channel number of the station and the interrupt priority level (n) of the line. The channel number has the following format:

Bits 0 through 9 - The 10-bit channel address of the send or receive channel on the line.

Bits 10 through 13 - Must be set to zero.

Bits 14 and 15 - Specify n, the priority level at which a communications line interrupts the central processor. n can have a value of 0, 1, 2, or 3 as specified in the COMM directive. (See the description of the COMM directive for further information.)

[speed]

The data rate in bits per second.

For a discussion of permissible values for speed, see the description of the speed argument in the description of the LPHn directive in Section 6. The FACIT 4042 device is asynchronous.

For a complete description of the LPHn directive, see Section 6.

LDBU DIRECTIVE

The LDBU directive loads the bound unit required by the FACIT 4042 Paper Tape Reader/Punch. Include the following LDBU directive in the CLM file:

```
LDBU ZQLFAC
```

For a complete description of the LDBU directive, see Section 5.

DEVICE DIRECTIVE

A DEVICE directive is required for each FACIT 4042 Paper Tape Reader/Punch in the system in the following format:

```
DEVICE GENxx,lrn,level,X'channel',device_name,record_size,{B}
                                                           {N}
```

ARGUMENTS:

GENxx

The device unit GENxx identifies the type of device.

lrn

The logical resource number by which the device is requested. The value for lrn is a decimal integer from 3 through 252.

level

The priority level used by the device driver for the device. The value for level is a decimal integer from 7 through 58.

X'channel'

The four-digit hexadecimal channel number of the device.

device_name

A string of up to 12 ASCII characters that establishes a unique name used by the File System to refer to the device. The first character of the string must be alphabetic.

An asterisk can be specified that indicates a "private" device to be accessible only through physical input/output.

If this argument is omitted, the value of the device_unit argument is used for File System reference to the device.

record_size

The length in bytes of one physical record. If record_size is not specified, the default record size is as established by the device_unit argument.

$\begin{Bmatrix} B \\ N \end{Bmatrix}$

For normally unbuffered stations (such as BSC), B indicates that input/output to the station is buffered.

For normally buffered stations (such as TTY, VIP, PVE, ROP), N indicates that input/output to the station is unbuffered.

For TTY, VIP, ROP, PVE, ATD, and STD stations, the default is buffered. For BSC and XBSC stations, the default is unbuffered.

STTY DIRECTIVE

The STTY directive specifies the file characteristics of a device that is not a disk device or a private device. The characteristics specified by this directive override the default characteristics established by the associated DEVICE directive (or the characteristics established by a previous STTY directive).

FORMAT:

```
STTY  device_name,[length],[X'dsw']
```

ARGUMENTS:

device_name

This value identifies the device and must be the same as the one specified in the previous DEVICE directive.

[length]

The length, in bytes, of one physical record.

[dsw]

The reader/punch's device-specific word for connect/ disconnect and read/write requests. The value of dsw is a four-character hexadecimal number enclosed in apostrophes. (For detailed information, refer to the STTY directive in Section 6.)

SAMPLE FACIT PAPER TAPE READER/PUNCH CLM DIRECTIVES

The following directives configure a Facit Paper Tape Reader/Punch.

```
* FACIT TAPE READER/PUNCH
*
MODEM 5,X'20',X'20',X'20',X'00',X'20'
LPHDEF 2,100,200
LPH2 20,20,X'C901',5,2400
LDBU ZQLFAC
DEVICE GEN00,20,20,X'C901',FAC00,80,N
STTY FAC00,80,X'C000'
*
```

Appendix N
CONFIGURING THE MEMODYNE M-80 CASSETTE TAPE

If your system includes a Memodyne M-80 Cassette Tape unit, you must include the following CLM directives with appropriate arguments in your CLM file:

- LPHDEF
- LPHn
- LDBU
- DEVICE.

This appendix describes the required and optional arguments associated with each of these CLM directives.

LPHDEF DIRECTIVE

The LPHDEF directive defines the channel and station table requirements of the Line Protocol Handler (LPH). It must be included in your CLM file exactly as shown below:

```
LPHDEF 3,100,200
```

For a complete description of the LPHDEF directive, refer to Section 6.

LPHn DIRECTIVE

The LPHn directive identifies the first (or only) station on a line serviced by a user-written line protocol handler. It must be included in your CLM file as follows:

```
LPH3 lrn,level,X'channel',,speed
```

ARGUMENTS:

lrn

The logical resource number associated with the station. The value for lrn is an integer from 2 through 252. A program can use this number to identify the station when requesting an input/output operation to the station.

level

The priority level at which the communications supervisor processes requests for an input/output operation to the station. The value for level is an integer from 7 through 58. It can be the same as the level specified for other communications stations, but it must be a higher number than the communications interrupt level(s) specified in the COMM directive. The level specified for one or more communications stations cannot also be used for noncommunications devices or tasks.

X'channel'

A four-digit hexadecimal number (from X'040n' to X'FF8n') specifying the channel number of the station and the interrupt priority level (n) of the line. The channel number has the following format:

Bits 0 through 9 - The 10-bit channel address of the send or receive channel on the line.

Bits 10 through 13 - Must be set to zero.

Bits 14 and 15 - Specify n, the priority level at which a communications line interrupts the central processor. n has a value of 0, 1, 2, or 3 as specified in the COMM directive. (See the description of the COMM directive for further information.)

[speed]

The data rate in bits per second.

The firmware initializes the speed at 1200 bits per second. To change this speed, you must issue appropriate commands to the device.

For a complete description of the LPHn directive, refer to Section 6.

LDBU DIRECTIVE

The LDBU directive loads the bound unit required by the Memodyne M-80 Cassette Tape. Include the following LDBU directive in the CLM file:

```
LDBU ZQLCAS
```

For a complete description of the LDBU directive, refer to Section 5.

DEVICE DIRECTIVE

A DEVICE directive in the following format is required for each Memodyne M-80 Cassette Tape in the system:

```
DEVICE MTCxx,lrn,level,X'channel',device_name,record_size,{B}
                                                           {N}
```

ARGUMENTS:

MTCxx

The device unit MTCxx identifies the type of device.

lrn

The logical resource number by which the device is requested. The value for lrn is a decimal integer from 3 through 252.

level

The priority level used by the device driver for the device. The value for level is a decimal integer from 7 through 58.

X'channel'

The four-digit hexadecimal channel number of the device.

device_name

A string of up to 12 ASCII characters that establishes a unique name used by the File System to refer to the device. The first character of the string must be alphabetic.

An asterisk can be specified that indicates a private device to be accessible only through physical input/ output.

If this argument is omitted, the value of the device_unit argument is used for File System reference to the device.

record_size

The length in bytes of one physical record. If record_size is not specified, the default record size is as established by the device_unit argument.

$\begin{Bmatrix} B \\ N \end{Bmatrix}$

For normally unbuffered stations (such as BSC), B indicates that input/output to the station is buffered.

For normally buffered stations (such as TTY, VIP, PVE, ROP), N indicates that input/output to the station is unbuffered.

For TTY, VIP, ROP, PVE, ATD, and STD stations, the default is buffered. For BSC and XBSC stations, the default is unbuffered.

SAMPLE MEMODYNE M-80 CASSETTE TAPE CLM DIRECTIVES

The following CLM directives configure a Memodyne M-80 Cassette Tape:

```
*CASSETTE (MEMODYNE M-80)
*
LPHDEF 3,100,200
LPH3 21,21,X'C981',,1200,HDX
DEVICE MTC00,21,21,X'C981',CAS00,256,N
LDBU ZQLCAS
*
```

Appendix O
TRANSFERRING FROM REMOVABLE TO FIXED PLATTER

If your system has one or more fixed-platter disk devices, you may wish to copy the contents of your system volume to the fixed platter of a two-platter drive. To copy your system volume to a fixed-platter, refer to the Software Installation Guide.

Appendix P
COMMAND ACCOUNTING

Command accounting is a software component used to collect data concerning command usage and report on that usage. For each command, the following information is provided:

- User ID
- Account
- Command line
- Starting date and time
- Elapsed time
- CPU time
- Amount of memory used
- I/O activity
  - Communications
  - Disk
  - Other.

Deferred print activity is also monitored, and the same information is provided with the addition of a report of the action taken by the deferred print processor. For this report, the command line is replaced with the file pathname that was printed, and the total line count replaces "Other I/O activity."

The following paragraphs describe the general operation of command accounting, its installation, and activation.

GENERAL OPERATION

The operation of command accounting occurs in two phases:

1. Online data preparation and collection
2. Offline data reduction and reporting.

In the first phase, data is collected from the system either in the command processor or in the deferred print processor and stored in a binary file. In the second phase, this binary file is converted to an ASCII file, which is then sorted among several fields and reported.

INSTALLATION

To install command accounting, you need the following files which are supplied in the indicated directories:

- SID
  - ZXSTAT
  - DPSTAT
  - ZXDJ
  - ZJRNL
- SYS_CTL
  - JRNL_CMDS.UP
  - JRNL_CMDS.MC
  - JRNL_CMDS.QC
  - CACCNV.EC
  - JRNL_UP.EC
  - JRNL_PROC.EC
  - ACTRPT.EC
  - SDATE.EC
  - SACCT.EC
  - SUSER.EC
  - SCMDS.EC
- SYSLIB2
  - JRNL
  - CACONV
  - JOBACT

Also, the following CLM directives must be added to your CLM_USER file:

```
*        LDBU ZXSTAT
*        LDBU DPSTAT
```

ACTIVATION

To activate command accounting, execute the following OCL command:

```
EC >>SYS_CTL>JRNL_UP [n]
```

ARGUMENT:

n

The size of the binary data collection log file. The size is given in the number of 256-word blocks of 2 sectors each. The default value is 512. This provides a file of 1024 128-word sectors. The binary data file records approximately three command accounting log records per two 128-word sectors. Assuming a rate of one command every 2.5 seconds, the log would grow at the rate of 960 sectors per hour. The default size gives enough space for approximately one hour of data collection.

DESCRIPTION:

JRNL UP.EC activates JRNL and initiates data collection for command accounting. JRNL is a data collection program that collects binary data in a log file. Command accounting shares the log file with other applications collecting data (e.g., LISTENUR). If other applications are using JRNL to collect data, their initiation is also done by this EC.

Monitoring of each user group begins as soon as command accounting is activated. Information concerning all ECL and DP commands is collected as binary data in a log file. This file is created anew each time command accounting is activated or the log file is closed.

MONITORING

The binary log file records approximately three command accounting records per two 128-word sectors. Assuming a rate of one command every 2.5 seconds, the log grows at the rate of 960 sectors per hour. Thus, you should close the file for offline processing before it becomes too large. Either of the two following methods closes the log file:

1. Issue the following OCL command

   ```
   JRNL -C
   ```

   This command closes the log file, renames it, and executes JRNL_PROC.EC. The log file is prepared for further data collection and JRNL continues to collect command accounting data. This method is automatically executed when the log file runs out of space.

2. Issue the following OCL command

    JRNL -Q

This command closes the log file, renames it, and executes JRNL_PROC.EC. Command accounting is then terminated. If other applications are using JRNL to collect data, their operation is also terminated.

When the binary log file is closed, the log is renamed and JRNL_PROC.EC is executed. The file is renamed to:

    JmmddhhmmA

where mmdd is the month and day the file was closed and hhmm is the hour and minute it was closed. The renamed file is placed in the >>SYS_CTL directory. JRNL_PROC.EC then initiates the offline log processing.

Monitoring may be discontinued without closing the log file by executing the following OCL command:

    JRNL -Q C

This does not close the log file, but command accounting data collection is discontinued. If other applications are using JRNL to collect data, their operation is not terminated. Monitoring for command accounting may be resumed by executing the following OCL command:

    JRNL -M C

CONVERTING

When the binary log file is closed, JRNL PROC.EC is executed. This EC invokes the conversion and reporting utilities for command accounting. Conversion of the binary data to ASCII data is an offline process and is performed by the CACONV utility. The following is a description of this utility.

Command Accounting Conversion (CACONV)

Convert binary command accounting data into raw command accounting ASCII data.

FORMAT:

    CACONV binary_file [-COUT output_file]

ARGUMENTS:

binary_file

The pathname of the binary file containing the binary command accounting data collected online.

-COUT output_file

The pathname of the file to contain the raw command accounting ASCII data. The default pathname is: binary_file.C.

DESCRIPTION:

The CACONV command processes the specified binary file in order to create the records in an ASCII output file. Any old output file with the same name as that specified by the command is deleted. Processing continues through the entire file or until the BREAK key is pressed. (SR continues processing; UW or PI terminates the processing.) At regular intervals (every 512 records) during the processing, CACONV reports to the error-out file the number (in hexadecimal) of records thus far processed. At the end of CACONV, the total number (in hexadecimal) of records processed is reported.

The ASCII data record is contained in 132 bytes--one print line. The first character is the slew character (unprintable); the remaining 131 characters are data. Figure P-1 illustrates a portion of the raw ASCII data file. The fields of the record and their sizes in bytes are given in Table P-1.

Example 1:

CACONV >>SYS_CTL>J10300940A

Convert the binary command accounting data in the file whose pathname is >>SYS_CTL>J10300940A to ASCII command accounting data in the file whose pathname is >>SYS_CTL>J10300940A.C

Example 2:

CACONV ^VOL1>LOG>LOG_FILE -COUT OUTA

Convert the binary command accounting data in LOG_FILE in the directory ^VOL1>LOG to ASCII command accounting data in the file OUTA in the current working directory.

```
ABHENRY     SEC    A  00090000:00.75000000000081000021984/02/29 1435:32.81435:39.50209EC START_UP
CDHELEN     SEC    A  00090000:01.35000000000080000021984/02/29 1435:41.81435:48.50209EC START_UP
EFMARY      ESEC   A  00090000:00.85000000000081000021984/02/29 1435:50.81435:57.50209EC START_UP
GHLENNIE    ESEC   A  00090000:01.35000000000081000021984/02/29 1435:59.91436:06.60209EC START_UP
IJGEORGE    UTIL   B  00090000:01.30000000000080000021984/02/29 1436:08.91436:15.50209EC START_UP
KLKATHY     UTIL   B  00090000:01.25000000000086000021984/02/29 1436:18.01436:21.40209EC START_UP
HBHIS       L6     A  00090000:01.00000000000076000021984/02/29 1436:53.51436:56.10209EC START_UP
HBHIS       L6     A  00070000:01.45000000000105000021984/02/29 1436:57.11437:00.60000RON
ABHENRY     SEC    A  00130000:01.15000000000066000001984/02/29 1438:01.41438:11.90000CWD >SYSLIB2
CDHELEN     SEC    A  00130000:00.90000000000053000001984/02/29 1438:04.41438:13.60000CWD >SYSLIB2
EFMARY      ESEC   A  00130000:00.90000000000073000001984/02/29 1438:08.91438:22.30000CWD >SYSLIB2
GHLENNIE    ESEC   A  00130000:00.75000000000076000001984/02/29 1438:19.51438:37.50000CWD >SYSLIB2
IJGEORGE    UTIL   B  00130000:00.85000000000077000001984/02/29 1438:43.21438:50.90000CWD >SYSLIB2
ABHENRY     SEC    A  00740000:01.15000000000114000161984/02/29 1438:12.61438:51.00000LS -PN >PRUGS -HF
CDHELEN     SEC    A  00740000:00.70000000000056000161984/02/29 1438:13.81438:53.70000LS -PN >PRUGS -HF
EFMARY      ESEC   A  00740000:01.20000000000117000161984/02/29 1438:23.31439:02.70000LS -PN >PRUGS -HF
GHLENNIE    ESEC   A  00740000:01.45000000000119000161984/02/29 1438:37.71439:04.80000LS -PN >PRUGS -HF
ABHENRY            A  00110000:03.05000000000187000001984/02/29 1438:51.51439:05.70000NOP30
                      0000:00.20000000000160000001984/02/29 1438:54.01439:06.50000NOP30
                           .10000000000180000001984/02/29 1439:02.81439:07.60000NOP30
                              000000069000001984/02/29 1438:50.41439:08.60000CWD >SYSLIB2
                                   0017000001984/02/29 1439:04.91439:09.90000NOP30
                                      00011984/02/29 1439:05.91439:12.10000NOW
                                          84/02/29 1439:06.91439:13.70000NOW
                                              /29 1439:07.91439:14.20000NOW
                                                   439:10.21439:15.50000NOW
                                                      .41439:15.90000NOW
                                                          17.40000NOW
```

Figure P-1. Raw Command Accounting ASCII Data File

Table P-1. Data Fields and Sizes of Raw Command Accounting ASCII Data Record

Field	Bytes
Print Slew Character	1
Group Name	2
Person.Account.Mode	27
Task Memory Used (x 100 words)	4
CPU Time (hhmm:ss.tht)	11
Comm I/O	5
Disk	5
Other (i.e., unit record) I/O	5
External Start Date/Time	20
Elapsed time	9
Task Return Status	4
Command Name (truncated, if necessary)	39

NOTE

An asterisk (*) preceding the command name field denotes action taken by the deferred print processor; the pathname of the report is substituted for the command name.

REPORTING

Command accounting reports can be generated for the following sorted groupings:

- Date/time
- Command name
- Account name
- User name.

Reporting is done automatically when the log file is closed by the JRNL_PROC.EC. This EC sorts on the date/time grouping.

The command accounting report can be sorted by executing the following OCL command:

```
EC ACTRPT inpath outpath type
```

ARGUMENTS:

inpath

Pathname of the raw command accounting ASCII data file.

outpath

Pathname of the report file. This file must be created prior to the execution of this EC.

type

Specifies that the report be ordered according to one of the following groupings:

- DATE (date/time)
- CMDS (command)
- ACCT (account)
- USER (user).

If the type specified is ACCT or USER, you are asked to supply the name of the account or user.

Figure P-2 illustrates two portions from a command accounting report. The following applies to Figure P-2 for command accounting:

- Under the heading TASK NAME, an X preceding the command name means the command returned an error status. Usually this is a "not found" type error.

- Under the heading ELAPS, the elapsed time is given in either minutes (M) or hours (H).

PAGE 1 J O B A C C O U N T I N G 11/07/85

USER-ID	ACCOUNT	TASK NAME	START DATE TIME	ELAPS	CPU	MEM	COMM	DISK	OTHER
HENRY	XEC	X EC START_UP	85/11/04 1113:29.4	.22M	.650S	1.9K	0	184	2
HELEN	XEC	X EC START_UP	85/11/04 1113:35.7	.20M	1.000S	1.9K	0	196	2
MARY	EXEC	X EC START_UP	85/11/04 1113:43.9	.23M	.500S	1.9K	0	197	2
LENNIE	EXEC	X EC START_UP	85/11/04 1113:51.4	.15M	.900S	1.9K	0	200	2
GEORGE	UTIL	X EC START_UP	85/11/04 1113:58.9	.12M	1.100S	1.9K	0	192	2
KATHY	UTIL	X EC START_UP	85/11/04 1114:04.9	.04M	.500S	1.9K	0	186	2
HENRY	XEC	X EC >TEST>START_UP	85/11/04 1114:48.1	2.93M	10.700S	4.5K	0	1166	33
HELEN	XEC	X EC >TEST>START_UP	85/11/04 1114:51.4	2.91M	10.050S	4.5K	0	1123	33
MARY	EXEC	X EC >TEST>START_UP	85/11/04 1114:55.6	2.99M	9.600S	4.5K	0	1206	33
HENRY	XEC	CWD >SYSLIB2	85/11/04 1114:56.9	.08M	1.100S	2.3K	0	276	5
LENNIE	EXEC	X EC >TEST>START_UP	85/11/04 1115:00.5	2.95M	9.550S	4.5K	0	1154	33
HELEN	XEC	CWD >SYSLIB2	85/11/04 1115:01.3	.11M	1.400S	2.3K	0	285	5
HENRY	XEC	LS -PN >PROGS	85/11/04 1115:02.1	.31M	1.850S	9.1K	0	352	10
MARY	EXEC	CWD >SYSLIB2	85/11/04 1115:06.7	.14M	1.000S	2.3K	0	302	5
GEORGE	UTIL	X EC >TEST>START_UP	85/11/04 1115:07.1	2.83M	10.150S	4.5K	0	1095	33
HELEN	XEC	LS -PN >PROGS	85/11/04 1115:08.2	.28M	2.000S	9.1K	0	360	10
NNIE	EXEC	CWD >SYSLIB2	85/11/04 111			2.3K	0	316	5
	UTIL	X EC >TEST>START_UP	85/				0	1163	33
		LS -PN >PROGS						390	10
		NOP30							10
									10
GEORGE			1117:26.2	.15M	7.400S				
LENNIE			5/11/04 1117:27.2	.17M	6.700S	2.0K			
KATHY	UTIL		85/11/04 1117:33.5	.13M	6.900S	2.0K	0		
MARY	EXEC	DP	85/11/04 1117:41.2	.23M	7.750S	4.5K	0	119	
GEORGE	UTIL	DP >TRANS>ACTIVE.T	85/11/04 1117:43.2	.22M	7.850S	4.5K	0	1089	
LENNIE	EXEC	DP >TRANS>ACTIVE.T	85/11/04 1117:44.7	.20M	7.100S	4.5K	0	1145	32
KATHY	UTIL	DP >TRANS>ACTIVE.T	85/11/04 1117:48.4	.14M	7.500S	4.5K	0	1155	32
HENRY	XEC	*DP ^R4V84>TRANS>ACTIVE.T	85/11/04 1117:53.3	.31M	1.650S	0.0K	0	74	56
HELEN	XEC	*DP ^R4V84>TRANS>ACTIVE.T	85/11/04 1118:19.0	.29M	1.650S	0.0K	0	59	56
MARY	EXEC	*DP ^R4V84>TRANS>ACTIVE.T	85/11/04 1118:43.0	.29M	1.650S	0.0K	0	59	56
GEORGE	UTIL	*DP ^R4V84>TRANS>ACTIVE.T	85/11/04 1119:07.1	.29M	1.800S	0.0K	0	59	56
LENNIE	EXEC	*DP ^R4V84>TRANS>ACTIVE.T	85/11/04 1119:31.4	.29M	2.350S	0.0K	0	59	56
KATHY	UTIL	*DP ^R4V84>TRANS>ACTIVE.T	85/11/04 1119:55.5	.29M	2.100S	0.0K	0	59	56

85/11/04 DATE SUB-TOTAL ELAPSED TIME 34.3900 MINUTES
SUB-TOTAL CPU HOURS .1317

TOTAL ELAPSED HOURS .5727
TOTAL CPU HOURS .1317

Figure P-2. Sample Command Accounting Report

- Under the heading CPU, the CPU time is given in either minutes (M) or seconds (S).
- Under the heading MEM, the memory used by the command is in increments of tenths of K, where K equals 1000 words.
- Under the headings COMM, DISK, and OTHER, the number of communications, disk, and peripheral device I/O requests are reported for the elapsed time period.
- Under the heading TASK NAME, an asterisk (*) in front of DP or WP means the call came from a deferred print or a word processor print. If an *DP or *WP appears, the OTHER heading specifies the number of lines printed, not the number of I/O requests.
- Under the heading TASK NAME, the task name is truncated after 35 characters.
- For nested ECs, the outer EC includes the elapsed time for all of the ECs nested under it.

Appendix Q
TERMINAL PRESENTATION FACILITY

This appendix describes the function and configuration of the Terminal Presentation Facility (TPF).

TPF is an extension of Display Formatting and Control (DFC) software and is used to simulate VIP780x/77xx terminals on the VIP73xx/72xx family of terminals. Such simulation allows the user to substitute, without extensive recoding, character mode terminals (VIP73xx/72xx) for forms mode terminals (VIP780x/77xx). TPF is thus designed for those users who want to use VIP7200/7300 terminals in such forms-mode-oriented applications as OAS and DEF-II.

To configure your system with TPF, you must issue the VDAM directive in the CLM_USER file. The directive is:

```
VDAM [lrn,level][,maximum_terminals][,U][,78][,77]
```

ARGUMENTS

lrn

The logical resource number associated with the DFC task. The value for lrn is an integer from 3 to 252, or 256 to 4002. This value must be specified if asynchronous Display Formatting and Control processing is desired. Two free lrns are automatically reserved for DFC processing.

level

The priority level at which Display Formatting and Control processing operates. The value for level must be an integer from 6 through 61. This value must be specified if asynchronous processing is desired.

Either both lrn and level must be specified, or neither argument should be specified. If neither one is specified, the default is that no asynchronous orders are issued by Display Formatting and Control.

[maximum terminals]

The maximum number of terminals that can use DFC processing in one task group. The default value is 10.

[U]

Get memory from the user memory pool.

[78]

Load the 78xx-to-73xx simulation package. This argument is required if 78xx-to-73xx mapping is desired.

[77]

Load the 77xx-to-72xx or 77xx-to-73xx simulation package. This argument is required if 77xx-to-72xx or 77xx-to-73xx mapping is desired.

Appendix R
CONFIGURING PREALLOCATED BUFFER COPY QUEUES

This appendix describes the requirements for configuring preallocated buffer copy queues for the Executive. Preallocated buffer copy queues are a mechanism by which the performance of the Executive can be improved. The buffer copy queues are used in various places within the Executive, but most notably in the physical I/O (PIO) interface. In PIO operations, the buffers are used as system memory space to copy the PIO Request Block along with any buffers associated with the request. Without the buffers, a block of system memory would have to be requested and returned for each PIO request. This requesting and returning of system memory would lower the system performance.

To configure the preallocated buffer copy queues, load the SIOBQ bound unit for each queue you wish to configure. To load the SIOBQ bound unit, use the MLDBU directive which allows the bound unit to be loaded more than once. Use the MLDBU CLM directive as follows:

FORMAT

```
MLDBU SIOBQ,count,size[,STAT]
```

ARGUMENTS

count

The number of preallocated buffers within the queue.

size

The size in words of each of the preallocated buffers.

[,STAT]

Indicates whether statistics associated with the allocation of buffers are to be gathered. The default is not to gather the statistics.

The statistics feature is used to aid in determining the proper values for the count and size arguments. The recommended approach to determining these values is to initially configure the system with an estimate of the expected size and count with the statistics gathering feature. After running the system, XRAY can be used to examine the contents of the buffer queue. The statistics gathered to date can be displayed for each queue. The format for the statistics is:

Field	Bytes	Description
BS_NAL	4	Number of allocations for this queue
BS_SUM	4	Total number of words allocated
BS_SSQ	4	Sum of squares of the difference from mean
BS_MPT	2	Number of times queue was found empty
BS_NBE	2	Number of times queue was not big enough
BS_MIN	2	Minimum buffer allocated in words
BS_MAX	2	Maximum buffer allocated in words
BS_MFR	2	Minimum number of free buffers
BS_RFU	2	Reserved
BS_LEN	34	Length of statistics area

These statistics can be used to adjust the count and size arguments for future system boots. Once you are satisfied with the chosen values, statistics gathering should be turned off, since gathering statistics adds to the overhead of managing the queues.

MANUAL DIRECTORY

Base Publication Number	Manual Title
HE01	ONE PLUS Guide to Software Documentation
CZ02	GCOS 6 MOD 400 System Building and Administration
CZ03	GCOS 6 MOD 400 System Concepts
CZ04	GCOS 6 MOD 400 System User's Guide
CZ05	GCOS 6 MOD 400 System Programmer's Guide - Volume I
CZ06	GCOS 6 MOD 400 System Programmer's Guide - Volume II
CZ07	GCOS 6 MOD 400 Programmer's Pocket Guide
CZ09	GCOS 6 MOD 400 System Maintenance Facility Administrator's Guide
CZ10	GCOS 6 MOD 400 Menu System User's Guide
CZ11	GCOS 6 MOD 400 Software Installation Guide
CZ15	GCOS 6 MOD 400 Application Developer's Guide
CZ16	GCOS 6 MOD 400 System Messages
CZ17	GCOS 6 MOD 400 Commands
CZ18	GCOS 6 Sort/Merge
CZ19	GCOS 6 Data File Organizations and Formats
CZ20	GCOS 6 MOD 400 Transaction Control Language Facility
CZ21	GCOS 6 MOD 400 Display Formatting and Control
CZ22	GCOS 6 VISION Reference Manual
GZ13	GCOS 6 MOD 400 R3.1 to R4.0 Migration Guide
HC01	GCOS 6 MOD 400 Application Development Overview

INDEX

HONEYWELL INFORMATION SYSTEMS
Technical Publications Remarks Form

TITLE	DPS 6 GCOS 6 MOD 400 SYSTEM BUILDING AND ADMINISTRATION ADDENDUM A	ORDER NO.	CZ02-02A
		DATED	September 1986

ERRORS IN PUBLICATION

SUGGESTIONS FOR IMPROVEMENT TO PUBLICATION

Your comments will be investigated by appropriate technical personnel and action will be taken as required. Receipt of all forms will be acknowledged; however, if you require a detailed reply, check here. ☐

FROM: NAME ______________________ DATE ____________

TITLE ______________________

COMPANY ______________________

ADDRESS ______________________

____

PLEASE FOLD AND TAPE–
NOTE: U. S. Postal Service will not deliver stapled forms

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 39531 WALTHAM, MA02154

POSTAGE WILL BE PAID BY ADDRESSEE

HONEYWELL INFORMATION SYSTEMS
200 SMITH STREET
WALTHAM, MA 02154

ATTN: PUBLICATIONS, MS486

Honeywell

HONEYWELL INFORMATION SYSTEMS
Technical Publications Remarks Form

TITLE	DPS 6 GCOS 6 MOD 400 SYSTEM BUILDING AND ADMINISTRATION ADDENDUM A	ORDER NO.	CZ02-02A
		DATED	September 1986

ERRORS IN PUBLICATION

SUGGESTIONS FOR IMPROVEMENT TO PUBLICATION

Your comments will be investigated by appropriate technical personnel and action will be taken as required. Receipt of all forms will be acknowledged; however, if you require a detailed reply, check here. ☐

FROM: NAME ______________________ DATE ____________

TITLE ______________________

COMPANY ______________________

ADDRESS ______________________

____

PLEASE FOLD AND TAPE–
NOTE: U. S. Postal Service will not deliver stapled forms

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 39531 WALTHAM, MA02154

POSTAGE WILL BE PAID BY ADDRESSEE

HONEYWELL INFORMATION SYSTEMS
200 SMITH STREET
WALTHAM, MA 02154

ATTN: PUBLICATIONS, MS486

Honeywell

Together, we can find the answers.

Honeywell

Honeywell Information Systems
U.S.A.: 200 Smith St., MS 486, Waltham, MA 02154
Canada: 155 Gordon Baker Rd., Willowdale, ON M2H 3N7
U.K.: Great West Rd., Brentford, Middlesex TW8 9DH Italy: 32 Via Pirelli, 20124 Milano
Mexico: Avenida Nuevo Leon 250, Mexico 11, D.F. Japan: 2-2 Kanda Jimbo-cho Chiyoda-ku, Tokyo
Australia: 124 Walker St., North Sydney, N.S.W. 2060 S.E. Asia: Mandarin Plaza, Tsimshatsui East, H.K.

45023, 0486, Printed in U.S.A. CZ02-02