TOPS–20
System Manager's Guide

Electronic Distribution

June 1990

This document is intended for the person who is responsible for making final decisions for setting up and maintaining the efficient operation of a TOPS–20 installation.

Change bars in margins indicate material that has been added or changed since the previous printing of this manual.

OPERATING SYSTEM: TOPS–20 (KL Model B) Version 7.0

digital equipment corporation
maynard, massachusetts

First Printing, September 1985
Revised, June 1988
Revised, June 1990

CONTENTS

INDEX

FIGURES

TABLES

PREFACE

The TOPS–20 System Manager's Guide is written for the person who is responsible for establishing policies and procedures for a timesharing and/or batch processing installation using the TOPS–20 Operating System. Usually, this person is responsible for setting up and maintaining both the system hardware and software. The Site Management Guide, the TOPS–10/TOPS–20 Operator's Hardware Device and Maintenance Guide, and the TOPS–20 Operator's Guide provide you and your operations people with the necessary information to maintain your system hardware. These three manuals are referenced throughout this guide.

This guide deals primarily with your system software. It contains general suggestions for planning the installation of your software and for setting up your computer room to begin operations. The guide contains hints and suggestions for your system's operation, including when, and many times why, particular functions or procedures should be considered. It assumes that your system operator is responsible for implementing many of the decisions you make. In most cases, where lengthy implementation procedures are required, the appropriate reference is noted.

Chapters 1 and 2 describe the documentation, system logs, and special forms that you should have available to you, and in some cases, to system users. Chapter 2 also includes preliminary planning functions that you can do before the software is installed.

Chapter 3 describes the system directories and files that your system contains immediately after you install the software. It also describes the mechanisms you can use to change the installed TOPS–20 batch system and to test the integrity of your newly installed or updated system. In addition, it describes requirements for remote printing and for printing on devices attached to terminals.

Chapter 4 describes using your disk–pack and disk–drive resources to set up disk structures in a way that best suits your installation's needs. It also includes guidelines for determining the available disk space that you have to create user directories.

Chapter 5 describes creating and maintaining directories. It includes a detailed description of the three methods of administration you can choose from to control the creation and maintenance of directories. It describes how to use directory and file protection codes to expand or limit the type of access users can have to directories and files, and how to place users and directories in groups so that users can share files.

Chapter 6 describes the TOPS–20 accounting facility. This description includes how to choose an accounting scheme, how to create accounting files, and how to set the system to begin validating accounts.

Chapter 7 describes backing up your disk structures onto magnetic tape soon after software installation. It recommends the supplies needed and procedures that you should follow to save all your directories and files on a daily basis, and how to create a system crash tape in the event of a major problem with the file system.

Chapter 8 describes how you can use magnetic tapes to store important files (file archiving) and to save valuable disk space by copying infrequently accessed files to tape (file migration). It also describes how to give control of tape drive usage to the system and the operator (tape drive allocation), and how to set up your system to use labeled tapes (tape labeling).

Chapter 9	describes the procedures you must follow in the event that you have a problem with the file system or that a user has lost the files in a directory. It also describes using your system crash tape and your daily backup tapes to resolve these problems. In addition, it describes how to prevent "offline" disks from hanging user jobs, and discusses dumping memory for nonfatal system errors.
Chapter 10	describes the tuning mechanisms that allow you to change the behavior of your system. Each description includes why you may want to use a particular mechanism, how to use it, and the effects it may have on your system.
Chapter 11	describes the access control mechanisms that you can use to alter system policy decisions or to increase security against unauthorized system use. This chapter includes the type of policy changes you may want to make at your installation.
Chapter 12	describes the Common File System, a software feature of TOPS-20. This chapter discusses the rules, options, and restrictions associated with sharing files among systems.
Chapter 13	describes the Local Area Transport (LAT) software, for use with terminal servers in Ethernet local area networks.

The following conventions and symbols are used throughout this guide:

Convention/Symbol	**Description**
n-	n refers to the latest version of a particular file, for example, 7-CONFIG.CMD.
UPPERCASE	In user input representations, indicates information that must be entered exactly as shown.
lowercase	In user input representations, indicates variable information that is determined by you.
underlining	Indicates the information that you must type at your terminal.
()	In user input representations, encloses guide word information. Pressing the ESCAPE or ALTMODE key on your terminal causes guidewords to be printed by the computer.
<RET>	Indicates you should press the RET or RETURN key on your terminal. Unless otherwise noted, pressing RETURN terminates all command or input strings.
<ESC>	Indicates you should press the ESC key on your terminal.
CTRL/key	Indicates you should press the CTRL key on your terminal. The CTRL key is always used in conjunction with another key, for example, CTRL/Z.

CHAPTER 1

DOCUMENTATION

Section 1.1 describes the documentation provided by DIGITAL and recommends the manuals with which you should be familiar to manage your system. Section 1.2 describes adding your own documentation, for example, special forms, to the documentation you receive from DIGITAL. Be sure you have all available documentation convenient to your system users.

1.1 DOCUMENTS AVAILABLE FROM DIGITAL

All documentation for the TOPS-20 Operating System is contained in the TOPS-20 Software Notebook Set. This notebook set contains information pertaining to the most recent version of TOPS-20. It is organized functionally to facilitate referencing manuals. Each manual contains cross references to other manuals within the set that further explain a subject.

This manual assumes that you are familiar with some of the manuals in the notebook set. In particular, you should be familiar with the information in the TOPS-20 Operator's Guide, the TOPS-20 User's Guide, the DECSYSTEM-20 Technical Summary, the TOPS-10/TOPS-20 Operator's Hardware Device and Maintenance Guide, and the TOPS-20 KL10 Model B Installation Guide.

Any additional documents that you need depend on the configuration of your system. For example, if your system has IBM emulation and termination (DNxx), you should be familiar with the IBM Emulation-Termination Manual. It includes installation procedures and descriptions of the operator and user interfaces. If your system has DECnet, you should be familiar with the various DECnet-20 manuals. If you are using LAT terminal servers in an Ethernet local area network, refer to the documentation that is provided with LAT terminal servers, in addition to chapter 13 of this manual.

In addition to the TOPS-20 Software Notebook Set, you receive the TOPS-20 Beware File Listing. It is distributed with the software

installation and distribution magnetic tapes. Before installing a new version of the software on your system, read the Beware File. It contains last-minute changes to the software that have not been documented, and hints or suggestions for installing or using the new software.

With each new system, you should also receive two stand-alone documents, which are documents not included in the notebook set. These manuals assist you in 1) preparing your site for the hardware installation, the Site Preparation Guide, and 2) maintaining and reporting problems about your system's software and hardware, the Site Management Guide.

NOTE

Your Sales Representative delivers the Site Preparation Guide, and your Field Service Representative delivers the Site Management Guide.

This manual (the TOPS-20 System Manager's Guide) deals primarily with installing and maintaining the software on your system. Therefore, it is assumed that you have already used the Site Preparation Guide to install your system hardware.

The Site Management Guide is designed for use by both you (along with your operations people) and your Field Service Representative. You should begin using this manual immediately after you install your hardware. It contains schedules, procedures, and logs for recording and evaluating all information pertinent to the operation and care of the system. The manual belongs to DIGITAL, but it is kept and maintained at your computer site. For added convenience and organization, many system managers keep all their important system information in the same binder as the Site Management Guide. For example, they keep system logs and operator shift change information in the same binder, along with other special forms. Section 1.2 describes several forms that you may include in a system log book or, as suggested here, in the Site Management Guide.

DIGITAL places a major emphasis on the documentation provided to its customers. The Software Publications Department continues to solicit suggestions for improvement and corrections from the users of its documentation. Encourage users to comment on the manuals you receive with your system. For convenience, a Reader Comment Form is located at the back of each manual.

1.2 DOCUMENTS PREPARED AT YOUR INSTALLATION

Sections 1.2.1 through 1.2.5 describe some forms that may be useful at your installation. A sample form is provided in each section.

1.2.1 System Log

Every system must have a system log for recording problems and procedures relating to both hardware and software. All operators and system programmers should record the following types of activities in the log, along with the date, time, and their names:

- o System backup procedures
- o Beginning and ending of timesharing (for example, the times the system was started and stopped for preventive maintenance or repair)
- o Problems in hardware or software AND the actions taken to correct the problems (always save the CTY (operator terminal) output or copy of the typescript)
- o New or revised software installed
- o New users or changes to existing user data or directories
- o New structures or changes to existing structures

Most system problems are easier to solve (and, hence, less costly) if you keep an accurate record of all activities. The Site Management Guide has a section set aside for system log information. This section contains preprinted forms that you can use to record system log information, or you can design your own forms. You can store these forms in the Site Management Guide or in a separate binder.

You should design your log so that it is easy to use and read. Remember, you are likely to have the most problems when the system is new, so NOW is the time to start using the log. The following two pages contain sample left- and right-hand pages of a log book. The left-hand page (Figure 1-1) contains information concerning hardware maintenance; the right-hand page (Figure 1-2) is a problem report, containing:

- o The time of the entry
- o A "Y" or "N" answer to whether the system had to be reloaded
- o The name of the person making the entry
- o A few words describing the nature of the activity
- o A record of calls to Digital Field Service (F/S)
- o A description of the device or program causing the problem
- o Remarks about the entry

SYSTEM LOG
MAINTENANCE PERFORMED DATE_____

Figure 1-1: Sample System Log (Hardware Maintenance)

PAGE_____

SYSTEM LOG DATE_____

TIME	RELOAD	NAME	MONITOR OR HARDWARE MAINTENANCE ACTIVITY	F/S ATTN	DEVICE OR PROGRAM	ENTRY

Figure 1–2: Sample System Log (Problem Report)

1.2.2 Mountable Structure Sign–Up Log

In addition to keeping the system log, you should also record requests from users to mount structures. (Chapter 4 describes how to set up and use structures.) Without a formal scheduling procedure, some users may monopolize the use of a structure and frustrate other users, who do not have the opportunity to mount and use their structures, usually because there are no disk drives available. To avoid this situation, set up a procedure whereby users inform the operator when they need to use a structure. The operator can then schedule the length of time specified on the request log. On a busy day, when many users are issuing mount requests for structures, the operator checks the log before granting or denying the mount requests. This scheduling allows you to service many requests for mounting structures in a fair and orderly manner. The sample mountable structure sign–up log shown in Figure 1–3 contains:

- The scheduled mounting time
- The scheduled time needed to use the structure
- The actual time the structure was mounted
- The actual time the structure was removed
- The name of the user who initiated the request
- The structure name (or pack ID)
- A column for any special instructions or notes

Remember that this log is only a sample; you should design a form that best suits your own requirements.

1.2.3 System Access Request Form

Some installations have many users requesting access to the system for the first time. You need standard information from these users before you can process their requests and create directories for them. For example, you must know which system they need to access (if you have more than one system), their names, selected passwords, departments, accounts, etc. You can organize these requests by providing a System Access Request Form that is kept in an easy–to–access area, perhaps outside the computer room. You can require signatures of department managers on the access form to ensure that prospective users have approval to charge computer usage to accounts. Figure 1–4 is a sample of a system access request form.

If you are using CFS–20 software, refer to Chapter 12, The Common File System, for further considerations in assigning users to systems.

MOUNTABLE STRUCTURE
SIGN – LOG

PAGE_____

DATE_____

SCHEDULED		ACTUAL				
MOUNTING TIME	TIME NEEDED	MOUNTING TIME	TIME REMOVED	USER NAME	PACK ID(s)	NOTES

Figure 1–3: Sample Mountable Structure Sign–Up Log

SYSTEM ACCESS REQUEST

SYSTEM NAME:_____ DEPT.:_____
YOUR NAME:_____ ACCT.:_____
PROJECT:_____

PERM. ACCESS? ____YES ____NO (FROM:_____TO:_____)

SUPERVISOR:_____ MGR:_____
SIGNATURE SIGNATURE

DIRECTORY NAME (1–39 CHAR.):_____

PASSWD.:_____ DIR. PROT.(DEF.777700) |__| OTHER:_____

*DO YOU REQUIRE PRIVILEGES ON THE SYSTEM? |__| N |__| Y (TYPE:_____)

DO YOU WANT TO CREATE SUBDIRS.? |__| N |__| Y (HOW MANY?_____ MAX.= 8)

DO YOU WANT TO BE IN A GROUP WITH OTHER USERS OR DIRECTORIES? |__| N

|__| Y (NAME OF USER(S) OR DIR.(S):_____ _____ _____)

BRIEFLY DESCRIBE THE TYPE OF WORK YOU WILL PERFORM. FOR EXAMPLE, CREATING AND EDITING FILES, APPLICATIONS PROGRAMMING, COMPILER PROGRAMMING, ETC._____

OPERATIONS USE ONLY

DIR.	PASSWD.	STRUCTURE	WORKING QUOTA	PERM. QUOTA

USER GROUP	DIR. GROUP	ACCOUNT

SUBDIRECTORIES	SCHED. CLASS	PRIVILEGES	DATE CREATED

COMMENTS:_____
*MUST BE APPROVED BY OPERATIONS MANAGEMENT

Figure 1–4: System Access Request

1.2.4 Operator Work Request Form

You may want a form that allows users to request work from the operator. Examples of requests made to the operator are initializing tapes, transferring files between systems, and making changes to directories. You should set up a procedure for handling these requests. Figure 1–5 is a sample of an operator work request form.

1.2.5 Operator Shift Change Log

You may want to set up a binder to contain operator shift change information. Each operator records new procedures, or special instructions that the incoming operator needs to know. The incoming operator reads the operator shift log before starting the new shift. For example, the first shift operator changes the procedure for storing tapes, and records the new procedure in the shift change log. The information in the shift change log should not concern problems with the system, but should contain important information about the system or the computer room. The incoming operator still reads the system log book to determine the status of the system and any problems that have occurred during the previous shift. Figure 1–6 is a sample of an operator shift change log.

OPERATOR WORK REQUEST

NAME:_____ || NAME OF SYSTEM:_____
DIRECTORY NAME:_____ || PRIORITY: NORMAL_____ RUSH_____
DATE SUBMITTED:_____ || DEPT. NO.:_____
PHONE EXT.:_____ || ACCOUNT:_____

JOB 1	INPUT _____	OUTPUT _____	DONE	INSTRUCTIONS:

JOB 2	INPUT _____	OUTPUT _____	DONE	INSTRUCTIONS:

OPERATOR:_____ || OPERATOR COMMENTS:_____
SYSTEM:_____ ||
DATE COMPLETE:_____ ||

Figure 1–5: Operator Work Request

OPERATOR SHIFT CHANGE LOG

DATE	OPERATOR	SHIFT	COMMENTS

Figure 1–6: Operator Shift Change Log

CHAPTER 2

PREPARING FOR SOFTWARE INSTALLATION

You can establish many of the policies and procedures for your computer site before you install the software. It may help you later if some of the preliminary decisions and preparations are done before you begin setting up the system and handling requests from users. The following suggestions for preparing your installation are not all–inclusive. Some TOPS–20 installations have specific requirements or restrictions that are not considered here. You can use this list as a guideline for the types of decisions you can make in the early stages of setting up your computer site.

2.1 SECURING THE COMPUTER ROOM

Select the type of computer room security you need and a method of enforcement. Many system managers do not allow non–operations people to enter the computer room. Establish an open– or closed–door policy, and notify users of your policy. If you decide on a closed–door policy, notify users of the procedures that they should use to contact you (or the operator) and to submit their job requests.

2.2 HANDLING USER REQUESTS

Determine how user requests will be handled. You can handle jobs on a first–come basis, or on a priority basis. You can set up request boxes outside the computer room that the operator checks regularly. You can also establish a location where users can leave disks and tapes for the operator to mount. Post a sign–up sheet so that users can specify the time they need the tape or disk mounted. Chapter 1 describes sample forms that can be completed by users to request initial access to the system and to request that work be done by the operator.

2.3 ORDERING SUPPLIES

Assign someone the responsibility for ordering paper supplies, ribbons, cards, and magnetic tapes. Chapter 7 provides an estimate of the number of tapes you should have to begin a backup procedure immediately after you install the software. Be sure you have enough CTY (operator terminal) and line printer paper to begin operations.

2.4 SCHEDULING OPERATOR TASKS

The operator performs tasks either on a regular basis or on an as–needed basis. Decide which operator tasks will be performed on a regular schedule. Be sure to include hardware, software, and documentation related tasks. These regularly scheduled tasks can be performed daily, weekly, or monthly.

The following lists are samples of hardware– and software–related tasks that your operator may perform.

Hardware–Related Tasks

Regular Schedule	As–Needed Schedule
Clean tops of disk drives.	Replenish paper in the line printer.
Clean magnetic tape drives.	Remove reports from the line printer and distribute (perhaps to mail boxes).
Vacuum line printer to remove paper chad.	Replenish paper in operator’s console.
Load mountable structures according to a schedule.	Physically load and unload magnetic tape and disk drives.

Regular Schedule	As–Needed Schedule
Software–Related Tasks	
Bring up system after weekly maintenance.	Bring up system after a crash.
Run scheduled batch production jobs.	Maintain the batch system for users.
Save the contents of disk on magnetic tape.	Save special disk areas on magnetic tape.
Create a system "crash" tape for backup.	Restore selected user disk areas as needed.
Run the SPEAR program for daily error analysis.	Interact with users.
Submit a daily control file for accounting.	Create and update user directories.
Create the Message–of–the–Day with the MAIL program.	Monitor disk space.

Establish a location for keeping the hard–copy output from the CTY. Your Field Service Representative needs this information if you have problems with your system. Have the operator tear off the copy and store it daily.

Documentation–related tasks include:

- keeping a hand–written log of system activities (System Log)
- recording operator shift change information (Operator Shift Change Log)
- coordinating the mounting and dismounting of structures (Mountable Structure Sign–up Log)

Chapter 1 describes creating a system log, an operator shift change log, and a mountable structure sign–up log.

2.5 SELECTING SYSTEM FEATURES

Determine the system features you want to enable during software installation. When you install the software, you create a file called n–CONFIG.CMD. This file is read by a start–up program (n–SETSPD) when the system is started for the first time and each subsequent time that you reload and start the system. The n–CONFIG.CMD file defines the line speeds for your terminals and many system parameters. Most of the decisions you must make concerning the parameters in this file are described throughout this manual. As you read each chapter, you can list the parameters that you want to place in the n–CONFIG.CMD file. Many system managers choose to introduce new pieces of software slowly. Therefore, you may want to disable some of the parameters until you have run the new software for awhile. You can edit the n–CONFIG.CMD file to add new software features to the system. You should edit the file at a convenient time before you reload the system. Then, when the system restarts, the new software features are enabled.

NOTE

The operator can run the n–SETSPD program interactively during timesharing to override many of the n–CONFIG.CMD options, as described in the TOPS–20 Operator's Guide.

Chapters 3 through 13 describe setting up and maintaining your system. Read these chapters thoroughly. They contain important information to help you make decisions both before and after you install the software.

CHAPTER 3

AFTER SOFTWARE INSTALLATION

3.1 OVERVIEW

After you install the TOPS-20 software, your system contains all the directories and files necessary for you to start preparing for timesharing and batch processing. This chapter describes the directories, files, and system logical names created during software installation. Also included are suggestions for creating additional directories and logical names to assist you and system users.

3.2 SPECIAL SYSTEM DIRECTORIES

You initialize the file system during software installation. At this time, the system automatically creates nine directories on the disk that you defined as the system structure. These directories are:

<ROOT-DIRECTORY>
<SYSTEM>
<SUBSYS>
<NEW-SYSTEM>
<NEW-SUBSYS>
<ACCOUNTS>
<OPERATOR>
<SPOOL>
<SYSTEM-ERROR>

Sections 3.2.1 through 3.2.7 describe these directories and their use. Section 3.2.8 describes additional directories you can create and how they are useful.

Chapter 5 also describes creating directories and discusses the structure of directories.

3.2.1 <ROOT-DIRECTORY>

The <ROOT-DIRECTORY> contains a separate file for each first-level directory on the system structure as follows:

(where STR: is the name of the structure).

The <ROOT-DIRECTORY> is the most important directory created. Without it, directories and files cannot be accessed. You must NEVER modify this directory. The system maintains a backup copy of <ROOT-DIRECTORY> that can be accessed if the original copy is destroyed. (Refer to Section 9.3, RESTORING <ROOT-DIRECTORY>.)

Each structure you create in addition to the system structure has a <ROOT-DIRECTORY>. The <ROOT-DIRECTORY> on any structure points to all the first-level directories created under the <ROOT-DIRECTORY>.

After you install the software, give the DIRECTORY command for <ROOT-DIRECTORY>. The output on your terminal appears similar to the example below. Note that each directory is a file in the <ROOT-DIRECTORY>. The differences between this list and the one on your terminal depend on the model system you have and the type of unbundled software you have purchased.

$DIRECTORY (OF FILES) STR:<ROOT-DIRECTORY><RET>

```
  STR:<ROOT-DIRECTORY>
ACCOUNTS.DIRECTORY.1
BACKUP-COPY-OF-ROOT-DIRECTORY.IMAGE.1
BOOTSTRAP.BIN.1
DSKBTTBL..1
FRONT-END-FILE-SYSTEM.BIN.1
INDEX-TABLE.BIN.1
NEW-SUBSYS.DIRECTORY.1
NEW-SYSTEM.DIRECTORY.1
OPERATOR.DIRECTORY.1
ROOT-DIRECTORY.DIRECTORY.1
SPOOL.DIRECTORY.1
SUBSYS.DIRECTORY.1
SYSTEM.DIRECTORY.1
SYSTEM-ERROR.DIRECTORY.1
UETP.DIRECTORY.1

Total of 14 Files
```

3.2.2 <SYSTEM>

The directory <SYSTEM> contains data and program files that the system uses during normal operation. Table 3–1 lists many of the files that appear in this directory.

Table 3–1: <SYSTEM> Files

File Name	Explanation
0DUMP11.BIN	Contains a dump of front–end memory after the front end crashes.
2060–MONBIG.EXE	The smallest runnable non–ARPANET monitor.
2060–MONMAX.EXE	The largest runnable non–ARPANET monitor.
n–CONFIG.CMD	Contains definitions of line speeds, system logical names, printer VFU files, magnetic tape logical unit numbers, DECnet parameters, and additional system–dependent parameters. These system parameters are set every time the system starts. The value n equals the latest release of TOPS–20.
n–PTYCON.ATO	Contains the commands that are given automatically at the operator's console every time the system starts. You may modify this file to suit your own installation. The value n equals the latest release of TOPS–20.
n–SETSPD.EXE	Program that reads the n–CONFIG.CMD file and sets up the parameters that it contains. The value n equals the latest release of TOPS–20.
n–SYSJOB.EXE	Program that runs in a process created by the monitor and takes commands from the file n–SYSJOB.RUN. The value n equals the latest release of TOPS–20.

Table 3–1: <SYSTEM> Files (Cont.)

File Name	Explanation
n–SYSJOB.RUN	Contains commands that SYSJOB processes. The value n equals the latest release of TOPS–20.
ACCOUNTS–TABLE.BIN	Contains the information necessary to validate accounts.
AN–MONBIG.EXE	The smallest ARPANET timesharing monitor.
AN–MONDCN.EXE	A monitor that includes ARPANET and DECnet.
AN–MONMAX.EXE	The largest ARPANET timesharing monitor.
BUGS.MAC	Contains a list of all BUGHLT, BUGINF, and BUGCHK messages.
CHECKD.EXE	Program that creates structures and checks file–system consistency.
COMAND.CMD	An installation–specific systemwide COMAND.CMD file.
DEVICE–STATUS.BIN	Contains status information for tape drives, disk drives, and disk structures. It is maintained by MOUNTR.
DUMP.CPY	Contains a copy of main memory at the time of the last system crash. It is copied from DUMP.EXE to maintain a history of crashes. This file is written by the n–SETSPD program when the system is rebooted.
DUMP.EXE	Contains a copy of main memory at the time of the last system crash. You must have this file to get a system dump after a crash.
ERRMES.BIN	Contains binary system error messages.
EXEC.EXE	The TOPS–20 Command Processor.

Table 3–1: <SYSTEM> Files (Cont.)

File Name	Explanation
FEDDT.EXE	A DDT program used for debugging the front end.
HOSTS.TXT	Defines ARPANET host names and their number translations.
INTERNET.ADDRESS	Defines the system's TCP/IP address for AN20, CI20, and NIA20 interfaces.
INTERNET.GATEWAYS	Defines the network gateways for reaching host systems on remote networks.
INTERNET.NAMESERVERS	Lists name server hosts.
IPALOD.EXE	Program that loads the CI20 microcode. (The microcode is contained in the file.) After the loading has completed, TOPS–20 starts the CI.
KNILDR.EXE	Program that loads the NIA20 microcode. (The microcode is contained in the file.) It is run automatically at system startup to start the NI.
LOGIN.CMD	An installation–specific systemwide LOGIN.CMD file.
LOGOUT.CMD	An installation–specific systemwide LOGOUT.CMD file.
MONITR.EXE	The current monitor.
MONNAM.TXT	Contains the monitor name printed at the beginning of the system greeting line.
PROGRAM–NAME–CACHE.TXT	Contains a list of the programs that should be loaded into the program–name cache. Read by the MAPPER program.
REAPER.CMD	Contains a list of default commands to REAPER. The REAPER program reads this file each time it is run.

Table 3–1: <SYSTEM> Files (Cont.)

File Name	Explanation
RSX20F.MAP	Contains symbol locations for the front–end processor. It is used by the FEDDT program.
SYSJOB.HLP	Contains information about the SYSJOB program.
SYSTEM.CMD	Contains OPR commands and is read by the OPR program at system startup.
TAPNAM.TXT	Text file that contains the installation identifier that is written on VOL1 labels for labeled tapes.
TGHA.EXE	Program that analyzes and corrects MOS memory problems.
TGHA.HLP	Contains information about the TGHA program.
TOPS–20.DOC	Text file that contains summary information about the latest release of TOPS–20.

3.2.3 Restoring the Directory <SYSTEM>

If the contents of <SYSTEM> are accidentally lost or destroyed, you can restore the directory from the TOPS–20 Installation Tape or your latest system backup tape. (Refer to Chapter 7 for information about creating system backup tapes.) Use the procedure below to restore <SYSTEM> directory. If you have enabled tape drive allocation, use the MOUNT command instead of the ASSIGN command. (Refer to Section 8.3 for information about using tape drive allocation.)

1. Mount the appropriate tape (in this example, it is on drive MTA0:).

2. Give the following commands at your terminal.

```
@ENABLE (CAPABILITIES) <RET>
$ASSIGN (DEVICE) MTA0: <RET>
$SKIP (DEVICE) MTA0: 4 FILES <RET>
$RUN (PROGRAM) MTA0: <RET>
```

```
DUMPER> TAPE (DEVICE) MTA0: <RET>
DUMPER> RESTORE (TAPE FILES) DSK*:<*>*.*.* (TO) <SYSTEM> <RET>

DUMPER TAPE #1 , , FRIDAY 1-NOV-88 330
LOADING FILES INTO <SYSTEM>

END OF SAVESET
DUMPER>EXIT <RET>
$
```

3.2.4 <SUBSYS>

The directory <SUBSYS> contains system programs (and their help files) that the user may want to run. The directory protection code set for <SUBSYS> prevents users from changing the files in this directory. Many of the file protections require users to enable WHEEL or OPERATOR capabilities to use the files. (Refer to Chapter 5 for information about directory and file protections and special capabilities.) Table 3-2 lists the programs and files commonly placed in <SUBSYS>. An asterisk precedes all unbundled software.

Table 3-2: STR:<SUBSYS> Files

Programs	Explanation
ACTGEN.EXE	Program that takes information from accounting files and creates the account validation data base.
ACTGEN.HLP	Contains information about the ACTGEN program.
ACTSYM.UNV	A file of universal symbols for USAGE accounting programs.
ANAUNV.UNV	A file of ARPANET universal symbols.
*B362LB.REL	BLISS functions needed to rebuild the Record Management Services facility (RMS-20) from AUTOPATCH.
*BASIC.EXE	The BASIC compiler.
BATCON.EXE	Program that controls batch jobs.
CDRIVE.EXE	Program that controls card readers.

Table 3-2: STR:<SUBSYS> Files (Cont.)

Programs	Explanation
CHECKD.EXE	Program that creates structures and checks file-system consistency (same as in <SYSTEM>).
CHECKD.HLP	Contains information about the CHECKD program.
CHKPNT.EXE	Program that makes accounting entries in the file <ACCOUNTS>CHECKPOINT.BIN.
CHKPNT.HLP	Contains information about the CHKPNT program.
CMD.REL	A library file of routines for the COMND monitor call.
CMD.UNV	A file of universal symbols for the COMND monitor call.
*COBDDT.HLP	Contains information about COBDDT.
*COBDDT.REL	The COBOL debugging program.
*COBOL.EXE	The COBOL compiler.
*COBOL.HLP	Contains information about the COBOL compiler.
CREF.EXE	Program that produces a cross-reference listing.
CREF.HLP	Contains information about the CREF program.
DIL.LIB	A library file of data definitions for COBOL programs that use the Data Interchange Library (DIL) facility.
DIL.REL	The DIL subroutines.
DILV7.FOR	Contains data definitions for FORTRAN programs that use DIL.
DITV7.FOR	Contains data definitions for FORTRAN programs that use the data transmission component of DIL.
DIXV7.FOR	Contains data definitions for FORTRAN programs that use the data conversion component of DIL.
DLUSER.EXE	Program that saves and restores the directory parameters.
DLUSER.HLP	Contains information about the DLUSER program.

Table 3–2: STR:<SUBSYS> Files (Cont.)

Programs	Explanation
DUMPER.EXE	Program that saves and restores files to and from magnetic tape.
DUMPER.HLP	Contains information about the DUMPER program.
DX20LD.EXE	Program that loads DX20 microcode.
DXMCA.ADX	Microcode for DX20 tape subsystem controller.
EDDT.REL	A component of the debugging program for the TOPS–20 monitor.
EDIT.EXE	A line–oriented text editor.
EDIT.HLP	Contains information about the EDIT program.
FE.EXE	Program that is used when copying files from the front–end file system to the TOPS–20 file system and vice versa.
FE.HLP	Contains information about the FE program.
FEDDT.EXE	The debugging program for the front end.
FILCOM.EXE	Program that compares the contents of two files.
FILCOM.HLP	Contains information about the FILCOM program.
FILDDT.EXE	A DDT program used for examining the contents of system dumps (DUMP.CPY).
*FORDDT.HLP	Contains information about the FORDDT program.
*FORDDT.REL	The FORTRAN debugging program.
FORMAT.EXE	Program used to format RP04/RP06 disk packs while the system is in timesharing mode.
FORMAT.HLP	Contains information about the FORMAT program.
*FOROTS.EXE	The FORTRAN object–time system (operating system interface).
*FORTRA.EXE	The FORTRAN compiler.
GALGEN.EXE	Program that creates the parameter file (GALCNF) for building the GALAXY programs.

Table 3–2: STR:<SUBSYS> Files (Cont.)

Programs	Explanation
GLOBS.UNV	A file of universal symbols for the TOPS–20 monitor.
GLXLIB.EXE	Object–time system used by the GALAXY programs.
HELP.HLP	Contains information about the HELP command.
*IBMSPL.EXE	Spooling program that sends IBM–batch–job files to remote IBM host and retrieves the output.
INFO.EXE	Program that gives information to programs using IPCF.
*ISAM.EXE	Program that maintains COBOL single–key indexed sequential files.
*ISAM.HLP	Contains information about the ISAM program.
KDDT.REL	A component of the debugging program for the TOPS–20 monitor.
KNILDR.EXE	Program that loads the NIA20 microcode.
LCPORN.REL	The LCP subprocess to the OPR program.
LCPTAB.REL	The LCP command table.
*LIBARY.EXE	Program that creates, maintains, and lists the contents of COBOL library files.
*LIBARY.HLP	Contains information about the LIBARY program.
*LIBO12.EXE	The COBOL object–time system (operating system interface).
*LIBOL.REL	Contains the COBOL library subroutines.
LINK.EXE	Program that loads relocatable binary programs.
LINK.HLP	Contains information about the LINK program.
LISSPL.EXE	Cluster LPTSPL listener that receives print requests from remote cluster LPTSPLs and forwards the requests to QUASAR.
LP64.RAM	Translation RAM file for a 64–character line printer. Read by n–SETSPD.

Table 3–2: STR:<SUBSYS> Files (Cont.)

Programs	Explanation
LP96.RAM	Translation RAM file for a 96–character line printer. Read by n–SETSPD.
LPTSPL.EXE	Program that controls output to local line printers as well as TTY, LAT, DQS, and cluster printers.
MACREL.REL	Run–time file for macros in MACSYM.
MACRO.EXE	The MACRO assembler.
MACRO.HLP	Contains information about the MACRO assembler.
MACSYM.UNV	Contains system macros.
MS.EXE	Program that sends messages to users.
MS.HLP	Contains information about the MAIL program.
MS.EXE	Program that receives mail from the MAIL program and places it in the appropriate mailbox.
MAKDMP.EXE	Program that produces a standard DUMP.EXE file in <SYSTEM>.
MAKLIB.EXE	Program that creates relocatable subroutine libraries.
MAKLIB.HLP	Contains information about the MAKLIB program.
MAKRAM.EXE	Program that creates a translation RAM file for line printers.
MAKRAM.HLP	Contains information about the MAKRAM program.
MAKVFU.EXE	Program that creates a vertical formatting unit (VFU) file.
MAKVFU.HLP	Contains information about the MAKVFU program.
MAPPER.EXE	Program that loads the program–name cache. (Refer to Section 10.4, Improving Program Startup Time.)
MDDT.REL	A component of the debugging program for the TOPS–20 monitor.

Table 3–2: STR:<SUBSYS> Files (Cont.)

Programs	Explanation
MONSYM.REL	Object file that contains monitor call symbol definitions.
MONSYM.UNV	Contains symbol definitions for monitor calls.
MOUNTR.EXE	Program that mounts tapes and structures.
MSCPAR.UNV	A file of universal symbols used to build the MSCP component of the TOPS–20 monitor.
NEBULA.EXE	The cluster GALAXY message router between nodes in a cluster.
*NFT.EXE	DECnet file transfer program.
*NFT.HLP	Contains information about the NFT.EXE program.
*NMLT20	DECnet program that performs the network control program functions.
NORMAL.VFU	Vertical formatting unit file for line printers.
OPR.EXE	Program that the operator uses to interface with all jobs and devices on the system.
OPR.HLP	Contains information about the OPR program.
ORION.EXE	Program that processes messages sent by the OPR, MOUNTR, LPTSPL, QUASAR, EXEC, etc. programs.
OVRLAY.REL	Overlay manager for the LINK program.
PA1050.EXE	The TOPS–10 Compatibility Package.
PAT.EXE	Version of PA1050 that can be used in debugging.
PHYPAR.UNV	A file of universal symbols for TOPS–20 input/output programs.
PLEASE.EXE	Program that establishes a dialog with the operator.
PROLOG.UNV	A file of universal symbols used to build the TOPS–20 monitor.
PTYCON.EXE	Program that controls many jobs from a single terminal.

Table 3–2: STR:<SUBSYS> Files (Cont.)

Programs	Explanation
PTYCON.HLP	Contains information about the PTYCON program.
QUASAR.EXE	Program that does the central queuing and scheduling for the batch system.
RDMAIL.EXE	Program that allows a user to read mail sent with the MAIL program.
RDMAIL.HLP	Contains information about the RDMAIL program.
REAPER.EXE	Program that marks files for migration to magnetic tape.
REAPER.HLP	Contains information about the REAPER program.
*RERUN.EXE	Restarts COBOL programs.
*RERUN.HLP	Contains information about the RERUN program.
RETRFB.SPE	Contains SPEAR report templates.
RFB.EYE	Contains internal definitions for the RETRIEVE function of the SPEAR program.
RMS.EXE	RMS–20 used in Section 0 of memory to get XRMS.EXE.
RMSCOB.EXE	RMS–20 used by COBOL V12B programs.
RMSFAL.EXE	Program that 'listens' for DECnet file transfers.
RMSINI.REL	Routine called by BLISS and MACRO programs to initialize RMS–20.
RMSINT.R36	Unsupported BLISS interface file for RMS–20.
RMSINT.UNV	MACRO interface file for RMS–20.
RMSUTL.EXE	The RMS–20 file maintenance utility.
RSXFMT.EXE	Utility program used for converting TOPS–20 files to a format used by the front end and vice versa.
RSXFMT.HLP	Contains information about the RSXFMT program.
RUNOFF.EXE	Program that helps with text preparation.

Table 3–2: STR:<SUBSYS> Files (Cont.)

Programs	Explanation
RUNOFF.HLP	Contains information about the RUNOFF program.
SCAPAR.UNV	A parameter file of symbols for the SCA inter–system communication routines.
SDDT.EXE	DDT debugger for programs without a symbol table.
*SELOTS.EXE	Program that interfaces between the COBOL language and the COBOL object–time system (LIBOL). (It is used with versions of COBOL up to and including version 11.)
SERCOD.UNV	Contains definitions for the SYSERR error codes.
*SIX12.REL	The BLISS debugger.
*SORT.EXE	Program that sorts files record by record.
*SORT.HLP	Contains information about the SORT program.
SPRINT.EXE	Program that creates batch jobs from card input.
SPROUT.EXE	Output spooler for card punch, paper tape punch, and plotter.
SPEAR.EXE	Segment of SPEAR program.
SPRRET.EXE	Segment of SPEAR program.
SPRSUM.EXE	Segment of SPEAR program.
SYSJOB.HLP	Contains information about the SYSJOB program.
SYSTAP.CTL	Control file that creates a system backup tape.
TCX.EXE	A DIGITAL Standard Runoff index utility.
TCX.HLP	Contains information about the TCX utility.
TERMINAL.HLP	Contains information about the TERMINAL command.
TOC.EXE	A DIGITAL Standard Runoff utility for creating a table of contents.
TOC.HLP	Contains information about the TOC utility.
TV.EXE	A character–oriented text editor.

Table 3–2: STR:<SUBSYS> Files (Cont.)

Programs	Explanation
UDDT.EXE	DDT debugger for programs with a symbol table.
ULIST.EXE	Program for printing information about directories and users.
ULIST.HLP	Contains information about the ULIST program.
VERIFY.EXE	Program that is used during software installation to determine the integrity of files. It verifies checksums and version numbers of the .EXE files.
WATCH.EXE	Program for observing system performance.
WATCH.HLP	Contains information about the WATCH program.
*XPORT.REL	Library containing the BLISS transportable I/O, memory, and string functions.
XRMS.EXE	RMS–20 library that is mapped to an extended (nonzero) section for execution of RMS functions.

NOTE

All the .HLP files can be displayed using the HELP command; for example, the command @HELP WATCH displays the WATCH.HLP file.

3.2.5 Restoring the Directory <SUBSYS>

If the contents of <SUBSYS> are accidentally lost or destroyed, you can restore the directory from the TOPS–20 Installation Tape or your latest system backup tape. (Refer to Chapter 7 for information about creating system backup tapes.) Use the procedure below to restore the <SUBSYS> directory. If you have enabled tape drive allocation, use the MOUNT command instead of the ASSIGN command. (Refer to Section 8.3 for information about using tape drive allocation.)

1. Mount the appropriate tape (in this example, it is on drive MTA0:)

2. Give the following commands at your terminal.

```
@ENABLE (CAPABILITIES) <RET>
$ASSIGN (DEVICE) MTA0: <RET>
$SKIP (DEVICE) MTA0: 4 FILES <RET>
$RUN (PROGRAM) MTA0: <RET>

DUMPER> TAPE (DEVICE) MTA0: <RET>
DUMPER> SKIP (NUMBER OF SAVESETS) 1 <RET>
DUMPER> RESTORE (TAPE FILES) DSK*:<*>*.*.* (TO)     <SUBSYS>
<RET>

DUMPER TAPE # 1 , , SATURDAY, 3-NOV-88 330
LOADING FILES INTO STR:<SUBSYS>
END OF SAVESET

DUMPER> EXIT <RET>
$
```

3.2.6 <NEW-SYSTEM> and <NEW-SUBSYS>

The first time you install the TOPS-20 software, the DLUSER program creates the directories <NEW-SYSTEM> and <NEW-SUBSYS>. They do not contain files. You can use these directories when a new release becomes available and you are updating the existing system. When DIGITAL distributes an updated monitor on the TOPS-20 Installation Tape, you restore the first two savesets from this tape to the directories <NEW-SYSTEM> and <NEW-SUBSYS> respectively. You use these directories until you feel comfortable with the new software. Should you have any problems with the new software, you can easily revert to using the old software. Appendix A of the TOPS-20 KL Model B Installation Guide detail the procedures to update one software release to another.

If you have no problems with the new monitor, and you are comfortable with it, copy all the files in the directory <NEW-SYSTEM> into the directory <SYSTEM> and all the files in the directory <NEW-SUBSYS> into the directory <SUBSYS>. You can now delete all the files in <NEW-SYSTEM> and <NEW-SUBSYS>. The directories <NEW-SYSTEM> and <NEW-SUBSYS> remain empty until a new version of the TOPS-20 software is distributed.

NOTE

After you copy the new files into the directories <SYSTEM> and <SUBSYS>, you cannot revert to the old system software unless you reinstall the system using the old monitor or backup tapes.

3.2.7 <ACCOUNTS>, <OPERATOR>, <SPOOL>, and <SYSTEM-ERROR>

<ACCOUNTS> - After installation, the directory <ACCOUNTS> contains one file, SYSTEM-DATA.BIN. This file contains all the accounting system entries for each user. If the directory <ACCOUNTS> is destroyed, the accounting system creates a new SYSTEM-DATA.BIN file.

After the first LOGIN on the system, the system creates the <ACCOUNTS>CHECKPOINT.BIN file. This file stores accounting entries for each user during the time the user is logged in. After a user logs out, the accounting data stored in CHECKPOINT.BIN is copied to the SYSTEM-DATA.BIN file. When the system comes up after a crash, the monitor examines <ACCOUNTS>CHECKPOINT.BIN to determine which users were logged in at the time of the crash, and stores the data in CHECKPOINT.BIN in SYSTEM-DATA.BIN. Therefore, users who did not log out in the normal fashion, because of a crash, are still charged for their log-in time.

<OPERATOR> - The directory <OPERATOR> normally contains the file PTYCON.LOG. This file usually contains a record of all the activities that occur under the operator jobs that are controlled by PTYCON. The directory <OPERATOR> may also contain files the operator needs to run the system. **<SPOOL>** - The directory <SPOOL> contains files that the spooling system needs before performing any input or output. They are kept in this area until they can be output to a slow-speed device such as a line printer. This area is also used for input of files from the local card reader, if one is attached. It may also be used for input of files from IBM remote stations. The file PRIMARY-MASTER-QUEUE-FILE.QUASAR is created in this directory. It contains a copy of the input queues so that they are not destroyed if the system crashes. You must either delete this file or process all entries in the queues before installing a new version of the batch system that has a different queue format. The GALAXY.DOC file describes the new software components and tells you if the queue format has changed.

<SYSTEM-ERROR> - The directory <SYSTEM-ERROR> contains the file ERROR.SYS. The ERROR.SYS file contains entries about system errors and is read by the system error recovery program, SPEAR.

3.2.8 Other Useful Directories

You may want to create additional directories for storing different versions of programs or text. Some useful directories are listed below. You should give these directories the proper protection number and make them files-only directories.

Directory and File Protection

Directories and files that are executed or read by the entire user community should not be given the default protection 777700, which allows no access. They should be given the directory protection 777740 and the file protection 777752 or 777712. (Section 5.7 describes directory and file protections.)

<NEW> The directory <NEW> can contain versions of your software that are not completely tested or that are drastically different from the current versions. If you create a directory <NEW>, users will find it more convenient if you also create the system-logical name NEW: defined as PUB:<NEW>, SYS:, where PUB: is the system structure. This logical name allows them to run all new software by merely typing NEW: and the program name. If there is no file with the given name in <NEW>, the system uses the version currently on <SUBSYS>. (Refer to Section 3.3 for a description of logical names.)

<OLD> The directory <OLD> can contain the old version of software as newer versions appear on <SUBSYS>. If programs or data do not work with new software, the user has a chance to correct the problems before the older software is no longer available. Users will find it convenient if you also define the system-logical name OLD: as PUB:<OLD>,SYS:, where PUB: is the system structure.

By creating the directories <NEW> and <OLD>, you gradually introduce new software to system users. When a new version becomes available, place it in the directory <NEW>. When the software has been in use awhile, move the version in <NEW> to <SUBSYS>, and the version in <SUBSYS> to <OLD>. Store the version in <OLD> on a system back-up tape. Every time you change a version of the software, you should send a system-wide message to all users.

<HELP> The directory <HELP> contains documents and help files that describe the system software. As different versions of software appear on <SYSTEM>, <NEW>, and <OLD>, you should make a list of changes incorporated in the new versions and place it in the directory <HELP>. You can move all files with the file type .HLP from <SUBSYS> to the directory <HELP>. The HELP command still works correctly if you define the system-logical name HLP: to be PUB:<HELP>,SYS:, where PUB: is the system structure.

<REMARKS> The directory <REMARKS> contains messages from users to the operator. These messages are usually general system comments or complaints. When a user wants to send the operator a message that does not require an immediate response, he can send a message to the directory <REMARKS> using the MAIL program. (Refer to the description of the MAIL program in the TOPS-20 User Utilities Guide.) A typical message may be a request for supplies, for example, LA36 paper or ribbon. Creating the directory <REMARKS> avoids constant interruptions to the operator from users issuing PLEASE requests. The operator can read the messages in <REMARKS> at a specified time each day, or simply when he has time.

3.3 SYSTEM-LOGICAL NAMES

A logical name is a descriptive word used to establish a search route to locate files. It can be up to 39 alphanumeric characters; however, it is usually three to six alphanumeric characters. Because logical names are used in place of device names, they always end with a colon. Logical names tell the system where and in what order to search for files. When a user types a logical name, the system searches the directories in the order they were defined or listed by the logical name. Although users can define logical names for their own use (refer to the TOPS-20 User's Guide), the logical names described here can be used by all users of the system. You can define system-logical names in the n-CONFIG.CMD file.

During installation, several systemwide logical names are defined by the monitor, and may be overridden in the n-CONFIG.CMD file. They are SYS:, defined as PUB:<NEW-SUBSYS>, PUB:<SUBSYS>; SYSTEM:, defined as PUB:<NEW-SYSTEM>, PUB:<SYSTEM>; DEFAULT-EXEC:, defined as SYSTEM:EXEC.EXE; and POBOX:, defined as the public structure. (PUB: is the system structure.) You may decide to add other logical names to aid users in accessing files. If you want the logical names to be permanent, place the definitions (using an editor) in the <SYSTEM>n-CONFIG.CMD file on the system structure.

SYSTEM:, SYS:, DEFAULT-EXEC:, POBOX:, and some other frequently used system-logical names are explained below.

3.3.1 SYSTEM:

The logical name, SYSTEM:, defines a search list that contains all the system programs and files that the system needs to operate. SYSTEM: should always contain the directory <SYSTEM> on the system structure. If you are updating the system with a new monitor, the definition of SYSTEM: in the n-CONFIG.CMD file also contains the directory <NEW-SYSTEM>. For example,

```
DEFINE SYSTEM:  STR:<NEW-SYSTEM>,STR:<SYSTEM>
```

where:

STR: is the name of the system structure.

3.3.2 SYS:

The logical name SYS: defines a search list that contains all the system programs a user may want to run. SYS: should always contain the directory <SUBSYS> and any other library directories that contain commonly used programs. If you are updating the system with a new monitor, the definition of SYS: in the n-CONFIG.CMD file also contains the directory <NEW-SUBSYS>. For example,

```
DEFINE SYS: STR:<NEW-SUBSYS>,STR:<SUBSYS>
```

where:

STR: is the name of the system structure.

Be sure to set the protection on the library files in <SUBSYS> (or <NEW-SUBSYS>) to 777740. This protection allows access by all users.

3.3.3 NEW:

The logical name NEW: defines a search list containing a directory that has new software, followed by the system-logical name SYS:. The definition for this, which you would put in n-CONFIG.CMD, is the following:

```
DEFINE NEW: STR:<NEW>,SYS:
```


With this systemwide logical name, the user can give the command:

@DEFINE (LOGICAL NAME) SYS: (AS) NEW: <RET>

Now, when the user runs a program, the system looks first in the directory STR:<NEW>, and then in the normal system search list SYS:. This way, the user always gets the most recent version of any program.

3.3.4 OLD:

If you have old versions of programs, defining the system-logical name OLD: may be helpful to users. The usual definition of the logical name OLD: is:

```
DEFINE OLD: STR:<OLD>,SYS:
```

The definition OLD: has the same type of effect as the definition NEW:. If the user gives the command:

@DEFINE (LOGICAL NAME) SYS: (AS) OLD:<RET>

whenever he runs a program, he will get the oldest version available.

3.3.5 HLP:

If you want to keep programs and documentation in separate directories, you should store the documentation in <HELP>. The HELP command searches the directories identified by the logical name HLP:, so you must define the logical name HLP: to be the directory <HELP>.

The definition of HLP: in n-CONFIG.CMD should be:

```
DEFINE HLP:  STR:<HELP>
```

3.3.6 SERR:

The logical name SERR: is defined by the system at startup. It points to the area <SYSTEM-ERROR> on the system structure. The system writes the ERROR.SYS file to this area, which may be used later to produce reports.

3.3.7 DMP:

When the system is re-booted after a crash, the file DUMP.EXE is overwritten with a copy of memory. Upon system startup, the n-SETSPD program copies the contents of DUMP.EXE to the DUMP-version-name.CPY file on the system structure (name is the name of the bug, and version is the edit number of the monitor that was running at the time of the crash). System crashes cause DUMP.EXE to be overwritten, but new versions of the .CPY file accumulate. To keep the system structure clear of .CPY files, define DMP: in the n-CONFIG.CMD file as follows:

```
DEFINE DMP:  STR:<DIRECTORY>
```

The structure and directory are your choice; you should not specify a filename. Versions of the .CPY file hereafter accumulate in the defined area.

In CFS configurations, systems should not share a common DMP: definition, because this could lead to confusion about which dump came from which system.

If the DUMP-ON-BUGCHK feature is enabled, the n-SETSPD program is run after continuable system errors, and it copies to DMP: all new DUMP.EXE files that it finds from its scan of dumpable structures. Refer to Section 9.10 for complete information on DUMP-ON-BUGCHK.

As n-SETSPD copies a file to the area defined by DMP:, it sends a message to the CTY specifying where it's copying the file to and from. For example:

```
Copying system dump
        from:  STR:<SYSTEM>DUMP.EXE.1
        to:    PS60:<DUMPS>DUMP-12345-WSPNEG.CPY.1
```

3.3.8 DEFAULT-EXEC:

The logical name DEFAULT-EXEC: defines a search list that points to the TOPS-20 Command Processor (EXEC). When users log in or give the PUSH command, the EXEC program is activated. Some experienced users may choose to run their own copies of the EXEC, not the standard system version. Such users can define DEFAULT-EXEC: to be the file name for their private EXEC, and can take advantage of this feature after giving the PUSH command. This command must be given at the EXEC level, while in batch or interactive mode. PUSH commands issued from other program levels may invoke the standard system version, unless the program has been written to use DEFAULT-EXEC: if it is defined. By default, DEFAULT-EXEC: is defined as SYSTEM:EXEC.EXE.

Refer to the DECSYSTEM-20 Technical Summary and the TOPS-20 Commands Reference Manual for more complete information on the EXEC.

3.3.9 POBOX:

The logical name POBOX: defines a search list that points to structures where users' mail files reside. Mail sent to a user goes to the first MAIL.TXT.1 file in the user's directory that the system encounters in its search.

By default, POBOX: is defined as the public structure. You can redefine POBOX: in the n-CONFIG.CMD file. By redefining POBOX:, you can prevent users' mail files from filling up the public structure.

In CFS-20 configurations, redefining POBOX: is especially useful. You can define POBOX: to be the same structure for all systems, establishing a central location for all mail files in the configuration. Then, no matter what system users log onto, they are automatically directed to this one area when they give commands to access their mail. They do not have to spend time logging onto various systems to access mail that would otherwise have been sent to a public structure (which is a separate structure for each system unless the "login structure" feature is enabled). To set up this central location, the same DEFINE command should be entered in each system's n-CONFIG.CMD file. Refer to Chapter 12, The Common File System, for further information on CFS-20.

3.3.10 NRT:

The logical name NRT: (Network Remote Terminal) is applicable only if your system has DECnet communications software. When a user issues the SET HOST command to connect to a remote system, the CTERM-SERVER communications program is run by default. If the remote node does not support CTERM, the host system tries to connect the user again, this time using the program defined by NRT:.

Examples

1. For TOPS-20 to TOPS-20 communications, give the following definition:

   ```
   DEFINE NRT: SYS:SETHOST.EXE
   ```

2. For multi-operating system DECnet communications, you can specify the HOST program (located on the TOPS-20 tools tape):

   ```
   DEFINE NRT: HOST.EXE
   ```

3.3.11 SPOOL:

The logical name SPOOL: directs the system to the directory <SPOOL> on the system structure. The GALAXY batch and spooling components read and write files in this area. Also, the monitor writes spooled files to this area. (Section 3.2.7 contains detailed information on <SPOOL>.

3.4 CONSOLE FRONT-END FILES

The console front-end computer consists of a PDP-11 with 28K 16-bit words of memory. When the system is brought up for timesharing, the front-end monitor, RSX20F, is loaded in the PDP-11 memory and started. The TOPS-20 monitor is loaded in KL10 main memory and started. Thus, you have two computers working together. Both computers have their own monitor and related software.

The front-end file system consists of the RSX20F monitor and related programs (tasks) and files. During software installation, these front-end files are transferred from floppy disks to a special area on the system structure unless an RP07 is being used as the system structure. If an RP07 is being used as the system structure, only the files on the TOPS-20 Installation Tape will be placed on the RP07. The front-end files, on the floppy disks, must be placed on a dual-ported RP06 disk drive attached to the PDP-11 front end. (Refer to the TOPS-20 KL Model B Installation Guide for the procedure for creating the front-end file system when using an RP07 disk drive as the system structure.)

The area the front-end files are placed on is called the FRONT-END FILES area, or FILES-11 area. Once this area has been set-up, there is normally no need to get these files again from floppy disks. The floppy disks used to install the system become backup devices in case the system structure is destroyed, or in the case where an RP07 is being used as the system structure, they can be used to recreate your front-end file structure. It is a good idea to make an extra copy of your installation floppies in the event one of your original floppies is destroyed and you need to restore the FRONT-END FILES area. Refer to Chapter 7, System Backup Procedures, for a description of the COP program that is used to copy floppy disks.

As previously stated, the front-end files must always be placed on a dual-ported RP06 attached to the PDP-11 front end. This allows the front-end processor to access these files while the main processor accesses TOPS-20 files on the same or different disk packs.

The RSX20F monitor and its related programs do the following:

- Control input/output and communications devices.
- Interface with the main computer.
- Load the TOPS-20 monitor at system startup, and reload TOPS-20 if a crash occurs.
- Report system errors to TOPS-20.
- Perform system diagnostic functions.

Table 3-3 lists the programs and files located in the FRONT-END FILES area, with a brief description of each. Files with the file type TSK are programs that can be run under the front-end monitor RSX20F; files with the type MCB contain the microcode for the host processor (KL10); files with the type EXB are bootstrap programs used to load the TOPS-20 monitor; and files with the type CMD are programs that record information about system errors. This information is read by the system error recovery program, SPEAR. Beginning with TOPS-20 Version 6, console front-end filenames include edit-level numbers that indicate the versions of the particular programs, for example, F11ACP.TSK;1505.

Table 3-3: Console Front-End Files

File	Contents or Function
BF16N1.A11	MOS memory-timing RAM file. It is a nonexecutable file containing MOS memory data.
BF64N1.A11	Timing file for 64K RAM chips.
BOO.TSK	Used to boot RSX20F.
BOOT.EXB	The central processor disk bootstrap program that boots TOPS-20 from disk.
CLOCK.CMD	Saves contents of memory locations for diagnosing errors that are purposely induced by Field Service.
COP.TSK	Copies the contents of floppy disks.
CRAM.CMD	Saves contents of memory locations for diagnosing CRAM parity errors.

Table 3–3: Console Front–End Files (Cont.)

File	Contents or Function
DEX.CMD	Saves contents of memory locations for diagnosing Deposit Examine failures (PI level 0 interrupt).
DMO.TSK	Dismounts a front–end device and allows a reboot.
DRAM.CMD	Saves contents of memory locations for diagnosing DRAM parity errors.
EBUS.CMD	Saves contents of memory locations for diagnosing EBUS parity errors.
FMPAR.CMD	Saves contents of memory locations for diagnosing fast memory parity errors.
F11ACP.TSK	File handler for front–end disk files.
HALT.CMD	Saves contents of memory locations for diagnosing KL halt errors.
INI.TSK	Initializes a front–end files area.
KLDISC.TSK	Provides the KLINIK line disconnect service.
KLI.TSK	KLINIT program for initializing the central processor (KL20). Loads microcode, configures cache and main memory, loads bootstrap.
KLRING.TSK	Provides the KLINIK line ring service.
KLX.MCB	KL10 microcode file.
KPALV.CMD	Saves contents of memory locations for diagnosing keep alive cease errors.
LHALT.CMD	Saves contents of memory locations for diagnosing KL halt errors. Provides more information than HALT.CMD
LOGXFR.TSK	Transfers the PARSER.LOG file to the TOPS–20 error file.
MIDNIT.TSK	Updates the time of day through midnight.
MOU.TSK	Mounts a device for use with the front end.
MTBOOT.EXB	Boots a TOPS–20 monitor from magnetic tape.

Table 3–3: Console Front–End Files (Cont.)

File	Contents or Function
PARSER.TSK	The front–end command parser (prompts PAR>). The primary means of access to front–end programs.
PIP.TSK	Front–end program for file transfer.
RED.TSK	Tells the system where to look for the front–end files device, SY0:.
RP2DBT.EXB	A front–end BOOT file with the DX20 microcode loaded for the RP20 disk sybsystem.
RP2MBT.EXB	A front–end MTBOOT file with the DX20 microcode loaded for the RP20 disk sybsystem.
RSX20F.MAP	Contains symbolic definitions for RSX20F.
RSX20F.SYS	Virgin image of front–end monitor (RSX20F).
SAV.TSK	Saves the front–end monitor and bootstrap on disk.
SETSPD.TSK	Sets system parameters, such as line speeds.
TIMEO.CMD	Saves contents of memory locations for diagnosing protocol timeout errors.
TKTN.TSK	Terminates tasks, reports errors, and requests reloads.
T20ACP.TSK	Interfaces between front–end and TOPS–20 file systems.
UFD.TSK	Sets up user–file directories in the front–end files area.
ZAP.TSK	Makes binary modifications to task images.

3.5 TAILORING THE BATCH SYSTEM

Most installations use the parameters and defaults in the distributed version of the batch system. However, you can modify some of these parameters if required by the batch processing procedures at your installation.

DIGITAL distributes a program with the TOPS-20 software that allows you to tailor the standard batch system to the requirements of your installation. This program, called GALGEN.EXE, is located in the directory <SUBSYS> on the system structure. You can run GALGEN at the time the system software is installed or at a later date. In either case, you must have a working batch system before you can generate a new one using GALGEN. This means if you are installing the system, you must first install the batch system that is distributed with every new version of the TOPS-20 software (on the software installation tape). You can then run the GALGEN program and tailor the batch system before it becomes available for general use.

If you tailor the batch system at a later date, you can run the GALGEN program with users logged in. However, for safety reasons, the system should be stand-alone during the critical phase of stopping the old batch system and starting the new one. The batch queues, however, need not be empty. That is, batch jobs can be waiting to be processed at the time you bring the system down.

The TOPS-20 KL Model B Installation Guide contains the procedures for running the GALGEN program.

3.6 CHECKING THE SOFTWARE (UETP)

After the system software is installed, you or the Software Specialist can run the User Environment Test Package (UETP). UETP is a collection of programs, data files, and batch control files designed to allow you to test the integrity of various system elements. In addition to testing that the hardware has been properly installed, UETP ensures that the TOPS-20 Operating System is running and that the languages you have selected for your operation are available.

UETP creates a moderate load on the system, consisting of various defined procedures that closely resemble the load in an actual operation. Later, you may want to tailor UETP to test a software load that more closely resembles your particular system's use.

The TOPS-10/TOPS-20 User Environment Test Package Reference Manual describes UETP, the individual component tests, typical message information, and the procedures for adding new tests.

3.7 REMOTE PRINTERS

The DECSYSTEM-20s at your site may be included in a DECnet network, local area network, or CFS-20 cluster. These networks provide TOPS-20 users with numerous printing options. Users, not restricted to local printers on the systems they logged in to, can direct output to remote printers attached to:

- VMS systems in a DECnet network -- Distributed Queue Service (DQS) printers.
- LAT servers in a local area network -- LAT printers.
- Other TOPS-20 systems in a CFS-20 cluster -- cluster printers.

3.7.1 Remote Printing Requirements

- Use of DQS printers requires DECnet.

- Use of LAT printers requires a LAT server running at least Version 5.1 of the LAT protocol. Digital-supported LAT printers must be one of the following devices: LA50, LA75, LA100, LA120, LN03.

  When starting up a LAT printer, the operator must enter a description for the printer with the /TERMINAL-CHARACTERISTIC: switch to the OPR>START PRINTER command. (If the switch is omitted, the system prompts the operator for this information.) You or a system programmer must establish values for the switch argument in the LPTSPL module, LPTUSR.MAC. Instructions for doing this are included in the GALAXY documentation file.

- Use of a remote cluster printer requires DECnet.

  In addition, "cluster GALAXY" must be enabled on both the requesting and serving systems if users are to have access to the full set of remote-printer functions. These functions include: obtaining information on remote print queues, canceling remote print requests, and receiving notification of queuing and completing of the remote print job.

  A system that is servicing remote cluster print jobs must run LPTSPL and LISSPL.

  A system that is sending print jobs to a system that services them must run LPTSPL. A cluster printer must be started on this system also. For example, if system SYSA sends print requests to system SYSB, the operator gives the following command from SYSA:

  OPR>START PRINTER CLUSTER unit number NODE SYSB<RET>

  where:

  unit number designates a printer at SYSB

  Refer to the TOPS-20 KL Model B Installation Guide and the TOPS-20 Operator's Guide for further information on installing and enabling cluster printing.

3.7.2 Defining DQS and LAT Printers

To specify a DQS or LAT printer with the PRINT/REMOTE-PRINTER: command, users must first define that printer with the SET REMOTE-PRINTING PRINTER command, which indicates where the printer resides in the network. It can provide a useful alias for the printer as well. Users can issue the command interactively at the terminal or from a jobwide command file.

More conveniently, however, users can invoke the command for each available printer from a systemwide command file that you create. You should name this file REMOTE-PRINTING.CMD and place it in the SYSTEM: area. Users can invoke the commands in this file with the TAKE command or the SET REMOTE-PRINTING SYSTEM-DEFINITIONS command. (You can also place these commands in <SYSTEM>COMAND.CMD if all users on a system are expected to want to use this facility.) Here is a sample systemwide printer-definition file:

```
SET REMOTE-PRINTING PRINTER LN03    SWE$LN03 SYSA
SET REMOTE-PRINTING PRINTER LISTING SI$8700  SYSB
SET REMOTE-PRINTING PRINTER LATOP   LC14     LAT99
```

The SET REMOTE-PRINTING PRINTER command has three arguments. The first argument assigns an alias to the remote printer. The second argument is the name of a VMS printer queue (SWE$LN03) or a LAT port or service (LC14) or the name of an existing alias (SET REMOTE-PRINTING PRINTER LAT03 LN03). The third argument is the name of the system or LAT server that controls the printer.

Refer to the TOPS-20 Commands Reference Manual for a complete description of the command.

The REMOTE-PRINTING.CMD file can also contain specifications for DQS printing characteristics, as described in section 3.7.3.

3.7.3 Setting DQS Printing Characteristics

Users can specify how they want their DQS printed output to appear. For example, they may desire a "landscape" or "portrait" page orientation, or 65 or 90 characters per line. They can specify these and other characteristics with the PRINT/CHARACTERISTICS command. However, the printing characteristics must first be established with the SET REMOTE-PRINTING CHARACTERISTIC command. Like the printer definitions described in Section 3.7.2, the printing characteristics can be established by invoking the appropriate command from the terminal, a jobwide command file, or the systemwide command file, SYSTEM:REMOTE-PRINTING.CMD.

Here is a systemwide command file that contains printing characteristics and printer definitions:

```
SET REMOTE-PRINTING CHARACTERISTIC PORTRAIT 52
SET REMOTE-PRINTING CHARACTERISTIC P65       59
SET REMOTE-PRINTING CHARACTERISTIC P75       53
SET REMOTE-PRINTING CHARACTERISTIC P80       58
SET REMOTE-PRINTING CHARACTERISTIC P90       52
SET REMOTE-PRINTING CHARACTERISTIC P100      51
SET REMOTE-PRINTING CHARACTERISTIC P132D    54
SET REMOTE-PRINTING CHARACTERISTIC LANDSCAPE 0
SET REMOTE-PRINTING CHARACTERISTIC L100      50
SET REMOTE-PRINTING CHARACTERISTIC L132      0
SET REMOTE-PRINTING CHARACTERISTIC TR10P_BOLD 61
SET REMOTE-PRINTING CHARACTERISTIC TR14P_BOLD 62
SET REMOTE-PRINTING CHARACTERISTIC TR18P_BOLD 63
SET REMOTE-PRINTING CHARACTERISTIC CONDENSED 100
SET REMOTE-PRINTING CHARACTERISTIC OVERHEAD 101

SET REMOTE-PRINTING PRINTER   LISTING SI$8700   SYSA
SET REMOTE-PRINTING PRINTER   LN03    CS$LN03   SYSB
SET REMOTE-PRINTING PRINTER   LATOP   LC14      LAT99
```

The SET REMOTE-PRINTING CHARACTERISTIC command has two arguments. The first argument is the name of a characteristic, for example, PORTRAIT, or P90 (portrait orientation with 90 characters per line). The second argument is the numeric value that your site has assigned to that characteristic for the particular printer queue or to the name of an existing characteristic.

You can also place this command in the <SYSTEM>COMAND.CMD file if all users on a system are expected to want to use this facility.

Refer to the TOPS-20 Commands Reference Manual for a complete description of the command.

3.8 TERMINAL PRINTERS

Your site may have printers attached to dedicated, hardwired terminal lines, such as an LA50, LA100, LA120-RA, or LA120-RB. These local printers are called terminal printers. The operator starts them with the following command:

OPR>START PRINTER unit/DEVICE:TTYn/TERMINAL-CHARACTERISTIC:<RET>

where:

unit is the number that designates a printer

n is the terminal line number

You or a system programmer must assign a descriptive value to each of these printers in the LPTSPL module, LPTUSR.MAC. Instructions for doing this are included in the GALAXY documentation file. The argument that the operator gives to the /TERMINAL-CHARACTERISTIC: switch with the OPR>START PRINTER command must match a value in LPTUSR.MAC.

CHAPTER 4

CREATING STRUCTURES

4.1 OVERVIEW

One of the first decisions you must make about your new (or upgraded) installation is what type of disk storage environment best suits your needs. Some of the considerations that determine your decision are:

- How large will the data base be?
- How many users will be using the system?
- How experienced will these users be?
- Will there be a full-time operator?
- How often will you run diagnostics and how critical is it that the system remain available during this maintenance?
- Must all files be available to all users at all times during system operation?
- Are multiple systems part of a CFS configuration? Refer also to Chapter 12, The Common File System.

The mountable structure facility of TOPS-20 provides several options for making this decision. The option you choose depends on the answers to the previous questions and the number of disk packs and drives that are available. For example, if your installation has a number of disk packs and two or more drives, you can store data and program files on different structures.

A structure is a collection of data and program files contained on one or more disk packs and referenced under one name.

When you install your software, you create a structure known as the system structure (sometimes called the boot structure). All packs in this structure remain on-line at all times during system operation. If your system structure does not encompass all of your available drives you can create and mount other structures.

Sections 4.2 through 4.8 describe the system structure and how you can best utilize your disk resources and create and use other structures.

4.2 THE SYSTEM STRUCTURE

Sections 4.2.1 and 4.2.2 provide an overview of what the system structure is, including its relationship to the system and its contents.

NOTE

Unless you have enabled the "login structure" feature in a CFS-20 configuration, described in Section 12.6.2, the public, system, boot, and login structures all denote the same structure.

4.2.1 What Is the System Structure?

The system structure is the most important structure on your system. It is created and brought on-line at system installation when you answer the appropriate questions in the installation dialog. (Refer to the TOPS-20 KL Model B Installation Guide.) The name of the system structure can be up to six characters.

The system structure can be one or more disk packs, depending on the configuration of your system and your disk drive resources. You may use one or more RP06 or RP07 disks as the system structure; the maximum number of disks per structure is given in Table 4-4. You may NOT use an RA60, RA81, or RP20 as the system structure.

While installing the software, you copy the console front-end files to the system structure pack that is mounted on a dual-ported drive (usually drive 0). The dual port allows the front-end processor and the central processor to access the data on the system structure. However, if you are using an RP07 as the system structure, you must reserve space for the front-end files on an RP06 disk that is dual-ported to the PDP-11 front end. If the disk structure containing the front-end files has multiple packs, the first pack of the structure must be permanently mounted on the dual-ported drive.

All disk packs in the system structure must be online at all times, because this structure contains all the programs, files, and swapping area that the system needs to operate. The structure also contains all user directories necessary to support users logging into the system. (If the "login structure" feature is enabled in a CFS-20 configuration, the login structure contains these directories and must also be online at all times.) If the file system is destroyed on the system structure, or if a drive that contains all or part of the structure malfunctions, the system halts. Refer to Chapter 9 in this

manual and to the TOPS-20 Operator's Guide for the steps that you and the operator must follow if you have problems with the file system or if a drive goes down.

You can have another structure online that is capable of being used as the system structure. This structure must have a unique name, however, at least while it is mounted. (Section 4.5.2 provides information about mounting structures having the same name.) Section 4.5.5 discusses why you would have such a structure.

If you include the ENABLE JOB0-CTY-OUTPUT command in the n-CONFIG.CMD file, the operator is notified at the console terminal when disk space becomes low on the system structure.

If you are using CFS-20 software, refer to Chapter 12, The Common File System, for additional information on the system structure.

4.2.2 The Contents of the System Structure

The following list provides an overview of the contents of the system structure:

1. The default TOPS-20 command processor, EXEC, which is usually found in <SYSTEM> or <NEW-SYSTEM>.

2. A <ROOT-DIRECTORY> (Section 3.2.1) that points to the location on disk of all first-level directories on the system structure, including the special system directories.

3. All the files in the directories <SYSTEM> and <SUBSYS> (Sections 3.2.2 and 3.2.4).

4. The directories <NEW-SYSTEM>, <NEW-SUBSYS>, <ACCOUNTS>, <OPERATOR>, <SYSTEM-ERROR> and <SPOOL> (Sections 3.2.6 and 3.2.7).

5. The front-end monitor (RSX20F) and the console front-end files (Section 3.4). This normally appears in the <ROOT-DIRECTORY>. If you are using the RP07 as the system structure, the front-end file system must reside on an RP06 dual-ported disk drive.

6. The required swapping area. The size of this area depends on the TOPS-20 monitor you are using. For example, 2060-MONMAX uses up to 15,000 pages of disk space for swapping. (Refer to Section 4.7 for a description of the swapping area.)

7. A HOME block that contains the following parameters:

   - the structure's physical name
   - the number of disk packs in the structure
   - which pack this is in the structure
   - the address and number of pages used for the front-end file system (usually 950)
   - the address and number of pages set aside for swapping
   - the address of the <ROOT-DIRECTORY> and its backup copy
   - the serial number of the KL CPU to be booted from the structure

8. A directory for every user who requires access to the system. Users must log into a directory on the system structure to use the system. Afterwards, they can mount and connect to a different structure and directory. (If the "login structure" is enabled, CFS-20 users log in to the designated public structure.)

4.3 ONE-STRUCTURE SYSTEMS

A one-structure system consists of a single structure, the system structure, which is always on-line. All packs in the structure must be on-line for the system to operate.

Usually, a one-structure system has only one or two disk drives. Smaller TOPS-20 installations choose to keep all their directories and files on one structure for some of the following reasons:

- It is the simplest system.
- It is the easiest system to maintain.
- There is no requirement to physically remove packs from the drives, for example, for security reasons.
- The majority of users are inexperienced.
- All files are available at all times, and thus are easy to access.

- o A full-time operator may be unnecessary.
- o There is only one disk drive (only one structure supported).

Chapter 5 describes the methods you can use to create and maintain directories on your one-structure system.

4.4 MOUNTABLE STRUCTURES

If the system structure does not encompass all available disk drives, you can create and mount other structures on the unused drives. These "mountable" structures are created using the CHECKD program. The TOPS-20 Operator's Guide describes creating structures with CHECKD.

NOTE

The system structure is the only structure created at installation time. All other structures are created (using the CHECKD program) and brought on-line during system operation.

4.4.1 Differences Between Mountable and System Structures

Unlike the system structure, a mountable structure can be mounted and dismounted during timesharing. Also, it need not contain a front-end file system. Therefore, a mountable structure does not have to reside on a dual-ported disk drive. Although a mountable structure has its own <ROOT-DIRECTORY> and directory system, a user cannot log into a mountable structure, but must log in as a user on the system structure. A user can then mount a different structure and connect to directories. Table 4-1 summarizes the differences between a mountable and system structure.

4.4.2 Similarities Between Mountable and System Structures

There are, however, many similarities between the system structure and mountable structures. Both contain user directories and files. A mountable structure can have a front-end file system, and can be used in place of the system structure to load the system for timesharing. A mountable structure is created with the eight special directories (mentioned in Chapter 3) as for a system structure. Likewise, a mountable structure has a HOME block that contains information such as the name of the structure and the number of disk units in the structure. These and other similarities are summarized in Table 4-2.

Table 4-1: Differences Between Mountable and System Structures

System Structure	Mountable Structures
Always up during timesharing	Can be mounted and dismounted
Has a front-end file system (unless an RP07)	Need not have a front-end file system
Resides on a drive that is dual ported with the front-end computer (unless an RP07)	Need not reside on a dual-ported disk drive
Used for logging into the system	Cannot be used for logging into the system
Belongs to the system	Can belong to a private user
Has the <SYSTEM>, <SUBSYS>, <ACCOUNTS>, <OPERATOR>, <SYSTEM-ERROR>, and <SPOOL> directories	Need not have these directories unless the structure will be used as the public structure; can be deleted from the structure
Must contain a swapping area	Need not contain a swapping area unless the structure is to be mounted as the public structure

Table 4-2: Similarities Between Mountable and System Structures

System Structure	Mountable Structures
Has a HOME block	Has a HOME block
Has a front-end files area (unless an RP07)	Can have a front-end files area
Is used to load the system	Can be used to load the system if the proper file areas and files have been established
Contains system files	May contain system files

System Structure	Mountable Structures
Has a <ROOT-DIRECTORY>	Has a <ROOT-DIRECTORY>
Contains user files	Contains user files
All packs must be on-line for the system to operate	All packs in the structure must be on-line to use the structure

4.5 MULTIPLE-STRUCTURE SYSTEMS

A multiple-structure system consists of a system structure and one or more additional structures. Figure 4-1 illustrates a system with three disk drives and two structures. The two-pack system structure MASTER: must be online during timesharing. The one-pack mountable structure ADMIN: can be removed during timesharing. Another one-pack structure can be mounted in its place.

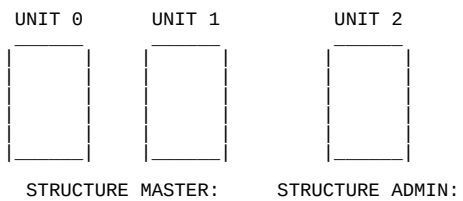


Figure 4-1: System with 3 Disk Drives and 2 Structures

Using Figure 4-1, suppose you want structure ADMIN: to remain on-line at all times during system operation. The structure is automatically mounted if you turn on the drive that contains the structure before the system is brought up. The TOPS-20 Operator's Guide describes mounting structures automatically.

In addition to the system structure and perhaps another permanent on-line structure, you may choose to keep one or more disk drives available for users to mount and dismount "private" packs during timesharing.

Figure 4-2 illustrates a system with three disk drives and three one-pack structures, the system structure MASTER:, ADMIN:, and PROG:.

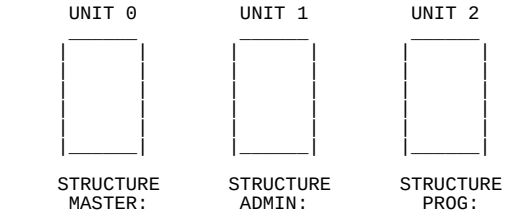


Figure 4-2: Three-Structure System

In this example, MASTER: contains all the directories necessary to support log-ins. ADMIN: contains the same, a superset, or a subset of the same directories as those on MASTER: and remains online at all times during system operation. The drive that contains PROG: is used for short-term mounting of different one-pack structures. PROG: remains online only for the time it is needed.

There can be up to 64 structures online at one time.

Several of the advantages in a mountable structure environment are:

- Some users or groups of users may require a structure exclusively for their use. They can 'own' or possibly pay for the use of certain structures.
- Service engineers can mount their own pack on the short-term drive and perform some diagnostics without disturbing normal system operation.
- Creating structures on mountable packs provides additional security to that already within the system. For example, you can create a structure that contains highly confidential data, remove it from the drive(s) when you are done with it, and lock it in a security cabinet or safe. At the end of the day, the operator locks up any confidential structures.
- In this type of environment, you are not limited in the size of your system's data base. You can create as many structures as you have disk packs to contain them, and you can mount as many at one time as your system can support.

The principal disadvantages of using mountable structures are the need for scheduling both access to the data and operator coverage to install and remove packs on the drives, as described in Section 1.2.2, Mountable Structure Sign–Up Log. There is also some risk that packs will be damaged during handling.

After the system is operating and structures have been created, the operator responds to requests from users to mount and dismount structures. Section 4.5.5 describes how to place user directories on your mountable structures to obtain maximum availability to priority jobs.

4.5.1 Choosing Structure Names

Each device on the system has a name, called the physical device name, which is used when giving commands to the software. Unlike the generic device name that applies to a class of devices, for example: TTY:, DSK:, LPT:, the physical device name applies to a particular device on the system; for example, TTY6: and LPT0:. The physical device names for disks are structure names. A structure name can be from one to six alphanumeric characters of your choice, and, like other device names, must be followed by a colon. The colon indicates to the software that a device is being used and not, for example, a file.

It is important to carefully assign a unique name to each structure that you create. Section 4.5.2, Mounting Structures Having the Same Name, explains why this precaution avoids confusion for users if an operator is unavailable during timesharing.

Because structure names are used in the device field (dev:) of a file specification, you should not create any structures with the same name as a defined (or valid) device name. Table 4–3 lists device names that may be defined in your system.

For the same reason, avoid naming a structure with a defined logical name, for example, SYS:, SYSTEM:, NEW:, OLD:, HLP:, etc., because the system searches the list of defined systemwide logical names before device names. (Refer to Section 3.3 for a description of logical names.)

Refer to Chapter 12, The Common File System, for structure–naming considerations in a CFS environment.

Table 4–3: Sample Device Names

DSK:	CDP:
MTAn:	FEn:
MTn:	TTY:
LPT:	TTYn:
LPTn:	PTYn:
PLPTn:	NUL:
CDR:	PLT:
PCDRn:	PLTn:
PCDPn:	DCN:
TCP:	SRV:

Where n is the unit number of the device.

To avoid duplication, you can get a list of all structure names known to the system by giving the operator command, SHOW STATUS STRUCTURE. These structures do not have to be online. Their presence in the listing indicates that they were previously specified in a SET STRUCTURE command, or once mounted on the system.

4.5.2 Mounting Structures Having the Same Name

A situation may arise requiring you to mount a structure that has the same name as a structure that is already online. Perhaps another installation has requested that you mount its system structure (named SYSA:) for testing purposes, but you already have a structure named SYSA: online. Because the system notices ambiguous structure names, you must mount the structure under a different name.

Each structure that is mounted is identified with two names: the physical identification and the alias. Usually, these names are the same. The physical identification is the actual structure name written in the HOME block of that structure. The alias is the name that you use to reference the structure while it is mounted. After a structure is mounted,it is known only by its alias. The MOUNT command is used to mount a structure and give it an alias different from the physical identification. This allows two or more structures with the same physical name to be mounted simultaneously. The system distinguishes them by their different aliases. (The TOPS-20 Operator's Guide describes this procedure.)

Note that the structures must be mounted one at a time. That is, structures cannot be online simultaneously before the MOUNT command is given. One of the structues must be mounted first. It may be necessary to power down disk drives and bring them up again.

4.5.3 Maximum Size of Structures

The maximum size of a structure is approximately 805,680 pages. A structure of this size requires 3 RP20 disk drives (5 spindles).

Structures of the maximum size, however, may not be practical for your installation. Smaller structures enhance the reliability and availability of the system. Remember that you can have up to 64 structures on-line at one time.

Also, if a structure is contained on more than one disk pack, the packs and drive units for that structure must all be the same type, that is, either all RP06s, RP07s, RP20s, RA60s, or RA81s. For example,

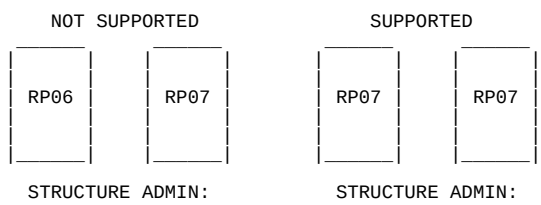


Table 4-4 shows the maximum structure size using RP06, RP07, RP20, RA60, and RA81 disk drives. For all drive types, there can be 12,000 directories per structure and 5,000 files per directory.

NOTE

The number of directories per structure and files per directory that can be created is approximate. This is because the disk space needed to create a directory or file varies according to the lengths of the names chosen for directories and files.

Table 4–4: Maximum Size Structures

Type of Disk Drive	Max.No.Packs Per Structure	No. Pages Per Pack
RP04	6	38000
RP06	3	76000
RP07	1	216376
RP20	5	201420
RA60	6	90516
RA81	5	200928

4.5.4 Increasing the Size of Structures

You can add more disk packs to increase the size of a mountable structure (not the system structure) during timesharing. To do this, you must:

- Dump the entire file structure onto magnetic tape using the DUMPER program
- Run the CHECKD program, specifying the new configuration, to re–create the structure
- Restore all the directories and files from magnetic tape using the DUMPER program

IMPORTANT

If possible, re–create the structure and restore the files to a different set of packs from the structure that you dumped. This precaution ensures that you do not lose any valuable data should you have problems reading the tape back to disk. That is, you still have the original structure intact and can rerun DUMPER and copy the structure to another tape.

To increase the size of the system structure, you must shut down the system and follow the installation procedure for bringing up this structure with more disk packs. Refer to Chapter 9, System Problems/Crashes, and to the TOPS–20 KL Model B Installation Guide.

4.5.5 Setting Up Structures for Maximum Availability

Before you create structures and place user directories on them, you should determine which users must be on the system at all times. Place these users' directories and files on the system structure, or on another structure that is always available during timesharing. Divide the remaining users of the system by priority, and place their directories and files on the other structures. Although these users have log–in directories on the system structure, their large working area where they create and store files is on the other on–line structures. You may want to help users set up their LOGIN.CMD and BATCH.CMD files so that they can mount, connect, and access the appropriate directory on the structure where their files are located, if it is not the system structure. Dividing users into categories and placing them on structures accordingly ensures that the failure of one disk drive does not prevent the most important users from using the system. For example, if the drive that contains ADMIN: goes down, you can remove the ADMIN: pack from the broken drive, and mount it on another drive that contains a less critical structure.

Also, on–line disk diagnostics can be performed during timesharing. Sometimes, the service engineer can dismount a non–critical structure, mount the maintenance pack, and perform preventive maintenance or trouble–shooting with only a portion of the user community off–line.

To increase system availability, you can create another system structure for backup using the CHECKD program. After you create this structure, you should follow the procedures in your software installation manual for creating the front–end files area. The TOPS–20 Operator's Guide describes using the CHECKD program to create a backup system structure. If you have problems with the primary system structure, having a second system structure available allows you to resume timesharing without reinstalling the system.

The backup system structure can be mounted and online at all times under another name, or it can be kept in storage and mounted as a backup if the regular system structure is destroyed. If the backup system structure is kept in storage, the operator must update the structure periodically with the System Backup Tape and the latest incremental dumper tapes. (Chapter 7, System Backup, describes creating and using your System Backup Tape and incremental tapes.) Occasionally updating your backup system structure (in storage) keeps it reasonably up–to–date.

If you keep your backup system structure online at all times, and you have important files that are constantly accessed by the user community, you can improve your system performance by placing these files on the backup system structure. Now your swapping area and the files that you access frequently are not on the same disk. This procedure is useful with any structure that you keep on-line at all times.

If multiple systems are part of a CFS configuration, refer to Chapter 12, The Common File System, for further discussion of placement of files and user directories.

4.5.6 Taking Structures Off-Line

When a structure must be taken off-line, the operator should notify users that it will be dismounted at a certain time. Users should give the DISMOUNT command for the structure before the specified time. If the users do not cooperate, the operator can dismount the structure (via the DISMOUNT command to OPR) without leaving files in an unknown state. Files that are open simply become inaccessible, and the user who had the files open receives an error.

For information on dismounting structures in a CFS environment, refer to Chapter 12, The Common File System.

4.5.7 Mounting Structures from Another Installation

If you mount a structure from another installation, or perhaps a structure that contains confidential data, some of the user names on this structure may match the user names on your system structure. You must mount this structure in what is called a FOREIGN state, to avoid the mishap of your users accessing directories that do not belong to them. The same is true if you bring one of your structures to another installation. You should have the operator at the installation SET the structure FOREIGN and then mount it. Figure 4-3 illustrates this concept.

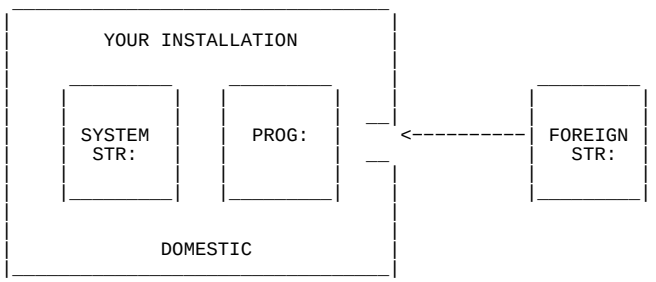


Figure 4-3: Domestic and Foreign Structures

A structure is brought online in one of two states, DOMESTIC or FOREIGN, according to the setting that the operator last specified for this structure with the SET STRUCTURE command. The system uses the FOREIGN state as the default if a SET STRUCTURE command has never been given for this structure. The structure remains in the specified state until the operator changes the state with the SET STRUCTURE or the UNDEFINE command. Note that the setting is unchanged across system crashes and reloads.

You should bring a structure online as DOMESTIC only if the directories on that structure were created for the same people as those on the system structure. One can be a subset of the other, but a given directory name should represent the same person on both. Conversely, you should bring a structure on-line as FOREIGN if the directories on that structure were not necessarily created for the same people as those on the system structure. This is because a user who is logged into a directory on the system structure is the owner of an identically named directory on a DOMESTIC structure, and can give the CONNECT or ACCESS command to that directory without giving a password (provided the directory protection allows this type of access for the owner, which is the usual case). However, a user who logs into the system structure and gives the CONNECT or ACCESS command to a directory with an identical name on a FOREIGN structure must give the associated password.

4.6 SHARING STRUCTURES (DISK DRIVES) BETWEEN TWO SYSTEMS

If you have two DECSYSTEM-20s and one or more structures that contain data common to both of these systems, you may want to set up your system to share disk drives alternately. For example, you could allow System A to use the drive that contains structure ADMIN: in the morning and allow System B to use this structure on the same drive in the afternoon. THE SYSTEMS CANNOT, HOWEVER, ACCESS THE DRIVE AT THE SAME TIME. (Refer to Chapter 12, The Common File System, for the exception.) Also, if one of your systems goes down, you can still use the drive that is connected to both systems.

A drive that is to be shared by two systems must be supported by both systems. For example, you cannot connect an RM03 disk drive to a DECSYSTEM-2020 and a DECSYSTEM-2065 because the 2065 system does not support RM03s. Also, the shared drive must be dual-ported. Your field service representative must make the appropriate connections from each DECSYSTEM-20 to a port on the disk drive. Be sure to have the field service representative tell you which system is connected to which port on the drive. Figure 4-4 illustrates this connection.

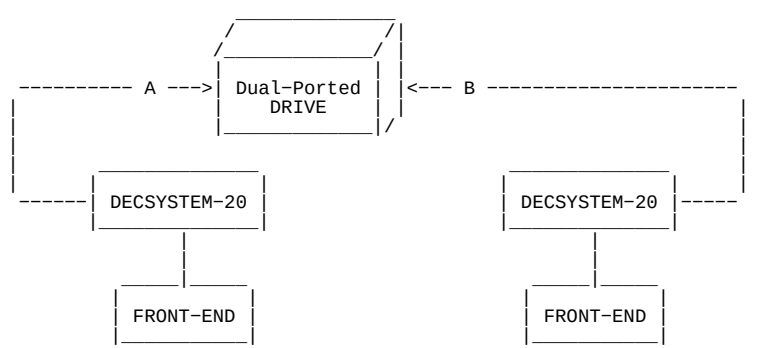


Figure 4-4: Shared Disk Drive

The port switch on this drive must be in either the A or B position, unless the systems are part of a CFS configuration. Otherwise, TOPS-20 will not permit either system to access the disk while it is in the A/B position. Error messages will be generated.

To use the drive, place the port switch in the position that corresponds to the first system that is using the drive (A or B). The operator mounts a structure using the normal procedure. After the first system is no longer allowed to use the drive, the operator gives the DISMOUNT command for the structure.

To use the drive on the second system, the operator leaves the pack on the drive (if you are using the same structure), turns the drive off-line, changes the port switch to the corresponding system, and turns the drive back on. Then, the operator (or a user) gives the MOUNT command to mount the structure on this system. The system automatically recognizes that another drive is on-line and mounts the structure.

4.7 DETERMINING SWAPPING SPACE ON THE SYSTEM STRUCTURE

Sections 4.7.1 and 4.7.2 describe what swapping space is and how to determine the amount of swapping space that you should allocate for your system.

4.7.1 What Is Swapping?

The number of user processes that can fit into main memory simultaneously depends on the size of the individual processes, the size of the memory-resident portion of the monitor, and the size of memory physically available. (Only a portion of the TOPS-20 monitor resides in main memory at one time.) If a user wishes to run a process that is not currently in memory, process space must be provided. This may necessitate moving some other process out of memory. The user's program or data that is transferred out of memory is placed on disk in the swapping area. The system sets aside a portion of the disk storage space on the system structure specifically for this purpose.

On some timesharing systems, a program must be entirely in main memory to execute. Swapping then consists of moving entire programs between disk and memory. Under TOPS-20, only portions of a program (those containing the instructions and data currently being referenced) need be in memory. Other portions of the program are brought into memory from disk as they are needed. In this case, swapping consists of moving portions of a program or data between disk and memory. The monitor decides which portions of which programs to swap, and when.

The size of a program is measured in a unit called a page. When swapping occurs, some of these pages are copied between memory and disk.

4.7.2 When to Increase Swapping Space

For the most part, the size of your swapping space depends on the cumulative size of processes you estimate will be on the system at any one time. Table 4-5 contains guidelines for estimating the amount of swapping space required for an approximate number of user jobs based on typical requirements. This amount is given in response to the question, "HOW MANY PAGES FOR SWAPPING?" in the software installation procedures.

The actual disk space used for swapping depends on the number of pages you give. The system rounds the number of pages given upward to an integral number of cylinders. The swapping space is divided equally among the disk packs in the system structure.

You can allow for swapping space on structures other than the system structure. However, this is necessary only if you plan to mount the structure as the system structure in the future. Allocating swapping space avoids re-creating the structure should you decide to mount it as the system structure.

All the monitors are designed to default to an appropriate number of pages for swapping. In most cases, you can take this default.

The guidelines in Table 4-5 apply to systems whose users perform many editing jobs and an average or small amount of debugging programs and production jobs. If your users perform a great number of debugging and production jobs and only a small amount of editing, you should double the size of your swapping space. However, if you double the size of your swapping space, check the maximum swapping space allowed for the monitor you are running. (The TOPS-20 KL Model B Installation Guide lists the maximum number of swapping pages you can use with each monitor.) You cannot exceed this maximum. If you enter a number that is larger than the maximum, the monitor uses the maximum allowed. If you must exceed the maximum, you can bring up a larger existing monitor, or you can tailor your monitor by following the instructions in the BUILD.MEM file. This file is located in the documentation files saveset on the TOPS-20 Software Distribution tape.

Table 4-5: Determining Swapping Space

Estimated Number of Jobs	Recommended Number of Pages for Swapping*
20 or less	3000
21 to 30	4500
31 to 40	6000
41 to 50	7500

* For each additional 10 jobs, increase the number of pages for swapping by approximately 1,500.

If disk space is available, it is better to overestimate the swapping space needed. If not enough swapping space has been reserved, system service may be disrupted.

If you include the ENABLE JOB0-CTY-OUTPUT command in the n-CONFIG.CMD file, the operator is notified at the console terminal when swapping space becomes low.

4.8 DETERMINING THE AVAILABLE DISK SPACE

4.8.1 Determining Disk Space Before Installation

To determine the available disk space that you will have to divide among your users before installing the system, first calculate the swapping space required by your system (Section 4.7.2). Second, insert the number you calculated for swapping space into the formula shown in Table 4–6, and perform the appropriate steps.

Table 4–6 outlines how to calculate the available disk space on the system structure. If you are calculating the available disk space on other structures, follow this same procedure but eliminate reserving space for any directories or areas that are not on that structure. If any possibility exists that a structure may be used as the system structure, reserve the swapping space.

Note that if you plan to enable the "login structure" feature in a CFS–20 cluster (see Section 12.6.2), much more swapping space is available on the system structures. This is because user directories will reside on the shared login structure.

NOTE

Remember that as your system expands, the number of pages in the <SYSTEM> and <SUBSYS> directories increases. Also, the number of pages reserved for directory <SPOOL> should be increased if: (1) you maintain large operator log files (2) users copy large numbers of files or large files to LPT:. A large backlog of user file retrieval requests can also use up much of the <SPOOL> area.

Table 4–6: Calculating Available Disk Space

TOTAL DISK SPACE:			
Number of RP06 disk drives *	76000 pages per drive	=	_____ TOTAL DISK SPACE
_____OR:_____			
Number of RP07 disk drives *	216376 pages per drive	=	_____ TOTAL DISK SPACE
_____OR:_____			
Number of RP20 disk drives *	201420 pages per spindle	=	_____ TOTAL DISK SPACE
_____OR:_____			
Number of RA60 disk drives *	90516 pages per drive	=	_____ TOTAL DISK SPACE
_____OR:_____			
Number of RA81 disk drives *	200928 pages per drive	=	_____ TOTAL DISK SPACE
RESERVED DISK SPACE:			
Front–end file system	= 950		
Swapping Space (Enter number of pages selected and allocated for swapping)	=		
<SYSTEM>	= 1876		
<SUBSYS>	= 1780		
<NEW–SYSTEM>	= 2		
<NEW–SUBSYS>	= 2		
<OPERATOR>	= 500		
<UETP.*>	= 1701		
<ROOT–DIRECTORY>	= 9		
<SPOOL (you should reserve)>	= 1000		
			_____ TOTAL RESERVED SPACE
SUBTRACT TOTAL RESERVED SPACE FROM TOTAL DISK SPACE			_____ AVAILABLE DISK SPACE

4.8.2 Determining Disk Space After Installation

Shortly after you have installed the system, you can log in as OPERATOR and give the command INFORMATION (ABOUT) DISK-USAGE. One line of output tells you the actual number of system pages that are available on the system structure. You can divide this disk storage among your users. (Refer to Chapter 5, Creating Directories.) However, be careful about over-allocating disk space on the system structure. If the space allowed to user directories exceeds the space free on the system structure after installation, then users can fill up the entire system structure. But if TOPS-20 cannot free up adequate space by expunging the system structure, then system service may be disrupted. Even if space can be freed to allow the system to keep running, some user programs may be disrupted.

CHAPTER 5

CREATING DIRECTORIES

A prospective user who requires access to the system must be assigned a user name (normally the user's surname), a password, an account, and disk storage quotas, and must have a directory created for him or her on the public structure. Optionally, you can assign certain capabilities and/or make the user or the user's directory a member of one or more groups. (Refer to Section 5.8 for a description of how to establish group relationships among users and Section 5.9 for a description of the capabilities you can assign to users.) You can also create additional directories for users on mountable structures.

This chapter describes three methods you can use to create and maintain directories. Using one method, the operator creates and maintains all the directories on the system. A second method allows you to delegate the responsibility for creating and maintaining directories to project administrators. The third method combines the first two methods, thus providing additional flexibility. Sections 5.1 through 5.3 explain these methods and the determining factors for choosing one of them. These sections also include some of the decisions necessary to assign user names and capabilities, and how to allocate disk storage according to the method you use to create and maintain directories. (Refer to Chapter 4 to determine the amount of disk space available to divide among the directories you create.)

Chapter 6 describes how to set up an accounting scheme and how to assign and validate accounts. You should read Chapter 6 if you want to allow users to log into the system and charge their computer usage to valid accounts immediately after you have created their directories.

Refer to Chapter 12, The Common File System, for considerations in creating directories in a CFS-20 environment.

5.1 HAVING THE OPERATOR CREATE AND MAINTAIN ALL DIRECTORIES (CENTRAL CONTROL)

In this type of installation, the operator creates a directory on the public structure for each new user and specifies the appropriate parameters. The name of the directory is the same as the assigned user name. Each user informs the operator when a change to his directory parameters is required. This type of operation allows you as system manager to have central administrative control over all directories and parameters. Therefore, central control means that you or the operator create and maintain all the directories for all your system users. Central control has two types of directory schemes. One scheme allows you to create up to approximately 5,000 directories per structure. The other scheme allows you to use subdirectories and create up to approximately 12,000 directories per structure.

NOTE

The number of directories allowed per structure is approximate, because the disk space needed to create a directory or file varies.

5.2 DELEGATING THE CREATION AND MAINTENANCE OF DIRECTORIES TO PROJECT ADMINISTRATORS (PROJECT CONTROL)

An alternative type of installation involves project administrative control. Under this type of control, the operator creates directories only for the users who have been designated as project administrators, (e.g., the representatives of major departments). The project administrators, in turn, create subdirectories for users within their departments or projects and control the assignment of those users' directory parameters. This type of control allows you to delegate the responsibility for creating and maintaining directories for other users and still maintain ultimate control over your system and its resources.

Therefore, project control means that most of the directories created by you or the operator are project directories (e.g., MATH might be the assigned name of a project). The system's resources, such as disk space, are divided among these project directories either equally or according to the expected size of the project. Subdirectories are then created under project directories for users within the project. The people who have been appointed project leaders or administrators are responsible for creating, assigning parameters to, and maintaining the subdirectories within their project. The resources that you allocate to the project directory are divided among its subdirectories by the project administrators. Under project control, you are allowed to create up to approximately 12,000 directories (including subdirectories) per structure.

5.3 COMBINING CENTRAL AND PROJECT CONTROL

A combination of central and project control can be used if you want to keep the majority of the user directories at the management level of control and separate only a portion of your system into projects and administrative control.

Therefore, combining central and project control means that the operator creates and maintains directories for most of the system users and creates project directories for special projects. The project administrators create directories under the project directories and are responsible for maintaining them. Combining the two types of control still allows up to approximately 12,000 directories per structure.

5.4 CENTRAL AND PROJECT CONTROL DESCRIPTIONS

Sections 5.4.1 through 5.4.4 describe the two types of central control, project control, and the combination of central and project control. Each description includes:

- The determining factors for choosing a particular control
- The directory format for each type of control
- The procedure for assigning user names
- The procedure for creating user directories
- The procedure for creating files–only directories
- The restrictions, if any, that apply to using a particular control

Also, any additional considerations that apply to headings within each description are included.

Read each description thoroughly. The first central control description contains very general information and suggestions that apply to all the directory schemes.

5.4.1 Central Control

DETERMINING FACTORS:

- Your business installation is relatively uncomplicated; therefore, there is no need to separate projects and assign the creation and control of directories to various administrators. All the directories on the system are created and maintained by you or the system operator.
- You are sure that the number of directories you need is less than approximately 5,000.

FORMAT:

<ROOT–DIRECTORY> can point to approximately 5,000 directories per structure.

All directories under <ROOT–DIRECTORY> are on one level.

ASSIGNING USER NAMES:

The user name that you assign should include the user's last name. This convention is true for any type of directory scheme that you use. The system uses this name when recording the authors of files, sending mail to users, and displaying system status. If you follow this convention, you can easily identify who is using the system when you give a SYSTAT command. If just using last names will not result in duplications, you can simply use the last names. Otherwise, you might want to use the first and middle initials followed by the last name. (If a user has no middle initial, you can use a dash in its place.) It is most convenient for users when user names begin with unique characters, allowing use of "recognition" when typing user or directory names. Use of leading initials often yields this result.

CREATING USER DIRECTORIES:

All directories are created using the ^ECREATE command. (Only users who have WHEEL or OPERATOR capability enabled can use this command.) In the next example, the operator is connected to the public structure PUBLIC: and uses the ^ECREATE command to create a new directory named <BECKER> for the user who has been assigned the user name BECKER. Also, the operator assigns the password MARTIN.

```
@ENABLE (CAPABILITIES)<RET>
$^ECREATE (NAME) PUBLIC:<BECKER><RET>
[NEW]
$$PASSWORD MARTIN<RET>
$$<RET>
$DISABLE (CAPABILITIES)<RET>
@
```

This directory is called the user's logged-in directory and is always on the public structure. Whenever the user logs into the system, he is connected to this directory. He can remain in this directory or connect to and use files in another directory.

Refer to the TOPS-20 Operator Command Language Reference Manual for a complete description of the ^ECREATE command that the operator uses to create new directories, and the ULIST program that prints information about all the directories on the system.

After creating a new directory (either files-only or user), remember to update the tape containing the monitor, TOPS-20 Command Processor, DLUSER, DLUSER data, DUMPER, <SYSTEM>, and <SUBSYS>. (Refer to Chapter 7, System Backup Procedures.)

CONSIDERATIONS:

If two users have mistakenly been assigned the same user name and you try to create the second directory with this duplication, the system prints [OLD] instead of [NEW]. Give the ABORT subcommand, assign the user a slightly different user name, and reissue the ^ECREATE command with the new directory name. A common practice is to precede such names with the user's first initial. This allows recognition on the user or directory name without typing the entire name and the distinguishing character. For instance, if you have the two users Stephanie Sheldon and Andrew Sheldon, you should assign them the user names S-SHELDON and A-SHELDON or SSHELDON and ASHELDON rather than use the names SHELDON-S and SHELDON-A.

CREATING FILES-ONLY DIRECTORIES:

If a user wants to have a library area in addition to his logged-in directory, you can create a files-only directory on the public structure or on another structure. The user can gain owner privileges to this directory by giving the CONNECT command and the password associated with the directory. If the directory is located on a regulated mountable structure,the user must also give the MOUNT command to use the structure before he gives the CONNECT command. The user cannot give the ACCESS command for or log into a files-only directory. For example, if you have a user named BECKER who processes payroll on a regular basis, he may want to develop the payroll programs in his directory and keep the payroll data in a more restricted directory. To accomplish this, you can create on the public structure the logged-in directory <BECKER> and the files-only directory <PAYROLL>. The directory <PAYROLL> can be on some other structure, (e.g., ADMIN:<PAYROLL>). BECKER now has normal protection on his directory, more restrictive protection on the directory <PAYROLL> and can still CONNECT to <PAYROLL> by giving its password. BECKER cannot, however, give the ACCESS command for or log into <PAYROLL>.

The next example shows how to create the directory <PAYROLL> on the public structure MAIN:.

```
@ENABLE (CAPABILITIES)<RET>
$^ECREATE (NAME) MAIN:<PAYROLL><RET>
[NEW]
$$PASSWORD MONIES<RET>
$$FILES (ONLY)<RET>
$$PROTECTION (OF DIRECTORY) 774000<RET>
$$DEFAULT (FILE) PROTECTION 770200<RET>
$$<RET>
$DISABLE (CAPABILITIES)<RET>
@
```

Now if user BECKER logs in and wants to use the files in <PAYROLL>, he can give the following CONNECT command:

```
@CONNECT (TO DIRECTORY) <PAYROLL><RET>
Password:  monies<RET>
```

The TOPS-20 Operator Command Language Reference Manual describes all the parameters you can give to directories and describes how to create directories on mountable structures.

CONSIDERATIONS:

When you create additional directories on mountable structures, consider if files-only directories are suitable. Some users will not want to give the CONNECT command to a directory each time they require owner access to the files in that directory. Also, files-only directories are members of groups only as directory group members and not user group members. Therefore, if you create ALL the directories on a structure as files-only, you cannot establish any valid user group relationships among those directories. (Refer to Section 5.8 for a description of setting up groups.)

Conversely, if you create user directories, users can give the ACCESS command to their additional directory and gain owner and group privileges without connecting to the directory, and they can use other directories on the structure as group members. Also, if the name of the directory you create is the same as the user's logged-in directory, and the structure is mounted as DOMESTIC, the user does not have to specify a password when giving the CONNECT or ACCESS command to the directory. (Refer to Section 4.5.7.) This is valuable when the user is submitting a batch job. No password is required on the batch input; therefore, security is preserved. In addition to creating user directories on the structure, you can create files-only directories to be used as library areas.

It is also possible to create only one user directory on a structure and to create all other directories as files-only. In this case, all users required to use a files-only directory on the structure could give the ACCESS command to the user directory (gaining owner and group privileges), and use the files in a files-only directory according to the group protection codes set. This would be useful if you have a private structure that contains several library areas that are common to the owners of the private disk pack(s). Each owner could give the ACCESS command for the one user directory and gain group privileges to all the library directories. Therefore, these users would need only one password to gain access to all the information on the pack.

Note that defined groups may provide better security controls than passwords. If the password for the ADMIN:<PAYROLL> directory is MONIES, it might be guessed, or user BECKER may write it down or tell it to some other user. On the other hand, group membership can be centrally controlled, and the access can be withdrawn, if necessary. Also, the directory structure provides a record of group memberships, which can be displayed with the ULIST program. Wise use of directory protections can allow user members of a group to connect to a files-only directory without giving a password.

Refer to the TOPS-20 User's Guide or the TOPS-20 Commands Reference Manual for a complete description of the CONNECT and ACCESS commands.

RESTRICTIONS:

The number of directories you create per structure cannot exceed approximately 5,000. This number is an approximation because the disk space that it takes to create a directory or file varies.

5.4.2 Central Control Using Subdirectories

DETERMINING FACTORS:

- o As stated in the previous directory scheme, your installation does not warrant segregation of projects and control. However, this directory scheme allows more directories per structure than the previous central control scheme. You or the operator can create up to approximately 12,000 directories per structure and assign and maintain all the directory parameters.

- o You can easily expand into a form of project control by adding project directories, and still maintain control at the management level over the majority of the user directories. (Refer to Section 5.4.4, Combined Central and Project Control.)

FORMAT:

<ROOT-DIRECTORY> points to 26 directories. The name of each directory is a letter of the alphabet, <A>, <B>, <C>, ..., <Z>. The directories point to all user and files-only directories. The single-letter directories are on one level below <ROOT-DIRECTORY>. The user and files-only directories are on a second level below <ROOT-DIRECTORY> and are pointed to by the alphabetic directories.

Also, the directories pointed to by the single letter directories, (e.g., <A.JONES>), can be allowed to create directories under them. Perhaps user A. JONES wants to create one or two subdirectories to store special files, such as memos. The user is responsible for maintaining the directory he created and is allowed to use only the disk quota you originally allocated to his logged-in directory. Refer to Section 5.4.3, Project Control, if you would like to allow some users to create directories of their own.

ASSIGNING USER NAMES:

Each user name that you assign should be as close as possible to the user's last name prefixed by a first initial and a period. For example, Charles Baker would be assigned the user name C.BAKER. Under this type of directory scheme, you must follow the principle of prefixing the name with the first initial and a period.

CREATING USER DIRECTORIES:

Have the operator create 26 directories using the ^ECREATE command. The name of each directory is a letter of the alphabet, that is, <A> through <Z>.

The theory behind creating these alphabetic directories is the same as described in Section 5.4.3. That is, you must create directories that are allowed to have subdirectories. The directories <A>, <B>, ..., <Z> can have approximately 5,000 user and files-only directories under them. Therefore, you must include some of the same parameters in these directories, as you would in project directories.

Refer to the TOPS-20 Operator Command Language Reference Manual for a complete description of the parameters that are defined when the operator uses the ^ECREATE command to create directories, and the ULIST program that prints information about all directories on the system.

The procedures for creating the alphabetic directories and the user and files-only directories under them are described below. In the example, COMMON: is the name of the public structure.

NOTE

The general considerations described in Section 5.4.1 for creating directories are also applicable to this directory description.

Even though the alphabetic directories are not associated with users, they must be created as log-in directories. However, do not assign passwords. This prevents users from gaining access to these directories.

```
@ENABLE (CAPABILITIES)<RET>
$^ECREATE (NAME) COMMON:<A><RET>
[NEW]
$$
```

Assign each directory a large number for creating subdirectories, for example, 400.

```
$$MAXIMUM SUBDIRECTORIES (ALLOWED) 400<RET>
```

Because many user directories are created under each of these alphabetic directories and the page quota (disk space) from these alphabetic directories is divided among (or passed on to) the user directories, you must assign the alphabetic directories a very large permanent and working page quota. Assigning them a sufficiently large page quota prevents any of these alphabetic directories from exceeding their page quota, thus requiring you to make a change to the quota at a later time. Therefore, assign each directory at least 500,000 pages of permanent and working disk page quota.

```
$$PERMANENT (DISK STORAGE PAGE LIMIT) 500000<RET>
$$WORKING (DISK STORAGE PAGE LIMIT) 500000<RET>
```

Assign a list of SUBDIRECTORY-USER-GROUP numbers to each directory. The list should be the same for each directory. The range of numbers you use depends on how many groups you plan to establish, (e.g., 5 groups, 10 groups, 40 groups). The numbers used in the following examples are for illustration; you can choose any sequence:

```
$$SUBDIRECTORY-USER-GROUP (ALLOWED) 200<RET>
$$SUBDIRECTORY-USER-GROUP (ALLOWED) 201<RET>
$$SUBDIRECTORY-USER-GROUP (ALLOWED) 202<RET>
$$SUBDIRECTORY-USER-GROUP (ALLOWED) 203<RET>
$$SUBDIRECTORY-USER-GROUP (ALLOWED) 204<RET>
```

Later, when you create a user directory and place that user in a user group, you must enter one of the numbers in this list. No other numbers will be valid. (Refer to Section 5.4.3, for a more detailed description of why you are using this list of numbers, and Section 5.8 for establishing valid group relationships.)

In the following example, the operator is connected to the public structure COMMON: and creates the first two directories, <A> and <B>:

```
@ENABLE (CAPABILITIES) <RET>
$^ECREATE (NAME) COMMON:<A><RET>
[NEW]
$$MAXIMUM-SUBDIRECTORIES (ALLOWED) 400<RET>
$$WORKING 500000<RET>
$$PERMANENT 500000<RET>
$$SUBDIRECTORY-USER-GROUP (ALLOWED) 200<RET>
$$SUBDIRECTORY-USER-GROUP (ALLOWED) 201<RET>
$$SUBDIRECTORY-USER-GROUP (ALLOWED) 202<RET>
$$SUBDIRECTORY-USER-GROUP (ALLOWED) 203<RET>
$$SUBDIRECTORY-USER-GROUP (ALLOWED) 204<RET>
$$<RET>
$
$^ECREATE (NAME) COMMON:<B><RET>
[NEW]
$$MAX 400 !You can use abbreviations<RET>
$$WORKING 500000<RET>
$$PERMANENT 500000<RET>
$$SUB 200<RET>
$$SUB 201<RET>
$$SUB 202<RET>
$$SUB 203<RET>
$$SUB 204<RET>
$$<RET>
$ !Continue creating directories <C>
$ !through <Z> in exactly the same manner.
```

After all 26 directories are created, you can give the DIRECTORY command to see all the directory files that have been created under <ROOT-DIRECTORY>.

```
$DIR COMMON:<ROOT-DIRECTORY>

 COMMON:<ROOT-DIRECTORY>

A.DIRECTORY.1
B.DIRECTORY.1
C.DIRECTORY.1
     .
     .
     .
Z.DIRECTORY.1
```

Next, after all 26 alphabetic directories have been successfully created, the operator again uses the ^ECREATE command and creates all the user directories.

In the example below, the operator is connected to the public structure COMMON: and creates a directory named <A.JONES> for the user who has been assigned the user name A.JONES. This directory is A. Jones' log-in directory on the public structure. Each time A. Jones logs into the system, he is connected to directory <A.JONES>. In this example, the operator also assigns the password 2BY4. The operator gives the directory the system default of 250 pages for both working and permanent disk quota. Because he is using the default, he does not have to make any entries for these two parameters. The 250 pages given to this directory are taken from the superior directory's quota (directory <A>). (The PRESERVE subcommand can be issued to avoid having the disk space quota subtracted from the superior directory's allocation.) The operator also places user A.JONES in user group 202 and places directory <A.JONES> in directory group 202.

```
@ENABLE (CAPABILITIES)<RET>
$^ECREATE (NAME) COMMON:<A.JONES><RET>
[NEW]
$$PASSWORD 2BY4<RET>
$$USER-OF-GROUP (NUMBER) 202<RET>
$$DIRECTORY-GROUP (NUMBER) 202<RET>
$$<RET>
$DISABLE (CAPABILITIES)<RET>
@
```

After creating any new directories (either files-only or user), you should update the backup tape that contains the monitor, TOPS-20 Command Processor, DLUSER, DLUSER data, DUMPER, <SYSTEM>, and <SUBSYS>. (Refer to Chapter 7, System Backup Procedures.)

CONSIDERATIONS:

If users have duplicate first initials and last names, you can use middle initials. For example, if two users have the name C.BAKER, you can assign either one of them a user name in the form CL.BAKER or C.LBAKER. If you use the form CL.BAKER you must create a directory <CL> in addition to directory <C>. If you do not create this additional directory with the user's first and middle initial, you will receive error messages and will not be able to create the directory <CL.BAKER>. If, instead, you assign user Baker the user name C.LBAKER (the preferred method), you can create the directory <C.LBAKER> as described above using the standard procedure. You do not need to create the additional directory <CL>. Alternatively, you could create two levels of "initial" directories, and assign users to third-level directories based on their first and middle initials followed by their last names, for example, C.L.BAKER.

If a user requires special capabilities to perform privileged functions, the operator can include the parameter for the capability in the user's directory accordingly. (Refer to Section 5.9 for a description of the capabilities you can assign to certain users who require them.)

CREATING FILES-ONLY DIRECTORIES:

If a user wants a library area in addition to the logged-in directory, you can create a files-only directory.

Files-only directories can also be prefixed by a letter and a period. Because the first name initials of users do not always encompass every letter in the alphabet, you may want to use those infrequently used letters as the prefix to the files-only directories, e.g., <X.FORLIB> or <Z.TESTS>. Alternatively, you can place the files-only directories under a special prefix, such as LIB., or place them directly under the root directory. The example below shows how to create the directory <X.FORLIB> on the public structure PUBLIC:. The operator assigns the password SQUASH, makes the directory files-only, takes the 250-page default for working and permanent disk storage quotas, and places the directory in directory group number 202. (User members of groups 202 can now accesss this directory and the files it contains according to group protections.)

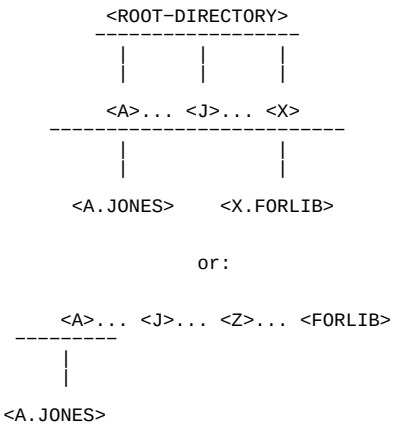
```
@ENABLE (CAPABILITIES)<RET>
$^ECREATE (NAME) PUBLIC:<X.FORLIB><RET>
[NEW]
$$PASSWORD SQUASH<RET>
$$FILES-ONLY<RET>
$$DIRECTORY-GROUP (NUMBER) 202<RET>
$$<RET>
$DISABLE (CAPABILITIES)<RET>
@
```

Follow the procedures in the TOPS-20 Operator's Guide for creating directories on mountable structures.

CONSIDERATIONS:

The CONSIDERATIONS described in the previous central control description (Section 5.4.1) for files-only directories also apply to this description.

If the number of files-only directories you want to create is small, you can create them on the same level as the alphabetic directories. That is, <X.FORLIB> can be created as <FORLIB>. Your directory scheme may look like:



or:

RESTRICTIONS:

The number of directories you can create per structure cannot exceed approximately 12,000.

The number of subdirectories under a single-letter directory, for example, <A>, cannot exceed approximately 5,000.

If you reach the maximum number of directories allowed per structure, the system prints the message:

?MAXIMUM DIRECTORY NUMBER EXCEEDED -- INDEX TABLE NEEDS EXPANDING

If you reach the maximum number of subdirectories that a single letter directory can point to, the system prints the message:

?SUPERIOR DIRECTORY FULL

You can define up to only 40 user groups with the ^ECREATE command, because all the user groups must be specified as subdirectory user groups in the superior single-letter directory. This is not as great a problem when all the user directories are directly under the ROOT-DIRECTORY.

If you make a superior directory FILES-ONLY, be sure to make all its subdirectories FILES-ONLY. Otherwise, you will be unable to recreate the subdirectories (refer to Chapter 9) if the structure is damaged, until all the superior directories are made not FILES-ONLY.

5.4.3 Project Control

DETERMINING FACTORS:

- The complexity and perhaps geography of your organization warrants separating small or large groups of users into projects. The responsibility for creating and maintaining the directories within a project can be given to an administrator. This is especially helpful if a large number of users have directories on the system. This method frees the operator from spending an excessive amount of time creating directories and changing directory parameters.

- Even though you delegate the task of creating and managing groups of directories to project administrators, you still maintain ultimate control of the overall system and its resources. This means that you still determine and allocate the disk space that each project uses. The administrator distributes the disk space you allocate to directories within the project. Also, administrators can create and maintain directories for their projects without having WHEEL or OPERATOR capabilities by using the TOPS-20 BUILD command. Therefore, you do not weaken the security of your system. Unless you give WHEEL or OPERATOR capabilities to an administrator, he cannot assign those capabilities to other users.

- In addition to allowing administrators to create directories for a project, you can allow other users of the system to create subdirectories. These users can separate and store files in a subdirectory. According to the protection they place on their subdirectories, they can share their files with other users without losing the security of their superior directory. The users are responsible for maintaining the directories they create.

- Up to 12,000 directories (including subdirectories) can be created per structure.

FORMAT:

<ROOT-DIRECTORY> can point to approximately 5,000 directories per structure.

Each directory under <ROOT-DIRECTORY> can point to approximately 5,000 subdirectories.

Each subdirectory can also point to approximately 5,000 subdirectories directly under it.

The number of subdirectory levels is determined by a maximum length of 39 alphanumeric characters, because each subdirectory name contains the name or names of any superior directories above it. For example, the user who owns directory <PHYSICS> under <ROOT-DIRECTORY> creates the subdirectory <PHYSICS.LAB-12>. The new subdirectory name (LAB-12) has its superior directory's name (PHYSICS) as its prefix. The period separates the different levels of the directory name and is counted as one of the characters in the directory name.

ASSIGNING USER NAMES:

The names that you assign to users should be as close as possible to the user's last name. In addition, the project names that you assign and that will be used for project directory names should be closely related to the project, e.g., PHYS might be used for Physics and PHYED for Physical Education.

When you give a SYSTAT command, the user surnames and obvious project names make it easier to identify who is using the system, and under which project.

CREATING PROJECT AND USER DIRECTORIES:

The user and project directories that <ROOT-DIRECTORY> points to (first-level directories) are created by you or the operator using the ^ECREATE command.

The procedures you should use and the parameters that you must include in these directories are described below.

Create all project directories as log-in (user) directories. You would not create a project directory as files-only, because files-only directories cannot have log-in directories created under them. However, log-in project (or user) directories can have both log-in and files-only subdirectories.

Assign a disk storage quota to each project directory. This quota must be large enough to accommodate both the files that are contained in the directory and the directories that are created under it. Each time a directory is created under a project directory, that directory's disk quota is taken from the project directory's disk quota. The total disk quota for directories created under a project directory cannot exceed the quota originally given to the project directory.

In the example below, the operator begins to create the project directory <CHEM>. He creates the directory as a log-in directory on the public structure ORANGE: and assigns the password H20. This procedure allows an administrator to log into the directory, giving its password, and create the required subdirectories. He may also want to store his files in this directory. The operator gives the directory a 10,000-page working and a 10,000-page permanent disk storage quota.

```
@ENABLE (CAPABILITIES)<RET>
$^ECREATE (NAME) ORANGE:<CHEM><RET>
[NEW]
$$PASSWORD H20<RET>
$$WORKING 10000<RET>
$$PERMANENT 10000<RET>
$$
```

Next, the operator enters the parameter that allows the owner of the project directory to create subdirectories. This parameter, called MAXIMUM-SUBDIRECTORIES (ALLOWED), specifies how many directories can be created under the directory. Unless you enter this parameter (the default is 0), the owner of the directory cannot create subdirectories. For example, all users of the system can type the BUILD command to the TOPS-20 Command Processor, but only those users who have the MAXIMUM-SUBDIRECTORIES (ALLOWED) parameter in their directory with a number greater than zero can actually use the BUILD command to create subdirectories.

The following entry in the sample project directory <CHEM> allows the administrator to create 100 subdirectories.

```
$$MAXIMUM-SUBDIRECTORIES (ALLOWED) 100<RET>
```

The administrator who is responsible for this sample project might create 60 directories under the project directory and give each subdirectory approximately 50 pages of working and permanent disk quota. He keeps enough pages in the project directory to allow that directory's files to grow and to create additional subdirectories. (Refer to the TOPS-20 Commands Reference Manual for the description of the BUILD command, including distributing working and permanent storage quotas and maximum subdirectory quotas.)

Also, some of the MAXIMUM-SUBDIRECTORIES (ALLOWED) quota given to the project directory can be given to a subdirectory so that directories under it can be created. The quota for the project directory is decremented by the amount of quota given to the subdirectory.

For example, directory <CHEM> is given a subdirectory quota of 100. The administrator creates the directory <CHEM.STUDENT> under <CHEM> and gives the directory a subdirectory quota of 10. The number of subdirectories that can now be created under <CHEM> is 89. If the administrator creates another subdirectory under <CHEM> called <CHEM.STUDENT2> and gives that directory a subdirectory quota of 6, the number of subdirectories that can now be created under <CHEM> is 82.

If the administrator gives an INFORMATION (ABOUT) DIRECTORY <CHEM> command, the output line for maximum subdirectory quota is:

```
MAXIMUM NUMBER OF SUBDIRECTORIES ALLOWED 84
```

The two directories <CHEM.STUDENT> and <CHEM.STUDENT2> that were created under <CHEM> account for the two subdirectories not shown in the subtraction.

Next, the operator enters the parameter that allows the administrator for this project to place users in groups. The administrator can use the group facility as described in Section 5.8 to set up library directories and allow file sharing among members of the project.

The SUBDIRECTORY-USER-GROUP parameter accepts a number between 1 and 262143 as its argument. You can list a range of numbers that the administrator can use to establish groups within the project; however, you must enter each number separately. Be careful to assign a range of numbers that is unique to that project. For example, project directory <CHEM> may be given the range:


```
$$SUBDIRECTORY-USER-GROUP (ALLOWED) 2600<RET>
$$SUBDIRECTORY-USER-GROUP (ALLOWED) 2601<RET>
$$SUBDIRECTORY-USER-GROUP (ALLOWED) 2602<RET>
$$SUBDIRECTORY-USER-GROUP (ALLOWED) 2603<RET>
$$SUBDIRECTORY-USER-GROUP (ALLOWED) 2604<RET>
```

Project directory <PHYSICS> may be given the following range of numbers different from project CHEM.

```
$$SUBDIRECTORY-USER-GROUP (ALLOWED) 3001<RET>
$$SUBDIRECTORY-USER-GROUP (ALLOWED) 3002<RET>
$$SUBDIRECTORY-USER-GROUP (ALLOWED) 3003<RET>
$$SUBDIRECTORY-USER-GROUP (ALLOWED) 3004<RET>
$$SUBDIRECTORY-USER-GROUP (ALLOWED) 3005<RET>
```

If you assign the same range of numbers to different projects, you can cause a security break among projects. For example, a user in group 2602 in project CHEM should not be able to access, as a group member, the directories and files in project PHYSICS.

The range of numbers placed in a project (or user) directory's parameter list does not imply that the directory or any of its subdirectories has access to those groups. It means only that the administrator (or owner of the directory) can use those group numbers to establish group relationships among that directory and its subdirectories.

The following example shows the completed parameter list for the sample project directory <CHEM>:

```
@ENABLE (CAPABILITIES) <RET>
$ ^ECREATE (NAME) ORANGE:<CHEM><RET>
[NEW]
$$PASSWORD H20<RET>
$$WORKING 10000<RET>
$$PERMANENT 10000<RET>
$$MAXIMUM SUBDIRECTORIES (ALLOWED) 100<RET>
$$SUBDIRECTORY-USER-GROUP (ALLOWED) 2600<RET>
$$SUBDIRECTORY-USER-GROUP (ALLOWED) 2601<RET>
$$SUBDIRECTORY-USER-GROUP (ALLOWED) 2602<RET>
$$SUBDIRECTORY-USER-GROUP (ALLOWED) 2603<RET>
$$SUBDIRECTORY-USER-GROUP (ALLOWED) 2604<RET>
$$<RET>
$ DISABLE (CAPABILITIES) <RET>
@
```

Refer to the TOPS-20 Operator's Guide for a complete description of the ^ECREATE command that the operator uses to create new directories, and the ULIST program that prints information about all the directories on the system.

After creating a new directory (either user or files–only), remember to update the backup tape that contains the monitor, TOPS–20 Command Processor, DLUSER, DLUSER data, DUMPER, <SYSTEM>, and <SUBSYS>. (Refer to Chapter 7, System Backup Procedures.)

CONSIDERATIONS:

If two projects or users have mistakenly been assigned the same name, and you try to create the second directory with this duplication, the system prints [OLD] instead of [NEW]. Give the ABORT subcommand, assign the user or project a slightly different name, and reissue the ^ECREATE command with the new directory name.

A subdirectory is just like any other directory. It can be logged into (if it is not specified as files–only), it can be a member of user and directory groups, and it obeys the usual protection mechanisms. Therefore, there are no implied rights between a directory and its subdirectories, or between two subdirectories of the same directory. Files have three protection fields: owner, group, and world; and each directory has the same three protection fields. Refer to Section 5.7 for a description of directory and file protections.

The only additional rights that the owner of a directory has over that directory's subdirectory is the power to change its parameters (e.g., directory protection, password, or group memberships), or to use the KILL subcommand, which deletes the subdirectory.

If you or another user choose to delete a directory, you must first delete any subdirectories under the directory. You cannot delete a directory or subdirectory that has existing subdirectories. This protection insures that someone (possibly an administrator of a project) does not accidentally delete a directory that points to a large portion of the database. The operator or administrator must connect to the directory immediately above the lowest level subdirectory to begin deleting any directories. For example, if the owner of the directory <PHYSICS> wants to delete the directory <PHYSICS.LAB–12>, he must first connect to directory <PHYSICS.LAB–12> and delete any of its subdirectories. Then, he connects to directory <PHYSICS> and gives the KILL subcommand to delete directory <PHYSICS.LAB–12>. Note that the operator is the only person who can delete the directory <PHYSICS>.

If you or the administrator choose to grant special capabilities to a user, you can include the parameter for the capability in the user's directory. (Refer to Section 5.9 for a description of the capabilities you can assign to certain users.) You should instruct the administrator to inform you when special capabilities are given to a system user. You are protected against users randomly giving other users special capabilities, because the operator or the administrator who assigns special capabilities to a user must have (as a user) those same capabilities. A person with WHEEL or OPERATOR capabilities can assign any capability to another user. Also, the user or operator who is assigning the capabilities must have those capabilities enabled at the time the privileged parameter is entered into the user's directory.

Once a SUBDIRECTORY–USER–GROUP number has been allocated to a project directory, be careful about removing it. If it is in use in any of the subdirectories, either as a USER–OF–GROUP number or as a SUBDIRECTORY–USER–GROUP number, you will be unable to recreate the subdirectories (refer to Chapter 9) if the structure is damaged, until you manually restore the SUBDIRECTORY–USER–GROUP number into all the superiors.

CREATING FILES–ONLY DIRECTORIES:

Administrators or users can have library areas in addition to their logged–in directories. They can use the BUILD command and create files–only directories under their logged–in directories, provided you have given them the capability to do so by adding the MAXIMUM SUBDIRECTORIES (ALLOWED) parameter to the directory that will contain the subdirectories.

CONSIDERATIONS:

Refer to the CONSIDERATIONS under the first description of Central Control, Section 5.4.1. These considerations also apply to Project Administrative Control.

RESTRICTIONS:

- You cannot exceed approximately 12,000 directories per structure.
- The number of directories that a superior directory points to cannot exceed approximately 5,000.

If you reach the maximum number of directories that you can create on a structure, the system prints the message:

?MAXIMUM DIRECTORY NUMBER EXCEEDED –– INDEX TABLE NEEDS EXPANDING

If either you or an administrator reach the maximum number of directories that can be created under a superior directory, the system prints the message:

?SUPERIOR DIRECTORY FULL

- Files–only directories cannot have log–in subdirectories. If you want to allow a user to create user (log–in) subdirectories under his directory, you must make his directory a log–in directory.

5.4.4 Combined Central and Project Control

DETERMINING FACTORS:

- Only a portion of your organization warrants being separated into projects. The directories for the majority of the user community are created and maintained at the central management level. But, where project administration is appropriate, the task of creating and managing directories within a project is given to administrators.
- For example, if your company has groups of users with terminals in several distant locations, you may want to have the administrator at the remote location create and maintain all the directories for that site. You can create a project directory for the remote location, perhaps using the name of the site as the project directory name (for example, <CHICAGO> or <CHIC>, <SEATTLE>, ...). The remaining user directories at the central location are created by the system operator.

FORMAT:

<ROOT–DIRECTORY> points to all the project directories and 26 alphabetically named directories, <A> through <Z>. The project directories point to the user and files–only directories that an administrator creates for a given project. The directories <A> through <Z> point to user and files–only directories created and maintained by the operator. These directories can also be allowed to have subdirectories.

ASSIGNING USER NAMES:

If you create any user directories that are pointed to by <ROOT–DIRECTORY>, assign project names and user names in the same manner as described under Project Control, Section 5.4.3. Again, assign the 26 directories that will point to the majority of the user directories, the names <A>, <B>, <C> .... <Z>. The user names that will be the directory names under the alphabetic directories should, as previously stated, be the user's surname prefixed by a first initial and a period. (Refer to Section 5.4.2, Central Control Using Subdirectories.)

CREATING USER AND FILES–ONLY DIRECTORIES:

Create the user, files–only, and <A> through <Z> directories by following the instructions in Section 5.4.2.

Create the project directories according to the instructions in Section 5.4.3, and distribute the description of the BUILD command (TOPS–20 Commands Reference Manual) to the administrators who are responsible for creating the user directories within their project.

CONSIDERATIONS:

All the considerations that apply to both Central and Project Control also apply to combining the two types of control.

You may want to allow users whose directories are created by the operator to create several directories under their logged-in directories. For example, user A.SMITH creates the subdirectory <A.SMITH.MEMOS> to store files that he wants to keep separate from his programming files. This user uses the BUILD command to create the number of subdirectories that he is allowed to create and divides the quota for his logged-in directory among the directories he creates.

In general, users can store files in these directories or, if they set the appropriate protection, can share the files in these directories with other users.

RESTRICTIONS:

Combined Central and Project Control allows up to approximately 12,000 directories per structure.

The number of directories that a superior directory can point to cannot exceed approximately 5,000.

If you reach the maximum number of directories per structure, the system prints the message:

?MAXIMUM DIRECTORY NUMBER EXCEEDED -- INDEX TABLE NEEDS EXPANDING

If you reach the maximum number of directories that a superior directory can point to, the system prints the message:

?SUPERIOR DIRECTORY FULL

5.5 ALLOCATING DISK STORAGE QUOTAS

In Chapter 4 you determined the amount of disk space that is available on the system structure after installation. Once you know the available disk space, you can decide how to allocate it among the directories you create. Each directory is given a number of pages for both working-storage and permanent-storage allocations. Working storage refers to the disk space that a user can have during the time he is logged-in. Permanent storage refers to the total disk space that a user can have to store files after he has logged-out.

The number of pages that you should assign to directories depends on whether you (or the operator) are creating all the directories on the system (central control) or you are delegating the task of creating and maintaining directories to project administrators (project control). When using central control, you may divide the disk space equally among directories, giving regard to special requirements of certain users. In the case in which the operator creates project directories, you should allocate a disk quota large enough to accommodate the expected size of each project. Remember that project directories must distribute their disk space to the directories created under them.

Several important points about working and permanent allocations are discussed below.

Assign a large (2000-3000 page) working-storage allocation to users who perform considerable sorting because the temporary files required for this operation can occupy substantial disk space.

As the number of users on the system increases and your disk space on the public structure becomes low, you can decrease the working-storage and permanent allocations on the public structure to add new log-in directories. If you have additional disk drives not used by the public structure, you can accommodate the directories with many or large files by creating other structures and directories. Users will log into their directories on the public structure, request the operator to mount the proper structure using the MOUNT command, and access their additional directories with the ACCESS and/or CONNECT command. Note that in setting up the system, it is easier to accustom users from the start to use other structures than to reorganize the structures and retrain users after space has run out on the public structure.

5.6 ENFORCING DISK STORAGE QUOTAS

Working-storage allocations are strictly enforced. Users cannot exceed their working-storage allocations unless they enable WHEEL or OPERATOR capabilities. (Refer to Section 5.9 for a description of the special capabilities that can be given to users who require them.) If users request additional space, you can increase their allocations as required.

If a user exceeds his working-storage allocation and attempts to create or change a file, the system prints the following error message:

?QUOTA EXCEEDED The user must decrease his disk usage to less than the working-storage allocation for the directory (in which the file is being changed or created) before he can create or change any more files.

The system informs a user if he is over this permanent storage allocation when he logs off the system or connects to a different directory. The system prints the following message after the CONNECT or LOGOUT commands:

<directory> OVER PERMANENT STORAGE ALLOCATION BY nn PAGES

This message reminds users that although they may not be over their working-storage allocation, they have exceeded their expected total disk usage. Users should delete any files that are unnecessary for their job. Also, because permanent quotas are not enforced, it is wise to instruct the operator or administrator to police each directory's disk usage. The operator should run the CHKPNT program daily to keep a record of each directory's disk usage. The TOPS-20 Operator's Guide contains the description of running the CHKPNT program. If you are using the file migration facility (refer to Chapter 8), you may want to run the REAPER program with the TRIM command to force users to stay below their permanent quotas.

Every time the available disk space on the system structure is less than 500 pages, the system prints the following warning messages:

13-Jun-98 10:20:33 Disk space low on structure STR:, 122 free

[STR: Deleted files will be expunged from structure STR: in 30 seconds]

After 30 seconds, the system starts expunging any deleted files in all directories on the structure mentioned in the warning message.

The system prints this message when the expunging is complete:

[STR: Expunge of structure STR: completed]

The operator gets the following message:

13-Jun-89 10:59:59 Expunge completed for structure STR:, 1993 free

If anyone tries to create or change a file when there is no more disk space available, the system prints an error message similar to the one below:

?FILE OR SWAPPING SPACE EXCEEDED

Again, the operator or administrator should check to see how many users are over permanent allocations. Also, if you are using the file migration facility, you may want to migrate files on the system more frequently if you are constantly running low on systemwide disk space. (Refer to Chapter 8 for a description of file migration.)

5.7 PROTECTING DIRECTORIES AND FILES

Every directory and file has a protection number associated with it. The system uses a default protection number for each directory and file when the directory or file is created.

Whenever a user accesses a file, the system first checks the directory protection. If that protection allows the user the appropriate access to the directory, the system then checks the protection of the individual file.

5.7.1 Directory and File Protection Digits

The directory and file protection numbers have three 2-digit fields. The first field applies to the owner of the directory or file, the second field to members of the same group as this directory, and the third field to all other users (or world).

Protection Code

Owner	Group	World
dd	dd	dd

The default protection for directories and files is 777700. A directory or file protection of 77 in any given field allows full access. For example, the default protection allows the owner and members of his group full access but all other users no access.

Protection Code

Owner	Group	World
77	77	00

Table 5–1 contains a list of the directory protection digits.

Table 5–1: Directory Protection Digits

Digits	Privilege
04	Permits creating files in the directory.
10	Permits connecting to the directory without giving a password and changing the accounts and protection numbers of the files therein. Thus it gives many of the privileges the directory owner has. (Refer to the TOPS–20 Monitor Calls Reference Manual.)
40	Permits, subject to the protection on the individual file, listing the names of the files with the DIRECTORY command and reading the file, e.g., via the TYPE, PRINT, or LIST commands.

These protection codes are actually bits in a protection word. To get more than one protection, add the digits (octal) corresponding to the protection you want. Thus, 44 allows listing the files and creating new files. There are unused bits in the protection number; therefore, to provide complete access to files, use 77. Useful digit pairs are:

00 Permits no access.

40 Permits the files to be listed and read.

77 Permits full (owner) access.

A file protection number has the same format as a directory protection number, but the meanings of the digits are different. Table 5–2 contains a list of file protection digits.

Table 5–2: File Protection Digits

Digits	Privilege
02	Permits wildcarding of the file.
04	Permits appending to the file.
10	Permits executing the file.
20	Permits writing and deleting the file.
40	Permits reading the file.

Obtain a protection number by adding the file protection digits of the different protections you need. For example, protection number 775200 allows the owner full privileges; the members of the same group reading, executing, and directory listing privileges; and all other users no privileges. Useful digit pairs are:

00 Permits listing the file with the DIRECTORY command only if the file is specified explicitly and completely.

12 Permits executing and using the DIRECTORY command to list the file only.

This protection is useful when, for example, you purchase a program and agree in your contract not to allow any of your system users to read, write into, or copy the file. Set the protection on an execute–only file to 771212. The TOPS–20 Beware file provides additional considerations for setting up execute–only files.

52 Permits reading, executing, and using the DIRECTORY command to list the file.

77 Permits full access.

The system checks protection numbers starting with the two rightmost digits. Therefore, users do not restrict members of a group by assigning the file protection 770052, because the group gets at least the execute, read, and directory list access (52) granted to all users.

Also, because the system checks the directory protection before the file protection, files that have been given a low file protection are still secure in a directory with the default directory protection. For example, suppose the user KOHN tries to type the file EDIT.MAC in the directory <HESS>. The protection on the directory <HESS> is 777700 and the protection on the file EDIT.MAC is 777752. User KOHN and directory <HESS> are not in the same group, so the world protection applies. First, the system checks the directory protection, 777700. The last two digits (00) apply and permit no access to the directory. User KOHN is not allowed to type the file, even though the corresponding protection on the file (52) would allow the file to be read, executed, and listed with the DIRECTORY command if KOHN were allowed access to files in the directory.

5.7.2 Changing Directory and File Protection

Users can change file protection numbers via the SET FILE PROTECTION command or the RENAME command.

Users can change directory protection numbers via the SET DIRECTORY PROTECTION or BUILD command. You can, however, prevent users from making changes to their directory protection numbers by including the DISABLE DIRECTORY-PARAMETER-SETTING command in the system file called <SYSTEM>n-CONFIG.CMD on the system structure. If you make this entry in n-CONFIG.CMD, only users with WHEEL or OPERATOR capabilities can change directory parameters (via the ENABLE and SET DIRECTORY PROTECTION commands).

NOTE

Make an entry in n-CONFIG.CMD only if you DO NOT want to allow users to change their directory protections; otherwise, the system assumes that you want to use the system default command of ENABLE DIRECTORY-PARAMETER-SETTING. (Refer to the TOPS-20 KL Model B Installation Guide for a description of the parameters that are placed in the n-CONFIG.CMD file.)

5.8 ESTABLISHING GROUPS

You can let users share files by placing users and directories in groups. Members of a group can access directories and files in that group according to the middle digits of the directory and file protection code fields, as described in Section 5.7.1.

Each group that you establish has two types of members: USERS and <DIRECTORIES>. Each group is identified by a number. This number is included as one of the directory parameters in each directory belonging to the group. Any directory (including subdirectories) or user can belong to as many as 40 groups. You can set up group relationships in the individual directories by using the DIRECTORY-GROUP and USER-OF-GROUP subcommands to the ^ECREATE and BUILD commands. The following example shows that you have placed user Smith in user group 268 and directory group 418:

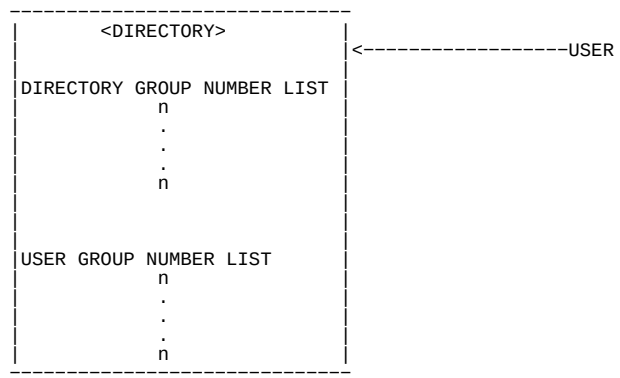
```
@ENABLE (CAPABILITIES)<RET>
$^ECREATE (NAME) MAIN:<SMITH><RET>
$$PASSWORD SOAR<RET>
$$WORKING 500<RET>
$$PERMANENT 500<RET>
$$USER-OF-GROUP 268<RET>
$$DIRECTORY-GROUP 418<RET>
$$
```

The DIRECTORY-GROUP or USER-OF-GROUP parameter that you place in the user's directory determines: 1) if this user can access another directory's files as a group member 2) if the files in this user's directory can be accessed by another user as a group member, or 3) both. The diagrams on the following pages illustrate the difference between being a member of a group as a directory and/or as a user.

When a user accesses a file in a directory that is a member of the same group, the system first checks to see if this user is the owner of the directory. When, in this example, it finds that the user is not the owner, the system then checks to see if the user is in the same group as this directory. In this case, the user and directory are in the same group; that is, the group numbers match. The system now checks the group protection code field of the directory being accessed. If the group protection allows the type of access that the user requested, the system proceeds to check the group protection on the individual file.

If you are setting up groups on different structures, there is no correlation between a group number on one structure and the same group number on another structure. For example, group 202 on MAIN: does not necessarily have the same user and directory members as group 202 on another structure.

Example

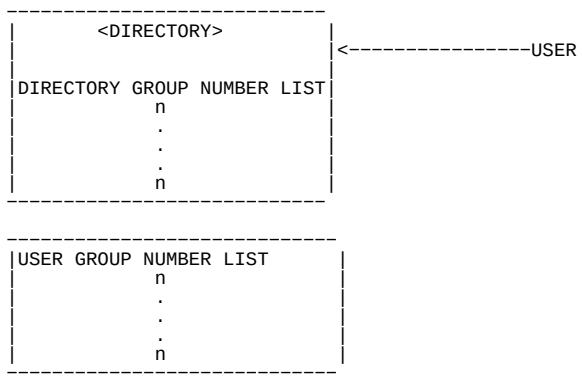


Each directory has two lists of group numbers: Directory Group Numbers and User Group Numbers.

Directory Group Numbers identify the various groups of which this <DIRECTORY> is a member.

User Group Numbers are associated with users and identify the various groups of which each user is a member.

Example



The Directory Group Numbers are important to users who require access to this directory. Those users who have a matching group number in the User Group Number List can access this <DIRECTORY> according to its group protection code.

The User Group Numbers are important to the owner of this directory. This owner can access any directory that has a matching group number in its Directory Group Number List. Note: Because files–only directories are not associated with a user, they do not contain User Group Numbers.

There are three common types of groups:

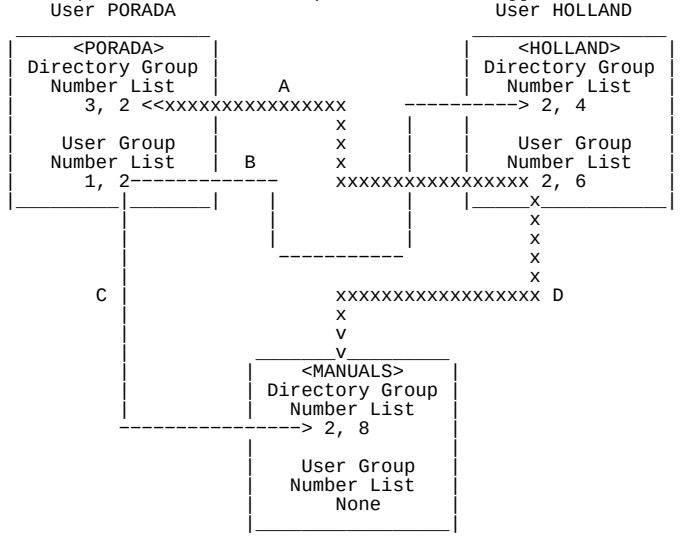
1. A file–sharing group, whose users can access a set of library directories and each other’s logged–in directories.

2. A library group, whose users can access a set of library directories and their own logged–in directories, but not each other’s logged–in directories.

3. A teacher–student group, in which the teacher can access the students’ directories and the students can access their own logged–in directories, but not their classmates’ directories or the teacher’s directory.

Figures 5–1 through 5–3 illustrate these three common groups and the association between USER and <DIRECTORY> members of a group.

In a file–sharing group (Figure 5–1), users share all their files according to the group protection field, both in the library directories (here it is <MANUALS>) and in their logged–in directories.



Legend:

1. A User HOLLAND can access directory <PORADA>
2. B User PORADA can access directory <HOLLAND>
3. C User PORADA can access directory <MANUALS>
4. D User HOLLAND can access directory <MANUALS>

Figure 5–1: File–Sharing Group

In Figure 5–1, the two users, PORADA and HOLLAND, are members of the same group (group 2). The directories <PORADA>, <HOLLAND>, and <MANUALS> are also members of group 2. Users PORADA and HOLLAND can access their own directory and files according to the owner protection code fields. PORADA can access directories <HOLLAND> and <MANUALS> according to the group protection code fields, and conversely, HOLLAND can access directories <PORADA> and <MANUALS> according to their group protection code fields. The other numbers shown in the figure indicate that a user or directory can be a member of more than one group.

In a library group (Figure 5–2), USER members can access all the <DIRECTORY> members but not each other's logged–in directories. The library directories are usually files–only directories. This figure illustrates a library group that consists of the files–only directories: <SUBROUTINES>, <TAPE–TESTS>, <MACROS>, and users: ALUSIC, BROPHY and KOHN. This library group illustrates that just because you are a member of a group as a user, your logged–in directory need not belong to the same group.

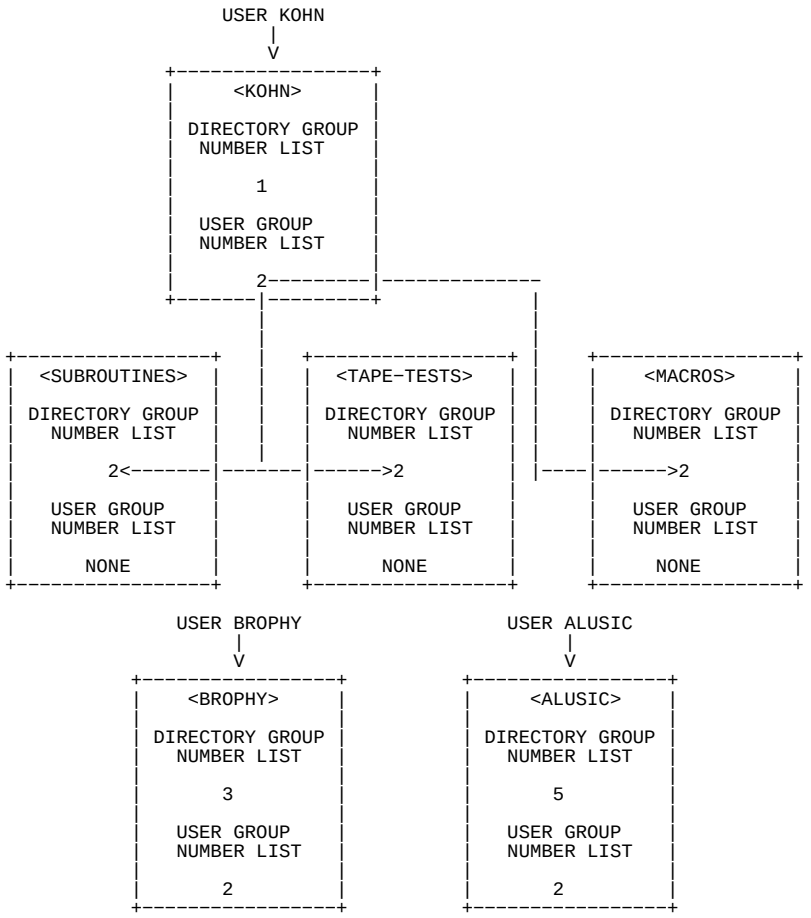


Figure 5-2: Library Group

In Figure 5-2, users KOHN, ALUSIC and BROPHY are not directory group members of the same group; however, they are all user group members in the same group (group 2). User KOHN can access directory <KOHN> according to the owner protection field and can access directories <SUBROUTINES>, <TAPE-TESTS>, and <MACROS> according to the group protection field. KOHN can access <BROPHY> and <ALUSIC> according to the "world" protection field. Although the arrows have not been drawn from users BROPHY and ALUSIC, their access privileges are the same as KOHN's.

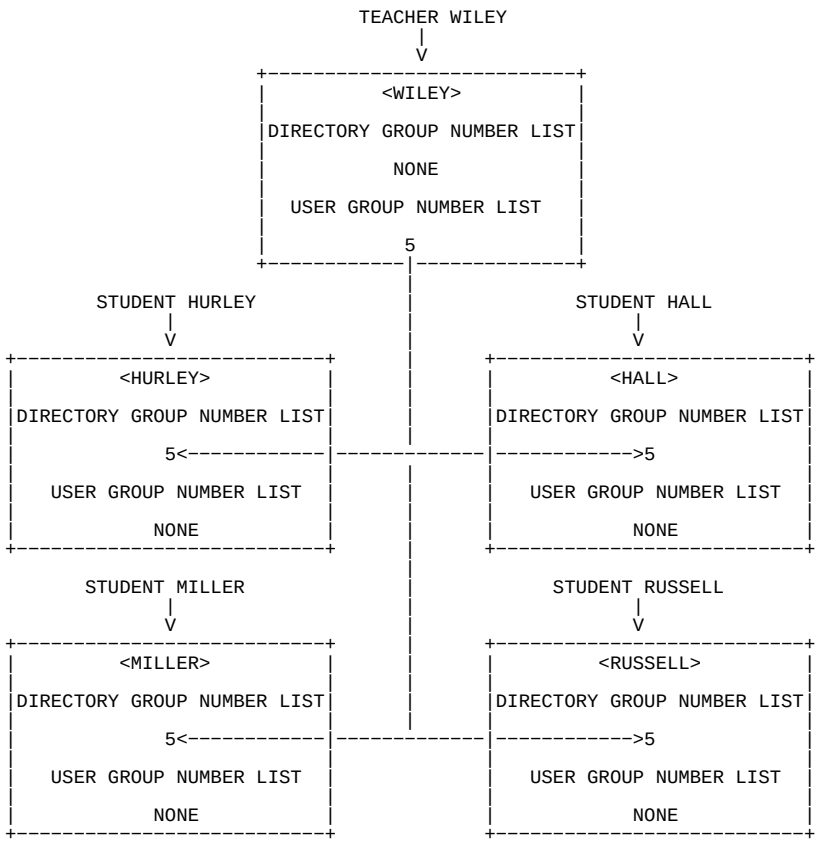


Figure 5–3: Teacher–Student Group

In a teacher–student group (Figure 5–3), the teacher, WILEY, is a member of the group as a USER, while the directories <HURLEY>, <HALL>, <MILLER>, and <RUSSELL> are <DIRECTORY> members. The teacher, WILEY, can access the files in the directories <HURLEY>, <HALL>, <MILLER>, and <RUSSELL> according to the group protection. The students whose logged–in directories are in this group as <DIRECTORY> members can access the files in <WILEY> according to the protection set for all users, because only their directories are members of the group; they are not members of the group as users.

5.9 GIVING USERS SPECIAL CAPABILITIES

You can give special capabilities to certain users; they are WHEEL, OPERATOR, SEMI–OPERATOR, CONFIDENTIAL, MAINTENANCE, IPCF, ENQ–DEQ, INTERNET–WIZARD, and ABSOLUTE–INTERNET–SOCKETS. Each capability that you give to a user is placed in the user's directory parameter list when you create or change the directory. The person who enters the capability in a user's directory must have that capability himself, and have it enabled at the time the capability is entered into the directory parameter list. You should grant these capabilities only to users who absolutely need them. Table 5–3 lists all the available capabilities and a brief description of their function.

Table 5–3: Special Capabilities

Capability	Description
WHEEL	Allows the user to modify any system parameters or data. In particular, the WHEEL capability is needed if the user wants to give the ^EEDDT or ^EQUIT commands.
OPERATOR	Allows the user all capabilities required to control the system. The user cannot, however, give the ^EEDDT or ^EQUIT commands.
SEMI–OPERATOR	Allows the user to execute only a subset of the operator commands –– to get system status information and to control certain devices.
CONFIDENTIAL	Allows the user to obtain accounting information for another user's job.

Capability	Description
MAINTENANCE	Allows the user (usually the field service representative) to perform certain maintenance functions, but he cannot give the ^E commands.
IPCF	Allows the user to perform the privileged functions of IPCF. (Refer to the TOPS-20 Monitor Calls Reference Manual.)
ENQ-DEQ	Allows the user to perform global ENQUEUE/DEQUEUE functions.
INTERNET-WIZARD	Allows the user to perform certain INTERNET privileged functions.
ABSOLUTE-INTERNET-SOCKETS	Allows the user to place absolute socket numbers in his programs.
INTERNET-ACCESS	Allows the directory owner to establish INTERNET network connections.
DECNET-ACCESS	Allows the directory owner to establish DECNET network connections.

With the exception of WHEEL and OPERATOR, these capabilities are not listed in a format where having one capability means you also have the capabilities listed below it. The user who has WHEEL capabilities can perform the OPERATOR, SEMI-OPERATOR, CONFIDENTIAL, MAINTENANCE, IPCF, and ENQ-DEQ functions. The user who has OPERATOR capabilities can also perform these privileged functions with the exception of the ^EEDDT and ^EQUIT commands. But the user who has CONFIDENTIAL capabilities can only perform functions that are allowed by this capability; that is, he cannot perform MAINTENANCE, IPCF, ENQ-DEQ, INTERNET WIZARD, and ABSOLUTE INTERNET SOCKETS functions, unless he has been given the individual capability. The same principle is true for the remaining capabilities.

Also, you are giving capabilities to a user, not the user's directory. Therefore, if user HALL has WHEEL capabilities, other users who connect to or access the directory <HALL> do not obtain WHEEL capabilities. However, if they log in as user HALL, they will obtain HALL's capabilities. For this reason, users with special capabilities (especially WHEEL and OPERATOR) should be especially careful in selecting and protecting their passwords. They should be encouraged to change them often, and to use passwords that cannot be readily guessed.

5.10 PRINTING DIRECTORY INFORMATION

The ULIST program prints information about directories on the system and is described in the TOPS-20 Operator's Guide. In addition to listing information about each directory, the ULIST program can list information about groups, capabilities, and related information.

The LIST subcommand of the ^ECREATE command prints information about the directory or user name you are currently creating, and the ^EPRINT command also prints the information on an individual basis.

CHAPTER 6

CREATING ACCOUNTS

The TOPS−20 accounting facility allows you to assign and charge computer usage to valid accounts. It provides you with a means for 1) adding security to your system, 2) determining charges for computer usage and billing users by account, and 3) associating classes with accounts for use by the class scheduler. You can use account validation for one or all of these reasons.

One or more accounts can be assigned to a user for specific tasks and validated each time they are used. All accounting data, including records of CPU time, structures used, and peripherals used under a valid account, are stored in a usage file and can be used later for reports and billing.

This chapter describes how to set up the system to use accounts and establish an accounting data base. The TOPS−20 USAGE File Specification describes how to create accounting reports from the Usage file and establish billing procedures. The following sections include:

- How to set up your system to use accounts
- How to select an accounting scheme
- How to use your accounting scheme and create the necessary base account and subaccount files
- How to run the account generator program (ACTGEN), which takes these account data files and creates the accounting data base
- What the operator can do if the accounting data base does not work properly
- How to initialize your system to start validating accounts

6.1 SETTING UP THE SYSTEM TO USE ACCOUNTS

6.1.1 Enabling or Disabling Account Validation

During software installation, you can specify whether you wish to create the account data base and validate accounts. You can make an entry in the n−CONFIG.CMD file that specifies either DISABLE ACCOUNT−VALIDATION or ENABLE ACCOUNT−VALIDATION. If you do not make an entry in the n−CONFIG.CMD file for accounting, the system assumes ENABLE ACCOUNT−VALIDATION.

If you enter DISABLE ACCOUNT−VALIDATION, meaning you do not wish to use the account validation facility, the system checks each account only for length. The purpose of the check is to ensure that the maximum number of alphanumeric characters has not been exceeded in each account. No other checking is performed. If a user attempts to use or create an account greater than 39 characters, the system simply truncates the entry to the 39−character maximum. The created entry is sure to be of valid length if you enable account validation in the future.

If you have instructed the system to ENABLE ACCOUNT−VALIDATION but have not yet created an account validation data base, you receive a warning on the console terminal (CTY) when the system starts operation. The message is:

```
<SYSTEM>ACCOUNTS−TABLE.BIN NOT FOUND − ACCOUNT VALIDATION IS DISABLED
```

The system continues its normal operation; however, no accounts are validated (except for length checking) until you create the necessary account data files and run the account generator program (ACTGEN) to create your account data base.

You should not receive the above warning message if you have created your account data base prior to bringing the system up for operation. Users can log into the system using their valid accounts.

6.1.2 Setting up Account Validation with Existing Files

If you are using account validation on a system that already has files, the accounts for these existing files should be updated before account validation is enabled in the n−CONFIG.CMD file. Notify the users who created these files to change the existing account on every file to their new account(s). This procedure ensures accurate billing immediately after the system is brought up and that daily DUMPER tapes contain files with valid accounts. This means that if you must restore files from a backup tape, the correct account for each file is properly restored; therefore, the disk file storage continues to be accurately charged. (Refer to the TOPS−20 Operator's Guide for the procedure to follow if all files do not get updated.)

6.1.3 Setting up the System for Accounting Shift Changes

The accounting facility also allows you to change your billing rates for system usage at selected times during the day. This action is called an accounting shift change. Accounting shift changes are selected by day-of-week and time-of-day.

You must enter the appropriate commands in the n-CONFIG.CMD file to initiate accounting shift changes. The n-SETSPD program reads these commands each time the system is reloaded. The format of the command placed in the n-CONFIG.CMD file is:

```
CHANGE time days-of-week
```

You can use any format for the time and day, that is, 1500, 15:00, 3:00pm, MONDAY, MON. Or, you can use the keywords ALL, WEEKENDS, and WEEKDAYS. The default for days-of-week is ALL. The following is a typical set of commands that may appear in the n-CONFIG.CMD file:

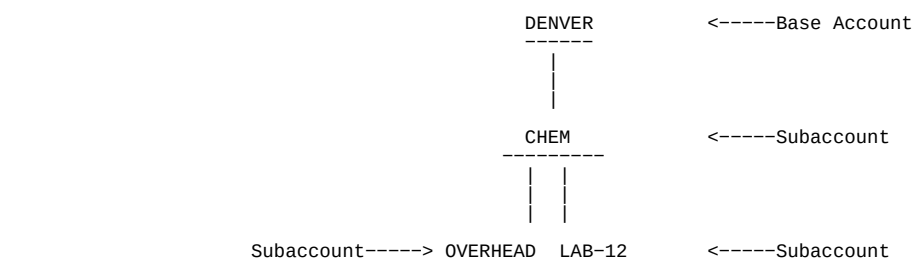
```
CHANGE 9:00 WEEKDAYS
CHANGE 10:00 WEEKENDS,MONDAY
CHANGE 12:00 TUESDAY,THURSDAY,SAT
CHANGE 17:00
```

The CHANGE (ACCOUNTING SHIFT NOW) command to the CHKPNT program provides you with a means of changing shifts during system operation. This command causes an accounting session to end and a new accounting session to begin for all active jobs on the system. Refer to the TOPS-20 Operator's Guide for a description of all the commands that can be given to the CHKPNT program.

6.2 SELECTING AN ACCOUNTING SCHEME

The first thing you must do before you create account data files is set up an accounting scheme. This procedure includes deciding which accounts you wish to create, their expiration dates if you are going to open and close accounts, the names of the users who can use (or charge to) those accounts and, if you are using the class scheduler, the scheduling class associated with each account.

The TOPS-20 account validation facility allows several levels of project administration in a group of accounts having the same base account. For example:



The accounts you would create using the above example are:

```
DENVER
DENVER.CHEM
DENVER.CHEM.OVERHEAD
and
DENVER.CHEM.LAB-12
```

In this example, users at Denver University taking a particular lab course in chemistry (e.g., 12) would log in and charge to their assigned account, DENVER.CHEM.LAB-12.

All accounts that you assign to users can have a maximum length of 39 alphanumeric characters. The system allows you to use a hyphen (-) within the accounts you create (e.g., LAB-12), but no other punctuation (including spaces) can be used. Note that the system uses the period (.) as a delimiter to separate each part of multi-level accounts and the period is counted as one of the 39 characters. Therefore, DENVER.CHEM.OVERHEAD is a user account with 20 characters.

The type of accounting scheme you use depends on the form of project administration you have at your installation. Multi-level accounts are usually created through a form of project administration similar to that used when allowing certain users (perhaps heads of departments) to create subdirectories. (Refer to Chapter 5, Creating Directories.) Remember that subdirectories are just like any other directory. Therefore, users must have accounts to log into their directory. Generally, all files that contain data pertaining to base accounts are created by you or the operator, and all files that contain subaccount data are created by one, or perhaps more than one, project administrator. A project administrator, for example, might be the head of the Chemistry Department. (This could be the same person who handles the subdirectory creation for a group or groups of users.) Allocating the subaccount file creation to a project administrator allows you to collect or budget for one base account (e.g., DENVER.CHEM) and not be directly concerned with the subaccounts. In the example, the head of the Chemistry Department is responsible for creating the subaccount files under DENVER.CHEM., that is, LAB-12 and OVERHEAD. Section 6.3 describes how to create these account data files using a sample accounting scheme.

Figures 6-1 and 6-2 illustrate several ways that you can set up your accounting scheme. Figure 6-1 is a simple scheme that a small organization might use. It also allows you, as system manager, to have complete control over all accounts because you are aware of every account assigned.

Figure 6-1 shows that the manager at Correct Data Company has decided to set up one base account for Correct Data and one base account for each customer using his system. He used the customer name for the accounts. All the people who use the system at Correct Data can charge their computer usage to the Correct Data account, CORRECT-DATA. Unionbank, L & P Food, and Town Square Magazine submitted the names of those people who will be using the system from their respective sites. These are the only people who will be able to log in from their site and charge to their assigned account. The manager at Correct Data also planned expiration dates for each customer account.


```
SAMPLE COMPANY:  Correct Data
TYPE OF BUSINESS:  Timesharing House
PRIMARY MODE OF OPERATION:  Batch

_____________________________________

ACCOUNT  CORRECT-DATA
USER  Hudson, Holland, Gerard, Gionet, King, Kelly, Kohn
 (Note: These 7 people are all the users at Correct Data)

_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _

ACCOUNT  UNIONBANK
USER  Warriner, Bloomstran, Prest, Pendergast
EXPIRES  June 1, 1986

_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _

ACCOUNT  LP-FOOD
USER  Schied, Queeny, Smith
EXPIRES  July 1, 1986

_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _

ACCOUNT  TOWN-SQUARE
USER  Markley, Gerhard, Dole
EXPIRES  July 15, 1986
```

Figure 6-1: Accounting Scheme 1

Using the same sample company, Figure 6-2 shows how a simple accounting scheme of this type can be expanded into a form of project administration. Here, Correct Data and one of its customers, Unionbank, broke down the base accounts into subaccounts.

Because Correct Data bills Unionbank for all its computer usage as one account, the manager at Correct Data is not concerned with how Unionbank subdivides its account, and is probably not aware of the subaccounts at Unionbank. The manager at Unionbank, however, is concerned with the computer usage costs incurred by each department within his company. He supplies Correct Data with the name of the file that contains his subaccount information.

```
SAMPLE COMPANY:  Correct Data
TYPE OF BUSINESS:  Timesharing House
PRIMARY MODE OF OPERATION:  Batch

_____________________________________________

ACCOUNT  CORRECT-DATA
USER  Hudson, Holland

 SUBACCOUNT PAYROLL
 USER  Gerard, Kelly

 SUBACCOUNT OVERHEAD
 USER  Gionet, Kohn, Kelly
 EXPIRES  December 31, 1986

 SUBACCOUNT PROGRAMMING
 USER  King, Carlson

_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _

ACCOUNT  UNIONBANK
USER  Warriner

 SUBACCOUNT TRUST
 USER  Bloomstran

 SUBACCOUNT LOANS
 USER  Prest

 SUBACCOUNT MSTRCHG
 USER  Prest, Pendergast

 SUBACCOUNT PAYROLL
 USER  Bloomstran
```

Figure 6-2: Accounting Scheme 2

Section 6.3 describes how to enter the information for Figures 6-1 and 6-2 into files that are used to create an account data base.

6.3 CREATING AN ACCOUNT DATA BASE

Sections 6.3.1 through 6.3.3 describe how to use your selected accounting scheme and create the necessary files for your data base. Specifically, these sections include how to enter accounting data into files, how to run the account generator program (ACTGEN) to create the data base, and what to do if an error occurs.

6.3.1 Entering Accounting Data into Files

Base and subaccount files are created using a text editor. The format below shows the combination of entries you can make in accounting files using the CREATE command. Each file you or a project administrator creates can contain one or more accounts. Each account can point to one subaccount file, where additional account information is stored pertaining to that account. Following the format is a summary of the valid commands in an account file.

ACCOUNT DATA FILE FORMAT

```
@CREATE (FILE) <directory>filename.type<RET>
INPUT:  filename.type.1

00100   ACCOUNT account/SUBACCOUNT:dev:<dir>filename.type-<RET>
00200   /CLASS:n/ALLOW:n,n<RET>
00300   USER name,name,name,...<RET>
00400   DIRECTORY dev:<directory><RET>
00500   GROUP (ON STRUCTURE) dev:/USER:user group number<RET>
00600   GROUP (ON STRUCTURE) dev:/DIRECTORY:directory group number<RET>
00700   <ESC>
*EU<RET>
```

<filename.type.1>

In addition to the above entries, each entry in the file can have an expiration date in the form:

/EXPIRES:dd-mm-yy hh:mm

This switch indicates when the account will no longer be valid for that entry in the file. For example:

USER name1,name2/EXPIRES:10-JAN-86,name3,name4

In the above example, name2 can no longer use this account after 10 January 1986. Name1, name3, and name4, however, can continue to use the account beyond that date. You could also place the switch immediately following the USER entry. For example:

USER/EXPIRES:10-JAN-86, name1, name2, name3, name4

This format specifies that none of the users in the list can use the account after a certain date. The account, however, remains open, and you can place another list of users in the file who can use the account.

Table 6-1 summarizes the account data file commands. You can type the entire command, or just the characters necessary to distinguish one command from another. For example, ACCOUNT can be typed as AC.

Because the ACCOUNT command has several modifiers, you may have to continue typing the modifiers on the next line. To do this, use a hyphen at the end of the line and continue typing the ACCOUNT modifiers on the next line. For example,

```
0100 ACCOUNT TEST/SUBACCOUNT:SYSA:<MARK> ACCOUNT.TXT-
0200 /CLASS:2/ALLOW:1,3
```

Table 6-1: Summary of Account Data File Commands

Command	Description
ACCOUNT	Specifies the name of the account that you or a project administrator wish to assign. Note: The ACCOUNT command must be the first entry in an account data file because all subsequent entries up to the next ACCOUNT entry are modifiers.
/SUBACCOUNT:	Modifies the ACCOUNT command. It includes the specification of the file where additional data for the account can be found. Note: The ACCOUNT command accepts only one /SUBACCOUNT:modifier. Example: One of your accounting files contains the following commands. ACCOUNT CORRECT-DATA/SUBACCOUNT: <GERHARD>ACCT.TXT ACCOUNT UNIONBANK/SUBACCOUNT: <WARRINER>ACCTG.TXT ACTGEN looks in <GERHARD>ACCT.TXT for more account data for the account CORRECT-DATA, and it looks in <WARRINER>ACCTG.TXT for more data for account UNIONBANK.

Table 6-1: Summary of Account Data File Commands (Cont.)

Command	Description
/CLASS:n	Modifies the ACCOUNT command and is used in conjunction with the class scheduler. It specifies the scheduling class that is valid for this account. For example, ACCOUNT CHEM-207/CLASS:3 means that class 3 is valid when using the account CHEM-207. ACTGEN places this information in the system's accounting data base for use by the class scheduling routines. Use the /CLASS:n switch only if you have selected to specify class scheduling by account. (Refer to Section 10.1 for a complete description of using the class scheduler by account.) When a user gives an account that does not have a valid class associated with it, the system uses the default class, class 0. If the account has a class associated with it, that class is used. The percentage of CPU time that classes can receive is defined in the n-CONFIG.CMD file. To use class scheduling by account, you must have: o made the appropriate entries in the n-CONFIG.CMD file. o updated your ACCOUNTS.CMD file (and subaccount files) to specify the classes that are associated with each account. o run ACTGEN with the INSTALL command to update the ACCOUNTS-TABLE.BIN file. o given the ENABLE CLASS-SCHEDULER command to OPR or brought the system down and back up again to start class scheduling.

Table 6–1: Summary of Account Data File Commands (Cont.)

Command	Description
/ALLOW:n,n	Modifies the ACCOUNT command and is used in conjunction with the class scheduler. It allows you to delegate the assigning of classes to subaccounts by project administrators. It specifies the class or classes that can be used by subaccounts of this account. For example, ACCOUNT CHEM/SUBACCOUNT:<ABC>MORE.TXT– /CLASS:2/ALLOW:1,3 means the CHEM account is in class 2, and that subaccounts created under CHEM can be in either class 1 or class 3. If no /ALLOW switch is given, the administrator is not restricted to using certain classes and, therefore, can give the subaccounts any class. For example, the administrator can give them the same class as the superior directory. The /ALLOW switch is useful when you want the superior account to be in perhaps a higher percentage class than its inferior accounts. Remember that if the administrator does not give a /CLASS switch to the subaccount, users who log into or change to this subaccount will be in class 0.
USER	Specifies one user or the list of users who are allowed to use this account.
*	Specifies that an account is valid for all users on a system. The * is a special argument to the USER command. Example: One instance when you might use * is if you have not established an accounting scheme but would like to allow users to log into the system. You could set up one account and use the * to indicate that all users can use that account. You could also use the * as follows: ACCOUNT MATH–101 USER: MATH.*

Table 6–1: Summary of Account Data File Commands (Cont.)

Command	Description
	This means that all users with a user name beginning with MATH can use the MATH–101 account. For example, the users assigned the user names MATH.SMITH, MATH.JONES, and MATH.BROWN can all use this account.
DIRECTORY	Specifies a directory name. It indicates that the account is valid for anyone with write access to the directory. This feature allows users to create or store files in systemwide or groupwide directories and to charge them to an "overhead" account different from their own account. The usage charged to this directory could be absorbed by system or project administration. This command also prevents users from being charged for file storage in files–only directories. Note: The DIRECTORY command also accepts a form of the wildcard entry. The valid forms are: *:<*>, dev:<*>, or *:<dir>. The asterisk indicates that users with write access to any of the directories matching the wildcard entry may charge their file creation to a certain account. Example: The file that contains account data for account CORRECT–DATA.UNIONBANK.FUND has the entry: DIRECTORY PS:<FORLIB> This entry means that anyone with write access to directory <FORLIB> can use the account CORRECT–DATA.UNIONBANK.FUND when storing files there.

Table 6–1: Summary of Account Data File Commands (Cont.)

Command	Description
GROUP	Specifies that an account is valid for use by certain user and directory groups on a structure. (Refer to Section 5.8 for a description of groups.)
/USER:nnn /DIRECTORY:nnn	Modifies the GROUP command and can be used in any combination. (nnn is a decimal user or directory group number.) /EXPIRES: can be placed after either modifier to indicate when an account becomes invalid for use by the group. Note: Using the GROUP command is helpful if many people are eligible to use an account. Specifying their group number (if they are in a group) eliminates typing a long list of names incorrectly.

6.3.2 Sample Data Files

The examples below show how you could enter the information in Figures 6–1 and 6–2 into account data files. The first example shows the data file for Figure 6–1. Tabs and comment lines beginning with an exclamation point (!) can be used within the file for ease in reading or formatting the file. This file in particular contains all the base accounts and should be stored in your directory. You may find it easier when you run ACTGEN if you name the file ACCOUNTS.CMD. ACCOUNTS.CMD is the default file that the TAKE command under ACTGEN looks for. (Refer to Section 6.3.3 for running ACTGEN and giving the TAKE command.)

```
@CREATE (FILE) <HUDSON>ACCOUNTS.CMD<RET>
Input:  ACCOUNTS.CMD.1


00100 !This file contains definitions of top-level accounts<RET>
00200 ACCOUNT CORRECT-DATA<RET>
00300   USER Hudson,Holland,Gerard,Gionet<RET>
00400   USER King,Kelly,Kohn<RET>
00500 ACCOUNT UNIONBANK/EXPIRES;1-JUN-86<RET>
00600   USER Warriner,Bloomstran,Prest,Pendergast<RET>
00700 ACCOUNT LP-FOOD/EXPIRES:1-JUL-86<RET>
00800   USER Schied,Queeny,Smith<RET>
00900 ACCOUNT TOWN-SQUARE/EXPIRES:15-JUL-86<RET>
01000   USER Markley,Gerhard,Dole<RET>
01100 <ESC>
*EU<RET>

<ACCOUNTS.CMD.1>
```

NOTE

Lines 300 and 400 contain all the users at Correct Data. However, you would not use the asterisk (*) because it would enable all the users of the system, including the users at Unionbank, L & P Food, and Town Square to charge to the CORRECT-DATA account.

Figures 6–3 and 6–4 show what information to enter into base and subaccount files for Figure 6–2. Each block in Figures 6–3 and 6–4 contains information to be entered into a separate file. Some of the blocks contain subaccount (/SUB:) entries that point to other files containing more information about that particular account.

Figure 6–3 shows which entries to make for account CORRECT–DATA and its subaccounts (the top half of Figure 6–2.)

Figure 6–4 shows which entries to make for account UNIONBANK and its subaccounts (the bottom half of Figure 6–2.) Note that all the base account entries for both Correct Data and Unionbank are contained in the file <HUDSON>ACCOUNTS.CMD.

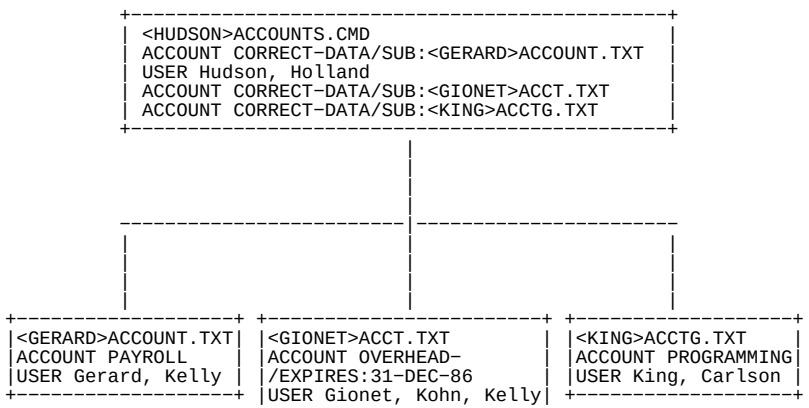


Figure 6–3: Correct–Data Accounting Files

The accounts that will be created are:

CORRECT–DATA.PAYROLL
CORRECT–DATA.OVERHEAD
CORRECT–DATA.PROGRAMMING

Note that users Hudson and Holland can use any account number that begins with CORRECT–DATA, but user Gerard can use only the account CORRECT–DATA.PAYROLL. User Kelly can use the accounts CORRECT–DATA.PAYROLL and CORRECT–DATA.OVERHEAD.

The file type for account data files is optional. Your project administrators can use any file type, for example, .TXT, .CMD, or .ABC.

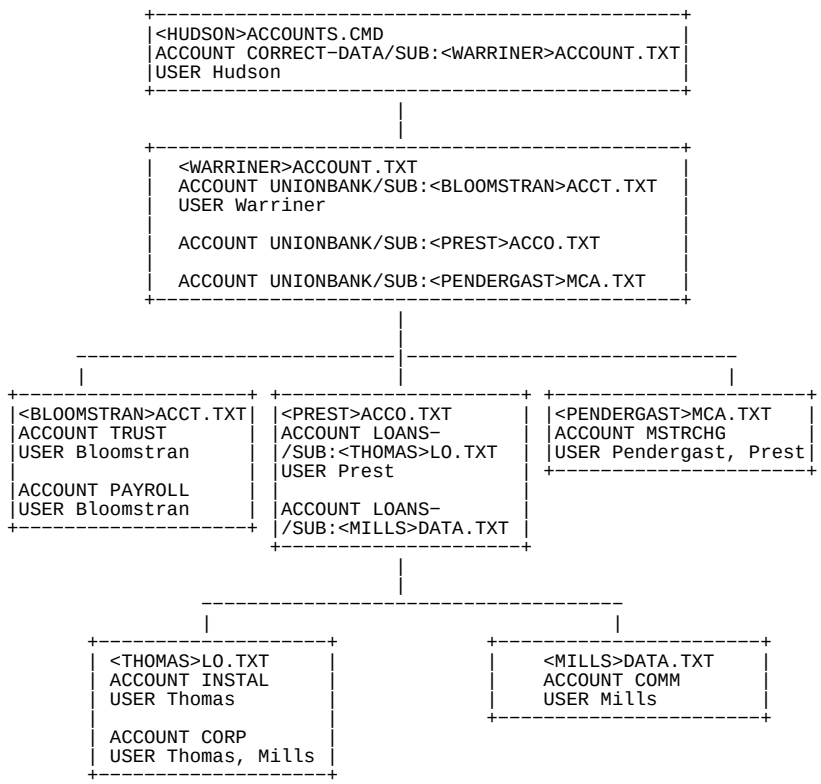


Figure 6–4: Unionbank Accounting Files

The accounts that will be created are:

```
CORRECT-DATA.UNIONBANK.TRUST
CORRECT-DATA.UNIONBANK.PAYROLL
CORRECT-DATA.UNIONBANK.MSTRCHG
CORRECT-DATA.UNIONBANK.LOANS.INSTAL
CORRECT-DATA.UNIONBANK.LOANS.CORP
CORRECT-DATA.UNIONBANK.LOANS.COMM
```

6.3.3 Running the ACTGEN Program

After you create the base account files and the project administrator notifies you that all his subaccount files are complete, you can tell the operator to run the ACTGEN program. ACTGEN takes the accounting information in these files and creates an account validation data base. It is through this data base that the monitor validates all accounts entered by the users of your system. ACTGEN is a privileged program, so you must enable WHEEL or OPERATOR capabilities before giving the ACTGEN command. The command appears as follows:

```
@ENABLE (CAPABILITIES)<RET>
$ACTGEN<RET>
ACTGEN>
```

Valid commands that can be given to ACTGEN are HELP, EXIT, TAKE, CTRL/A, and INSTALL.

The HELP command lists information to assist you when running the ACTGEN program.

The EXIT command terminates the program and returns you to the TOPS-20 command level ($).

The TAKE command accepts as its argument either a file specification or a carriage return. A carriage return defaults to your connected directory and the filename ACCOUNTS.CMD. If you do not name your base account file ACCOUNTS.CMD, the file you specify should be the one that contains your base account information and points to all the existing subaccount files. The TAKE command tells ACTGEN to look in the specified (or default) file for account information. It also tells ACTGEN to look at any subaccount file specifications for additional information pertaining to the account(s) in the base account file. Using Figures 6-1 and 6-2, the manager, Hudson, would specify the TAKE command as follows:

```
@ENABLE (CAPABILITIES)<RET>
$ACTGEN<RET>
ACTGEN> TAKE (COMMANDS FROM FILE)<RET>
```


If the manager in these examples had named his base account file MACCT.TXT, he would specify the TAKE command as follows:

```
ACTGEN> TAKE (COMMANDS FROM FILE) <HUDSON>MACCT.TXT<RET>
```

CAUTION

> If the data files that are pointed to by your base account file are located on structures other than the system structure, be sure the required structures are mounted. Otherwise, the ACTGEN program will fail on those accounts that have subaccount files on unmounted structures.

ACTGEN takes all the information specified in the account files, forms the valid accounts, and creates a new version of the file ACCOUNTS-TABLE.BIN in the directory where ACTGEN is running. Each time the ACTGEN program is run successfully, a new version of this file is created.

While ACTGEN is creating the data base file, it checks for duplicate entries, e.g., two accounts of the same name, and the length of the accounts. If an error occurs, an appropriate message is printed on the terminal where ACTGEN is running, but the program continues to build the data base, using only the accurate data. You should make a note of the error on an error log sheet that you have prepared and later correct the appropriate files using an editor.

ACTGEN also checks expiration dates. If two or more expiration dates are given for the same entry in a file, the system uses the earliest date. For example: if you specify May 15, 1987 as the expiration date for account MATH and your project administrator specifies August 30, 1987 for the account MATH.LAB-201, the system will stop validating all accounts beginning with MATH as of May 15, 1987.

You can press CTRL/A while ACTGEN is running to stop the program and return to ACTGEN command level. The data files are closed and no new version of the data base file ACCOUNTS-TABLE.BIN is created from this session.

The INSTALL command starts account validation. When you enter this command, ACTGEN copies the ACCOUNTS-TABLE.BIN file in your connected directory (or the directory where ACTGEN created the file) to <SYSTEM>ACCOUNTS-TABLE.BIN on the system structure and enables account validation using this new data base.

Because the new version of ACCOUNTS-TABLE.BIN is kept in the directory where ACTGEN was run and not in the directory <SYSTEM>, you have a means of correcting any errors that might occur without disturbing the version currently running in the <SYSTEM>ACCOUNTS-TABLE.BIN file on the system structure. You can give the INSTALL command after you have corrected any problems.

If you do not receive any errors while ACTGEN is creating the data base file (ACTGEN has successfully completed the accounting file and you receive the ACTGEN prompt), you can give the INSTALL command immediately.

You should keep track of which version of the <SYSTEM>ACCOUNTS-TABLE.BIN file you are using. A log book that contains the date that ACTGEN was run and the version number of the data base file is helpful should a system problem occur and you are not sure of which data base file you were using. To find out which version you are using, enable capabilities and give the DIRECTORY command for <SYSTEM>ACCOUNTS-TABLE.BIN on the system structure. The generation number indicates the version that is presently running. The system looks in the current data base file each time it validates a given account.

6.3.4 Data Base Failures/Recovery

If your accounting files were set up inaccurately, or you entered random incorrect data into the data base file, account validation will not work properly. You are aware of this because users cannot log in and/or use accounts that are normally valid for them. Therefore, the account OPERATOR is set up for the user OPERATOR and is always valid. The operator can log into <OPERATOR> with the OPERATOR account, fix the files that are in error, and run ACTGEN to get account validation working again.

6.4 VALIDATING ACCOUNTS

An account is validated when a user gives any one of the following TOPS-20 commands.

- o LOGIN - A user must have a valid account to successfully log into the system
- o SET ACCOUNT (TO) argument
- o Any queue commands, for example, PRINT, SUBMIT, if the account is different from the currently validated account
- o SET FILE ACCOUNT (OF FILES) arguments (TO) argument
- o File creation with an explicit account

The system records the computer time used for valid accounts. This includes CPU time, time used per structure,[1] and peripherals used per job, that is, the number of pages printed on the line printer, tape mounts, tape records read/written, card reader usage, and disk storage. The computer usage incurred by each account is stored in the accounting USAGE.OUT file. This file is used for reports and billing. (Refer to the TOPS-20 Usage File Specification for information about reading and using this file).

How often you run ACTGEN and create a new data base file depends on how frequently you change your accounts. If you expect to have frequent changes (e.g., opening and closing accounts), you may want to establish a standard time each week to run ACTGEN. Your administrators should inform the operator when changes are made to their subaccount files.

NOTE

Once ACTGEN is run and the data base file has been created, you can dump all the account files to magnetic tape and conserve some of your disk space. You must copy the files to disk the next time you need to run the ACTGEN program.

---

[1] To account for the time used on a structure, you must set the structure as REGULATED. Refer to the TOPS-20 Operator's Guide for a description of REGULATED and NONREGULATED structures.

CHAPTER 7

SYSTEM BACKUP PROCEDURES

All the disk packs on your system must be backed up on magnetic tape. This procedure provides both a permanent record of the contents of the disk and a precautionary measure in the event a disk pack and/or its contents are destroyed. On the first day of operation, start a system backup procedure that includes:

1. Saving all the files in all the directories on all structures
2. Saving the directory parameters and critical system programs
3. Saving the front-end file system (one time only)

These procedures should become a part of the operator's scheduled duties.

It is important to start backing up the system immediately after installation. If you follow the backup procedures as they are outlined here and in the TOPS-20 Operator's Guide, you can restore the file system quickly and easily should a mishap occur.

DUMPER

The following sections discuss using the DUMPER program to save files. Make sure when you restore these files with DUMPER that you are running the correct version of DUMPER with your TOPS-20 monitor and that the tape version is compatible with the software. Otherwise, directory passwords could become unusable and you may have to manually respecify them with the BUILD command. Refer to Section 11.2, Password Encryption, for details. In addition, project-programmer numbers, supported in TOPS-20 Version 6, may not be restored at all with incompatible versions of the software.

In a CFS configuration, DUMPER must run on the system to which the tape drives are attached.

To simplify backup and restore operations, you can create DUMPER command files for the operator. Rather than type a list of commands to DUMPER, the operator can then just give the TAKE command with a command file name as an argument. DUMPER will sequentially execute commands contained in the file.

The TOPS-20 User Utilities Guide and the TOPS-20 Operator's Guide discuss the DUMPER program in detail.

7.1 SAVING ALL FILES IN ALL DIRECTORIES

Have the operator run the DUMPER program (with the /FULL-INCREMENTAL switch to the SAVE command) to save all files in all directories. This procedure includes saving all the directories on the system structure and all the directories on any additional structures you have created. You can save all the files (a full dump) or just the files that have changed since the last time the operator ran DUMPER (an incremental dump).

Start a library of the magnetic tapes from the DUMPER operations. Each structure should be copied to a separate tape(s). Each tape should be identified with the system model number or name, for example, 2060, or System-A, the date, the type of save (full or incremental), the name of the structure, and the tape number. (A tape set name may replace the tape number, if labeled tapes are used.) A typical identification may look like:

SYSTEM-A (2060)		SYSTEM-A (2060)
30-JANUARY-1988		30-JANUARY-1988
Incremental	OR:	Full
ADMIN:		ADMIN:
Tape #1 of 2		Tape #3 of 3

In addition to keeping the tapes, keep the listing of their contents. (The operator includes a command to DUMPER to list the contents of the magnetic tapes on the line printer.) These DUMPER log files can be conveniently kept in a binder with the most recent listing on top. Identify each binder with the system model or name, for example, 2065 or System-A. (Chapter 9, System Problems/Crashes describes how to use these log files to restore directories and files.)

Tell users that backup files do exist and post the times when the operator normally creates the backup tapes. Many system managers do not allow users to enter the computer room to mount and use the tapes themselves.

NOTE

The DUMPER program DOES NOT save the files in the console front-end file system. If you lose these files, you must restore them from the floppy disks. (Refer to Section 7.6)

7.1.1 Full Dumps

Full dumps contain all the information on the system, with the exception of the console front-end files, and can be used to restore many of the files that were on the system to their previous state. Therefore, full dumps contain a copy of every file in every directory on every structure.

NOTE

Full dumps are known as FULL-INCREMENTAL dumps. This name corresponds to the /FULL-INCREMENTAL switch that you give with the SAVE command to DUMPER.

7.1.2 Incremental Dumps

Incremental dumps (using the /INCREMENTAL switch with the SAVE command) cause DUMPER to save the files that have never been saved (new files) and the files that have been updated since the last time an incremental DUMPER operation was performed. Many managers request an incremental dump Monday through Thursday and a full dump on Friday.

The File Descriptor Block (FDB) of each file contains the information necessary to determine if the file has been updated since it was last saved during an incremental DUMPER operation. A file that has changed since the last time it was saved is automatically saved again on tape; otherwise, it is passed over.

The operator should give the INCREMENTAL switch to DUMPER and specify each structure one at a time. By running DUMPER for each structure individually, the operator can copy each structure onto a separate tape. After running DUMPER and copying the structures that are presently on-line, the operator should mount any additional structures that have been used that day and run DUMPER for them also.

An incremental dump is faster than a full dump and requires less magnetic tape. By specifying a value with the /INCREMENTAL switch, you can cause modified files to be written to more than one incremental backup tape. This is helpful if you want to be certain you can recover the files, even if one of the tapes has data errors.

7.1.3 Security of Backup Tapes

It is a good idea to protect the security of your backup tapes. If you allow a non-privileged user to mount them, he or she may gain access to confidential information, such as other user's data files. If the unencrypted passwords were saved along with other directory parameters, a technically sophisticated user can retrieve them from the DUMPER backup tapes, and thus obtain unauthorized access to the privileged accounts on the system.

7.2 A COMMON BACKUP POLICY

A common backup policy is outlined below. You can set up your own backup policy.

1. Each day, take an incremental save of the files that have changed from the previous day's backup tape. Keep the incremental saves until a full save is made, at which time you can recycle the incremental tapes.

2. At the end of the week, take a full save of all the files on the system. Keep the full saves for six months, at which time you can recycle the tapes into the backup system.

3. Every six months, take a full save and keep it for a number of years, or if you choose, indefinitely.

7.3 MAGNETIC TAPE REQUIREMENTS

You need a supply of magnetic tapes to start a system backup procedure. This section provides a guideline for the number of tapes you should have on hand for your installation. It is assumed that there are 2400 feet per reel of tape.

Type of Disk Drive	Reels Per Drive	Bits Per Inch
RP06	7	1600
RP06	2	6250
RA60	9	1600
RA60	3	6250
RP20 (one spindle)	19	1600
RP20 (both spindles)	37	1600
RP20 (one spindle)	6	6250
RP20 (both spindles)	12	6250
RP07	7	6250
RP07	19	1600
RA81	6	6250
RA81	19	1600

Therefore, the number of tapes you stock depends on the type of disks at your installation. It also depends on the backup procedure you use. For example, if you save your daily incremental tape dumps for a longer time than usual, it takes longer to recycle these tapes into the backup system, and thus you need more tapes.

Generally, during the first month after installation, you may need approximately 36 (2400 ft.) tapes for each RP06 or RA60, 45 tapes (6250 bit/in) or 180 tapes (1600 bit/in) for each RP20 disk (2 spindles), and 72 tapes for each RP07 or RA81 (1600 bit/in).

NOTE

These estimates assume a magnetic tape blocking factor of 1. You can specify a higher blocking factor and save much space on your tapes. Before doing this, however, there are cautions that you must consider. The description of DUMPER in the TOPS–20 User Utilities Guide explains how and when you can increase blocking factors.

7.4 MAKING A SYSTEM CRASH TAPE

As the name implies, the system crash tape is used to re–create the system structure should it become unusable. This tape is created in addition to the tapes that contain FULL–INCREMENTAL and INCREMENTAL saves of all files and directories. You should make a new system crash tape whenever you add a new user, change any directory parameters, or make a change (patch) to:

- The monitor you are running
- The TOPS–20 Command Processor
- The DLUSER program
- The DUMPER program

Therefore, the crash tape contains only the files necessary to recover user directory parameters and important system programs. User files themselves are restored from the FULL–INCREMENTAL and INCREMENTAL saves of the public structure.

Label this tape SYSTEM BACKUP TAPE and include the system structure name; DECSYSTEM–20 model number or name, for example, 2060 or System–B; and the date and time the tape was created. You should follow this procedure once a day if users are allowed to change their own directory parameters. (Refer to Section 5.7.2 for information about allowing users to change directory parameters.) If you are not using password encryption (refer to Section 11.2), be careful to protect the backup tapes against reading by unauthorized users, because all the passwords for your users will be accessible.

If you are also using mountable structures, you should periodically run DLUSER to copy their directory parameters to a file on another structure, preferably the system structure. Then, if the mountable structure is destroyed, you will be able to recreate the directories.

NOTE

Do not use a labeled tape when creating a System Crash Tape. The reason for this is that the installation software that is used to create the system structure cannot read tape labels.

The order of files on the crash tape is:

1. <SYSTEM>MONITR.EXE
2. <SYSTEM>EXEC.EXE
3. <SYSTEM>DLUSER.EXE
4. DLUSER data files
5. <SUBSYS>DUMPER.EXE
6. DUMPER save sets containing the directories:

   <SYSTEM>
   <SUBSYS>
   <NEW-SYSTEM>
   <NEW-SUBSYS>
   <UETP>
   <GALAXY-SUBSYS>

Notice that the format of this tape is the same as the TOPS-20 Installation Tape that you used to install the TOPS-20 software.

7.5 MAKING A CRASH TAPE USING BATCH

You can create a batch job to make your crash tape or type the commands at the operator's terminal. Example 1 shows the standard control file that you can submit as a batch job to create this tape. In the example, PUB: is the name of the system structure.

EXAMPLE 1

SYSTAP Control File

```
@TYPE(FILE) SYS:SYSTAP.CTL<RET>

! Obtain a tape drive
@MOUNT TAPE TAPE:/WRITE/LABEL:UNLABELED

!Systems not using Tape Drive Allocation must replace the
!MOUNT TAPE command with @ASSIGN MTA0: and @DEFINE TAPE:
!(AS) MTA0: commands.

@ENABLE (CAPABILITIES)
@REWIND (DEVICE) TAPE:

!Save the monitor
@GET (PROGRAM) PUB:<SYSTEM>MONITR.EXE
@SAVE (ON FILE) TAPE:  !Save the TOPS-20 Command Language Interpreter
@GET (PROGRAM) SYSTEM:EXEC.EXE
@SAVE (ON FILE) TAPE:

!Save the DLUSER program
@GET (PROGRAM) SYS:DLUSER.EXE
@SAVE (ON FILE) TAPE:

!Run the same DLUSER program, saving the directory structure
!on tape
@START
*DUMP (TO FILE) TAPE:
*EXIT

!Save DUMPER
@GET (PROGRAM) SYS:DUMPER.EXE
@SAVE (ON FILE) TAPE:  !Run the same DUMPER, saving SYSTEM: and SYS:
@START
*TAPE (DEVICE) TAPE:
*LIST (LOG INFORMATION ON FILE) SYSTAP.LPT
*SSNAME SYSTEM-FILES
*SAVE (DISK FILES) PUB:<NEW-SYSTEM>,PUB:<SYSTEM>
*SSNAME SUBSYS-FILES
*SAVE (DISK FILES) PUB:<NEW-SUBSYS>,PUB:<SUBSYS>
*EXIT
```

```
!Print the DUMPER log file
@PRINT SYSTAP.LPT/NOTE:BACKUP TAPE

@DISMOUNT TAPE:
@

!Systems not using Tape Drive Allocation must replace the
!DISMOUNT TAPE: command with @UNLOAD (DEVICE) TAPE:  and
!@DEASSIGN TAPE: commands.
```

To run SYSTAP, submit the batch control file using the TOPS-20 SUBMIT command. In the event the system structure becomes unusable, the crash tape can now be used by following the instructions in the TOPS-20 KL Model B Installation Guide.

HINT:

Before you store the crash tape, verify that you have made a usable tape. That is, mount the new crash tape, follow the instructions in the TOPS-20 KL Model B Installation Guide to load the monitor, and use DUMPER to get a listing of the tape's contents.

7.6 SAVING THE CONSOLE FRONT-END FILE SYSTEM

The DUMPER program does not save the contents of the console front-end file system. You can, however, make a backup copy of the file system by copying your floppy disks using the front-end program COP (for copy). You should make at least one backup copy of your console front-end file system (refer to Section 3.4).

To run the COP program, follow the steps outlined below. You need not stop timesharing when following these steps.

1. At the operator's console, type CTRL/backslash; the system prints PAR>.

   ```
   CTRL/backslash

   PAR>
   ```

2. Type MCR COP and press the RETURN key; the system prints the COP prompt.

   ```
   PAR>MCR COP<RET>
   COP>
   ```

3. Place the floppy disk to be copied in drive 0 and the floppy to contain the new files in drive 1. Be sure to mount the floppy disks correctly; this includes checking that the paper containing the floppy directory is not accidentally attached to the back of the floppy disk.

4. Type DX1:=DX0: and press the RETURN key; the system starts the copying, which takes a few minutes. After the copying is complete, the system verifies the copy and prints a message. Type CTRL/Z to exit COP.

   ```
   COP>DX1:=DX0:<RET>
   COP - STARTING VERIFY

   COP>^Z
   ```

5. To return to TOPS-20 Command Level, type a CTRL/backslash; the system prints PAR>; type another CTRL/Z, or type QUIT and press the RETURN key.

   ```
   CTRL/backslash

   PAR>^Z

   @
   ```

CHAPTER 8

TAPE STORAGE

Chapters 1 through 6 deal primarily with setting up and using your disk resources. Chapter 7, System Backup, describes how to save all the data from your disk structures onto magnetic tape. These tapes are the backup tapes that you use to restore directories and perhaps entire disk structures if something happens to the disks (refer to Chapter 9). In addition to using magnetic tapes for system backup tapes, you can use magnetic tapes to store other types of data and save valuable disk space.

This chapter describes two other uses for magnetic tapes, File Archiving and File Migration. It also describes how you can allow the system and the operator to control all tape drive assignments, Tape Drive Allocation, and how you can set up some or all of your tapes to contain labels, Tape Labeling. These uses are described in the following order.

- FILE ARCHIVING
- FILE MIGRATION
- TAPE DRIVE ALLOCATION
- TAPE LABELING

In addition, the last section of the chapter discusses how to set up two DECSYSTEM-20s to share a TX02 tape subsystem.

File archiving provides you and users of the system with a voluntary way to move important files from the disk to magnetic tape for long-term storage. These tapes are stored separately from your system backup tapes. Users can access these tape files as easily as they access files on the disk. When users want to restore archived files to disk, they give a command to the TOPS-20 command processor. The system then tells the operator which tapes to mount and proceeds to restore the files. Section 8.1 describes why you would use the file archiving facility, and how to set up your system to archive files to magnetic tape.

File migration provides you with a means of controlling the use of disk space. File migration is especially useful if your disk space is very low on a particular structure, for example, the system structure. This type of disk space control is, for the most part, involuntary on the part of the user. Old unused disk files are periodically moved (migrated) to magnetic tape by the system operator. Again, you should store these tapes separately from your system backup tapes. Users still maintain easy access to these files and retrieve them the same way as they retrieve archived files. Section 8.2 describes why and when you would use the file migration facility and how to set up your system to migrate files to magnetic tape.

If you use the file archiving or file migration facilities, or both, remember that these tapes are in addition to your system backup tapes. They are not replacements. You must continue to run the DUMPER program and create full and incremental system backup tapes.

Tape drive allocation provides the system and computer operator with complete control over tape drive usage. This means that it prevents users from issuing the ASSIGN command and reserving tape drives for their jobs. When users issue the MOUNT command, the TOPS-20 Tape Drive Allocation system and the operator control the allocation of tape drives. You must use the tape drive allocation facility if you use tape labeling; however, using tape drive allocation does not restrict you to using labeled tapes.

Tape labeling provides a means of storing label information on the tape itself that identifies the tape and describes the data on the tape. This label information is in an industry standard format so you can read and write tapes to be used with different computers. Tape labels can also add more security and reliability to your tape system. Section 8.4 describes why you would use tape labels, and also how to set up your system to begin labeling tapes.

There are no dependencies among file archiving, file migration, and tape labeling. For example, you can use the file archiving facility without using file migration or tape labeling. Each tape facility can be used separately. There is, however, the dependency that tape drive allocation must be turned on to use tape labels.

8.1 FILE ARCHIVING

File archiving provides a means of storing data on magnetic tape and freeing valuable disk space. This type of off-line storage allows users to store (archive) important files on tape, keeping their disk space below their permanent allocation, and still have easy access to those files.

If your installation has more than one computer system, the archive tapes can be common to all systems. You can put files on tape from

one system, move a directory and its files to another system, and still retrieve files from the tape in the ordinary manner.

Unlike general system backup tapes, the tapes that contain archived files are usually kept for a much longer time, for example, seven to ten years.

8.1.1 Setting Up the System to Use File Archiving

When you receive the TOPS-20 Installation Tape and have brought up the TOPS-20 monitor, your system contains a built-in default of 3650 days for recycling archived tapes. To change the 3650 day (10 year) default, you can enter a command in the n-CONFIG.CMD file. The command you use is:

ARCHIVE-TAPE-RECYCLE-PERIOD days

Select a length of time that is appropriate for your installation. Place the ARCHIVE-TAPE-RECYCLE-PERIOD command in the n-CONFIG.CMD file during software installation, or edit the file at a later date when you are planning to reload the system.

Each time the DUMPER program copies an archived file to tape, it places the expiration date argument in the FDB of the file. The MAIL program notifies users when a file on tape has reached its expiration date. If the file is no longer needed, the user can discard (using the DISCARD command) the information in the file's FDB that points to the file on tape. After all the files on a tape have passed their expiration dates and no users have FDBs in their directories that point to that tape, the tape can be recycled. Refer to Section 8.2.5 for additional information on how to recycle tapes.

8.1.2 What Happens When Users Archive Files

Users archive files voluntarily by giving the ARCHIVE command. After a specific generation of a file has been archived (e.g., MYFILE.CBL.6), it cannot change. Users can obtain copies of archived files by using the RETRIEVE command, but cannot alter those files. The TOPS-20 User's Guide describes the ARCHIVE and RETRIEVE commands.

When you establish your installation's policy for file archiving and notify users of its availability, you may want to instruct users to archive source files only. For example, files with a file type

.CBL, .MAC, .TXT, .RNO, or .FOR

should be archived; but, files with the file type

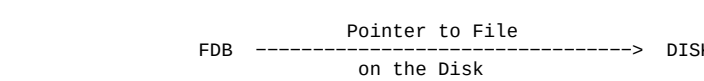
.REL, .EXE, or .MEM

should not be. This restriction saves space on your magnetic tapes.

To completely archive a file, two copies must exist in the archives. This means that each archived file is stored on two tapes. Having the archived file on two tapes provides you with a backup tape if later you cannot retrieve a file off one of the tapes. The DUMPER program, which is used to archive files, records both tape identifying numbers in the FDBs of the files being archived.

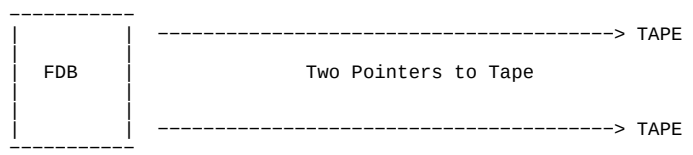
The diagrams below illustrate what happens when a user archives a file.

First, the user creates a file. The File Descriptor Block (FDB), among other file information, contains a pointer to the location of the file on the disk.



The user gives the ARCHIVE command for this file. The first or next time the operator runs DUMPER with the ARCHIVE command, the system locates all files that have been marked for archival by the ARCHIVE command and copies these files to tape. The FDB now contains pointers to the file on the disk and to the file on the tape.

The second or subsequent time the operator runs DUMPER with the ARCHIVE command (remember that archived files are contained on two tapes), DUMPER copies the file again to the second tape. DUMPER then deletes the pointer to the location of the file on the disk. DUMPER places another pointer in the file's FDB to the second tape that contains the file and deletes the contents of the file on disk. After a user archives a file, the file name no longer appears in the user's directory list. The user must give the DIRECTORY command with the ARCHIVE or INVISIBLE subcommand to see the file name.



8.1.3 What Happens When Users Retrieve Files

Users request files be returned to disk (retrieved) by using the RETRIEVE command. When the operator runs DUMPER to process retrieval requests, DUMPER notifies the operator of the second tape that contains the file. If the file cannot be copied from the tape (e.g., the tape is bad), DUMPER notifies the operator of the first tape that contains the file. When DUMPER returns a file to disk, the FDB of that file now contains two pointers to tape and one to the disk. The pointer to the file on the disk remains in the FDB until the file is deleted from disk.

8.1.4 When to Create Archive Tapes

You can select how often the operator runs DUMPER to archive files. However, running DUMPER (with the ARCHIVE command) every day before or after your general system backup procedure is probably closest to your present schedule.

The steps below provide an example of a typical procedure. These steps assume that this is the first time you are archiving files. Although you do not have to use this procedure, it is one that best utilizes your tape resources. (The TOPS-20 Operator's Guide describes the procedure for running DUMPER with the ARCHIVE command.)

Step	Procedure

1. The operator runs DUMPER and performs the normal (incremental or full) backup procedures for the entire system. (Refer to Chapter 7, System Backup Procedures.)

2. The operator mounts a brand new tape (a tape that has been initialized if you are using tape labels; refer to Section 8.4.3) to contain the archived files, for example, TAPE 1.

3. The operator runs DUMPER with the ARCHIVE command. DUMPER locates all files marked for archival and copies them to tape.

4. The next evening (or the next time system backup is performed), the operator mounts a brand new tape, e.g., TAPE 2. He does NOT mount the tape used the first time.

5. The operator runs DUMPER with the ARCHIVE command. This time DUMPER finishes the archival run of the previous night by making a second copy of those files. In addition, DUMPER locates all the files newly marked for archival and copies them to tape for the first time.

6. The operator repeats this process every day until the tapes are full.

7. For example, the third night, the operator mounts the first tape, TAPE 1.

8. The operator runs DUMPER. DUMPER finishes the archival run of the previous night by making a second copy of the previous night's files. It also locates all the files newly marked for archival and copies them to tape.

9. The fourth night, the operator mounts the second tape, TAPE 2. Again, DUMPER finishes the archival run of the previous night by making a second copy of those files. It also locates all the files newly marked for archival and copies them to tape.

NOTE

DUMPER checks tapes for duplicate files. It does not write both copies of the same file on the same tape. If the wrong tape is mounted, DUMPER outputs an error message.

8.1.5 Processing Retrieval Requests

When a user gives the RETRIEVE command to request an archived file, the request is stored in a queue. You must establish a policy for how often the operator should process the retrieval requests contained in the queue. (The TOPS-20 Operator's Guide describes how to process retrieval requests.) If you have encouraged users to archive their files, you should instruct the operator to process the request queue frequently.

8.2 FILE MIGRATION

Some installations must control the use of disk space by periodically migrating files to magnetic tape. This forced file migration allows the management of an installation to move old unused disk files to a less expensive storage medium. Similar to archived files, migrated files are still easily accessible to the user. File migration also allows you to keep users' directories below their permanent allocation. (Refer to Section 5.5 for a description of permanent and working storage allocations.)

Whether you use the file migration facility depends almost entirely on your disk space resources. If users are archiving files regularly, if their directories are usually below their allowed permanent disk allocation, and your system is not continuously interrupted with "disk space low" messages, you may choose not to migrate files. Otherwise, if you are constantly receiving the [CAUTION - DISK SPACE LOW ON structure name] message, you may want to forcibly migrate files from the disk.

Sections 8.2.1 through 8.2.3 describe using file migration. They include:

- the program you must run before migrating files to tape
- the command that can be placed in the n-CONFIG.CMD file to change the default tape recycle period
- when to run the REAPER program that marks files for migration and marks for deletion the contents of archived and/or migrated files
- a sample of the REAPER.CMD file that you can use as a default file to be read by the REAPER program
- when to run the DUMPER program that locates files marked for migration and copies them to tape

- when to process retrieval requests for migrated files
- when to recycle migrated (and archived) tapes

8.2.1 Setting Up the System to Use File Migration

When you receive the TOPS-20 Installation Tape and have brought up the new TOPS-20 monitor, your system contains a built-in default of 180 days for recycling migrated tapes. This default is placed in the FDB of each file as it is migrated to tape.

To change the 180-day default, you can enter a command in the n-CONFIG.CMD file to inform the system when you plan to recycle your migrated tapes. This command is:

TAPE-RECYCLE-PERIOD days

Select a length of time that is appropriate for your installation. The default of 180 days, however, is a standard time period. You can place the TAPE-RECYCLE-PERIOD command in the n-CONFIG.CMD file at the time you install the system (refer to the TOPS-20 KL Model B Installation Guide). Or, you can edit the n-CONFIG.CMD file at a later date. Remember that if you edit the file at a later date, you must reload the system to process the commands in the n-CONFIG.CMD file.

CAUTION

If you decide to change the 180 day default, place the TAPE-RECYCLE-PERIOD command with the new argument in the n-CONFIG.CMD file and reload the system. Otherwise, the default recycling period does not change until the next system reload.

8.2.2 Using the REAPER Program

The REAPER program is the tool used to free disk space. It performs the following functions:

- Marks for migration the files that have not been referenced for a specified period of time
- Marks for deletion the disk contents of archived or migrated files, either after they have been successfully copied to tape, or after they have been returned to disk with the RETRIEVE command, and have not been referenced for a specified period of time

- Trims directories that are over permanent disk allocation by marking files in those directories for migration

- Deletes (purges) the tape pointers in FDBs on the disk that have reached their tape storage expiration date. That is, the file's FDB will no longer contain a pointer to the contents of the file on tape.

You can instruct the operator to run the REAPER program and perform one, several, or all of these functions. The operator can give a list of commands to REAPER or use the TAKE command with the default argument SYSTEM:REAPER.CMD. After the system is installed, the directory <SYSTEM> contains a default REAPER.CMD file. You can use this file as it is or use an editor and change the default parameters. The default SYSTEM:REAPER.CMD file appears similar to the following.

```
$TYPE (FILENAME) SYSTEM:REAPER.CMD<RET>

!Sample REAPER policy file

!Directories not to be considered (specify the structure and
!directory)

SKIP PUB:<NEW-SUBSYS>,PUB:<NEW-SYSTEM>,PUB:<SYSTEM>,PUB:<SUBSYS>

PERIOD 60                 !Specifies the age limit on
                          !files

MIGRATE                   !Migrate files older than PERIOD days

DELETE-CONTENTS           !Delete the contents of
                          !unreferenced files older than
                          !PERIOD with tape backup

TRIM                      !Trim directories over perm allocation
                          !back to perm allocation

ORDER *.TMP,*.LST,*.REL   !TAKE files in this order
                          !during TRIM
```

Note that the SKIP command includes a list of directories that are not to be touched by the REAPER program. You can add other system or user directories to this list. The list can contain approximately 75 directories. Be sure that the operator always includes this command when running the REAPER program; otherwise, you may accidentally migrate some very important files from the disk. You can use more than one SKIP command to specify additional directories to be skipped, rather than list them all in one command. That way, if there is an error in processing one command, it will not affect the processing of the other commands. This is especially useful when SKIP commands are included in a file.

The REAPER program accepts the following commands.

```
BEGIN (Processing files)
DELETE-CONTENTS (Of old offline files)
EXIT (To monitor)
LIST (Output to file)
MIGRATE (Old files to offline storage)
ORDER (For trimming)
PERIOD (For migration)
POLICY (does a TAKE on SYSTEM:REAPER.CMD)

PURGE (Expired FDBs from disk)
SCAN (Only)
SKIP (Directories)
TAKE (Commands from file)
TAPE (Check of tapes in use)
TRIM (Directories over allocation)
```

The TOPS-20 Operator's Guide provides a complete description of all the commands that can be given to the REAPER program or placed in the REAPER.CMD file. Typically, you give a number of commands to REAPER, one for each operation you want performed.

The availability of disk space determines how often you run the REAPER program. If your disk space is low, you may want to run the REAPER program daily to free up as much disk space as possible. Other installations may run it once a month or less.

8.2.3 Using the DUMPER Program

After the REAPER program marks files for migration, the operator runs the DUMPER program to copy the files to tape. Similar to an archived file, a migrated file is not completely migrated until two copies of the file exist on magnetic tape. Section 8.1.4 describes a procedure for copying archived files to tape. You can use this same procedure for migrated files, by using the MIGRATE command instead of ARCHIVE.

If you use both the file archiving facility and the file migration facility, do not merge archived and migrated files on the same tapes. The expiration dates for migrated files differ greatly from the expiration dates on archived files. If you put them on the same tape, you will end up saving migrated files for approximately ten years and use up all your tape resources very quickly.

8.2.4 Processing Retrieval Requests for Migrated Files

When a user gives a DIRECTORY command, the files that have been migrated to tape still appear in the directory list; however, each file has a notation (;OFFLINE) beside the filename to indicate that the file is contained on tape and not on the disk. The versions of migrated files that have been copied to tape can be returned to disk, and unlike archived files, they can be altered and/or renamed in the ordinary manner. The user requests that a migrated file be returned to disk with the RETRIEVE command. These requests are stored in the same queue as archive requests until the operator processes the queue. The TOPS-20 Operator's Guide describes how to process retrieval requests. Retrieval requests for migrated or archived files should be processed frequently.

8.2.5 Recycling Migration (and Archive) Tapes

When all the migrated or archived files on a tape have passed their expiration dates and all pointers to these files on the disk have been deleted, you can recycle the tape.

The PURGE command to REAPER checks tape expiration dates and notifies users by the MAIL program when migrated or archived files on tape are about to expire. Users can retrieve the file to disk again or discard the tape pointer on disk if they no longer need the file.

The operator can determine if a tape can be recycled by giving the TAPE command to REAPER. If a tape is not mentioned in the TAPE output list, this means that none of the disk structures that are on-line at this time and specified to REAPER contain FDB pointers to that tape. However, be sure that you check all possible places for on-line (disk) pointers to this tape. That is, run REAPER with the TAPE command on all the disk structures on all systems that may contain pointers to this tape. If files have passed their expiration date and pointers to them still exist on the disk, the operator can run REAPER with the PURGE command to delete these pointers. The operator should be certain that files are no longer needed before using the PURGE command.

HINT

When a migration tape is full, have the operator use the PRINT command to DUMPER to obtain a hard-copy listing of the tape contents.

8.3 TAPE DRIVE ALLOCATION

Tape drive allocation provides the system, and not the user, with complete control over tape drive usage. When accessing a magnetic tape, the user must give a MOUNT command to request that the operator mount the tape on a drive. Once the operator responds to the user's request, the user can access the tape. When the user is finished with the tape, the user gives the DISMOUNT command to release the tape drive back to the system. From the user's point of view, the MOUNT and DISMOUNT commands replace the ASSIGN and DEASSIGN commands. The operator selects the drive for the user, and the system informs the user how to access the tape. Using tape labeling requires that you use tape drive allocation; however, this does not restrict you to the use of labeled tapes only.

8.3.1 When to Use Tape Drive Allocation

Table 8-1 lists the differences between using and not using tape drive allocation.

Table 8-1: Tape Drive Allocation

Tape Drive Allocation	No Tape Drive Allocation
You must make an entry in the n-CONFIG.CMD file to use tape drive allocation.	No entry required in the n-CONFIG.CMD file.
Users can use labeled and unlabeled tapes.	No support for labeled tapes. This means that all tapes, whether they contain labels or not, are treated as unlabeled.
Users cannot give the ASSIGN and DEASSIGN commands for allocated tape drives, but must give the MOUNT and DISMOUNT commands.	Users can give the ASSIGN and DEASSIGN commands for allocated tape drives and cannot use the MOUNT and DISMOUNT commands.
The operator should not use the UNLOAD button on tape drives, but should use the DISMOUNT command to OPR.	The operator may unload tapes using the UNLOAD button on the tape drive.

8.3.2 How to Enable/Disable Tape Drive Allocation

To use tape drive allocation enter the command

ENABLE TAPE-DRIVE ALLOCATION

in the n-CONFIG.CMD file.

You can disable tape drive allocation on a tape drive by using the SET TAPE-DRIVE MTAn: UNAVAILABLE command.

8.3.3 Tape Mounting Policy

Occasionally, you may mount a tape that the system cannot read. For example, the operator mounts a tape that has a density of 800 bits per inch (bits/in) on a drive that does not support this density. The system checks tapes for labels; even if this tape contains labels, the incorrect density prevents the system from recognizing them.

With such errors, the system can be set up to immediately unload the tape and protect it from being accidentally erased, or it can treat the tape as unlabeled and continue processing.

If you do not want the system to classify these tapes as unlabeled, you can place the TAPE-RECOGNITION-ERRORS command in the n-CONFIG.CMD file with the appropriate argument. The format of this command follows:

```
                              REGARD-AS-UNLABELED
TAPE-RECOGNITION-ERRORS
                              UNLOAD
```

The system uses REGARD-AS-UNLABELED if no entry is made in the n-CONFIG.CMD file. If REGARD-AS-UNLABELED is in effect, you should instruct operators to be especially careful when mounting tapes with write rings. The tape's contents cannot be overwritten if the write ring is not inserted.

8.4 TAPE LABELING

This section describes what tape labels are and how, as system manager, you can initiate using them.

Magnetic tape labels are records that are interspersed with user-defined data on a tape. They are informational records that describe the user data in a standard fashion that is recognized by many computer systems.

The TOPS-20 tape labeling system allows you to read and write tape labels that conform to ANSI (American National Standards Institute) and DEC standards. The tape labeling system also allows you to read tapes that are labeled according to IBM labeling standards.

Tape labeling is an option. You can start or continue to run your system using unlabeled tapes. If you have hundreds of unlabeled tapes at your installation, you may decide not to convert entirely to a labeled shop. Instead, you may have a combination of labeled and unlabeled tapes.

Sections 8.4.1 through 8.4.3 describe the advantages of using tape labels and how to set up your system to begin labeling tapes. The TOPS-20 Tape Processing Manual provides a complete description of the ANSI, DEC, and IBM standard label formats and how to use them. It also describes the interface between the operator and magnetic tapes and the user and magnetic tapes.

8.4.1 Why Tape Labels?

An unlabeled tape has a gummed label on the outside of the reel that identifies the contents of the tape. When the operator selects, mounts, and types the identity of an unlabeled tape, the system assumes that the mounted tape is the one that the user (or job) requested. No checking is performed by the system.

A labeled tape, however, contains standardized information on the tape itself that identifies and describes the data on the tape. This internal label information is in addition to the gummed label on the outside of the reel. With labeled tapes, the operator selects a tape (by looking at the outside gummed label), mounts the tape on any available drive, but does not type in any identifying information at the terminal. When a user issues a MOUNT request for a labeled tape that is already mounted, the system automatically locates the drive containing the tape requested, and checks to ensure that the correct tape has been mounted. This facility for automatically locating and checking tapes is described further below.

A labeled tape consists of a volume label group, followed by one or more files. (A volume is a reel of magnetic tape.) The volume label group is a set of one or more records at the beginning of the tape. It contains a volume identifier, commonly referred to as a VOLID, and other identifying information. (See Figure 8-1.) You, as system manager, select the VOLIDs to be used at your installation. The VOLID is a name containing from one to six alphanumeric characters. A user requests access to a specific volume by specifying its VOLID to the system.

Each file on a labeled tape contains a file header label group, the file data (written by the user program), and a file trailer label group. (See Figure 8–1). Optionally, the file can contain user labels, whose contents are specified and examined only by user programs. Volume labels, header labels, trailer labels, and user labels are described in the TOPS–20 Tape Processing Manual.

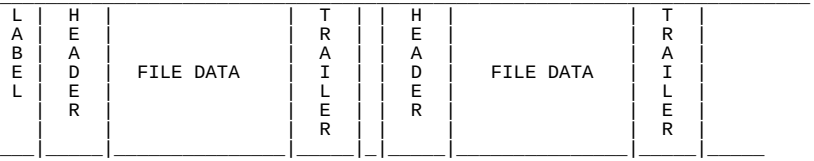


Figure 8–1: Organization of Labeled Tapes

Because every labeled tape contains a unique identifier, or VOLID, the system can read this VOLID and ensure that the correct tape has been mounted. This automatic checking improves the reliability of your tape system. It significantly reduces the likelihood of an operator mounting the wrong tape.
Also, some or all of your tape drives can be set to automatically recognize tape volumes as they are mounted. This process is called automatic volume recognition (AVR). Setting AVR means that after the operator mounts a tape, the system automatically reads the first record and inspects it for label information. If the tape contains no labels, the system classifies it as unlabeled and the operator must key in a volume identifier for the tape. If a request for the tape is pending, the system readies the tape for use by the requesting job. If a request for the tape is not pending, the system stores the VOLID in a table and waits for a request. Therefore, automatic volume recognition provides the following benefits.

- The operator does not have to type tape–identifying information to the system when mounting a labeled tape.
- It provides a faster connection between a user's job and the tape requested.
- The operator can mount a tape long before it is needed. When a job requests the tape, the system locates the drive that contains the requested tape and readies it for use.

Tape labels also improve the security of your tape system. DEC–standard labels identify the owner, as well as the volume, in the volume label. The file labels specify protection codes for the individual files. These labels protect a tape from being inadvertently written on and valuable data destroyed by a user who does not have the appropriate access rights to the tape or its files.

In addition to the added reliability, security, and volume recognition, labeled tapes provide you with a means of interchanging tapes between DECSYSTEM–20s and other computers. This interchange capability extends the mobility of data between different systems. You can write ANSI– and DEC–standard labeled tapes and mount these tapes on other systems using ANSI– or DEC–standard labels, and vice versa. You can mount a labeled tape that was written in EBCDIC with labels conforming to IBM's OS standards, and read it on a DECSYSTEM–20 as if it were ANSI–standard labeled.

Finally, if you are using the TOPS–20 tape drive allocation facility, you can charge users for their tape usage. Note that you must use tape drive allocation with labeled tapes, but you can use it with unlabeled tapes also. The accounting usage file contains entries for all tape–mount requests. The TOPS–20 USAGE File Specification describes the formats of these entries and how they are used in reports and billing.

8.4.2 Setting Up the System to Use Tape Labels

To use tape labels, you must have at least one tape drive that is 9–track and has the capability of using tapes at a density of 800, 1600, or 6250 bits per inch (bits/in). There is no restriction on the number of these drives you use. The TOPS–20 Tape Labeling system can be used with as many drives as are allowed for your system.

Also, you must enter the ENABLE TAPE–DRIVE–ALLOCATION command in the n–CONFIG.CMD file. The TOPS–20 KL Model B Installation Guide describes the format of this command and how to enter it into the n–CONFIG.CMD file at the time you install the system. If you do not enter this command during software installation, you can edit the n–CONFIG.CMD file at a later date. However, if you edit the file at a later date, you must shut the system down and bring it back up again to process the commands in the n–CONFIG.CMD file.

8.4.3 Initializing Tapes and Drives to Use Labels

Tapes must be initialized before they can contain labels. An initialized tape contains a volume label set followed by an empty file. The operator issues commands to OPR to initialize tapes for use by the TOPS-20 Tape Labeling system. All the necessary volume labels are then created on a tape. (Refer to the TOPS-20 Operator's Guide for a description of using OPR to initialize tapes.)

You should initialize as many scratch tapes as you will need to store your system's data before users start issuing MOUNT requests for tapes. Then, when a user issues a MOUNT command without specifying a VOLID, the operator mounts an initialized scratch tape of the appropriate label type. TOPS-20 then readies (loads) the tape for write access by the user program.

In addition to initializing tapes, you can set some or all of your tape drives to use the automatic volume recognition facility (AVR). As described earlier, AVR sets your tape drives to automatically recognize tape volumes as they are mounted. To turn on automatic volume recognition, have the operator enter the following command in the <SYSTEM>SYSTEM.CMD file:

```
ENABLE AUTOMATIC-VOLUME-RECOGNITION (FOR) object
```

Where object is either TAPE-DRIVE MTAn: or TAPE-DRIVES.

You can turn off AVR by entering the following command in the <SYSTEM>SYSTEM.CMD file:

```
DISABLE AUTOMATIC-VOLUME-RECOGNITION (FOR) object
```

These commands can be given to OPR at any time to enable or disable AVR for any or all drives on the system.

It is generally a good practice to enable AVR for all drives.

8.5 SHARING TAPE DRIVES BETWEEN TWO SYSTEMS

If you have two DECSYSTEM-20s, you can set up the systems to share a TX02 tape subsystem by use of the TX03 option, as Figure 8-2 shows:

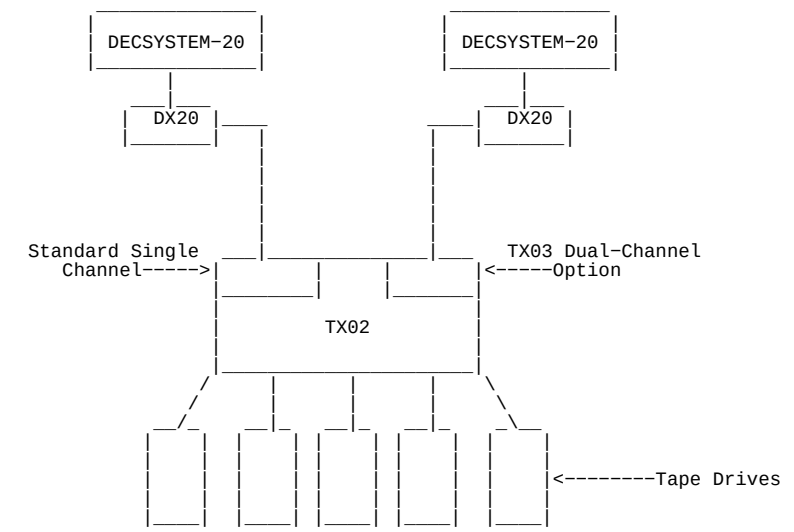


Figure 8-2: TX02 Tape Subsystem

Note, however, that use of the drives must be under strict operator control. Without this control, it is likely that two users, on different systems, will eventually end up using the same tape.

Operator control is required because:

- Unlike disk drives, there is no mechanism to control the porting of a tape drive between systems. If the tape drive is available to the TX02, then it is available to any system to which the TX02 is connected.

- Furthermore, the MOUNTR programs on the two systems do not communicate, so they cannot coordinate access to the drives. Thus, the MOUNTRs on the two systems would allow two users access to the same physical tape drive.

Operator Procedures

1. All drives that are not to be used by a particular system should be set unavailable to MOUNTR with the OPR command:

   OPR> SET TAPE-DRIVE MTAx: UNAVAILABLE

   This command takes away control of the tape drive from MOUNTR and writes an entry in the DEVICE-STATUS.BIN file. During system reloads, MOUNTR reads this file, and if a tape drive has been set unavailable, will never try to access or assign the drive. Note that if DEVICE-STATUS.BIN is ever damaged, then a new file is created at system startup. However, all data and settings will have been lost, including data for the drives that have been set unavailable. Therefore, the operator will need to repeat this step.

2. Setting the drive unavailable to MOUNTR does not prevent a user from reserving the drive with the ASSIGN command. To prevent a user from assigning a tape that is set unavailable to MOUNTR, a program under control of the operator should assign to itself all of the "unavailable" drives. This program should be run at system startup, before any users are allowed to log in.

   You could also control the assignment of tape drives with an access control program. Refer to Section 11.1, Access Control Program.

An example of re-porting a tape drive from system A to system B follows. The operator:

1. Removes the tape (if any) from the tape drive that is to be ported over to system B.

2. Sets the drive unavailable to MOUNTR on system A.

   (OPR> SET TAPE-DRIVE MTAx: UNAVAILABLE)

3. Assigns the drive to a process under control of the operator on system A.

4. Deassigns the drive from the process that is under operator control on system B.

5. Sets the drive available to MOUNTR on system B.

   (OPR> SET TAPE-DRIVE MTAx: AVAILABLE)

| In a CFS-20 cluster, a tape drive can be used from only one system at a time. A user may receive the error message: "Device in use by another system."

CHAPTER 9

SYSTEM PROBLEMS/CRASHES

This chapter describes the actions to take when you are faced with various system problems. You may have to correct a problem with the file system, act immediately after a power failure, or remove the CI from system use. There may be times when you cannot trace a problem. At such times, Digital Field Service can remotely run diagnostic programs on your system. The following sections address these topics.

Errors that require you to correct a problem in the file system seldom occur. However, if a problem of this nature arises, you can perform four classes of file system corrections. From the least to most severe, they are:

- Restore a single file in a directory
- Restore a single directory (other than <ROOT-DIRECTORY>)
- Restore <ROOT-DIRECTORY>
- Restore the entire file system

The TOPS-20 Operator's Guide provides all the necessary information for the operator to correct these types of problems. Sections 9.1 through 9.4 provide you with an overview of how these problems are solved.

9.1 RESTORING A SINGLE FILE

If you receive a request to restore a file for a user, you can use the following procedure.

1. Look in the binder that contains the listing of the DUMPER log files. (Chapter 7 describes creating DUMPER log files.)
2. Write down the file specification (including the structure and directory) to be restored, and the date and number of the tape containing the file. Be sure to indicate the destination structure and directory if one or both are different from the structure and directory from which the file was saved.
3. Submit a request to the operator to restore the file.

9.2 RESTORING A SINGLE DIRECTORY

If you receive a request to restore a directory, you can use the following procedure.

1. Determine the structure that contains the directory.
2. Make sure you have a copy of the files in this directory on a DUMPER tape. (Check the log files.)
3. Give the ^ECREATE command with the LIST subcommand. Write down the list of parameters, for example, the directory number, allocation, etc. You may need this information when you re-create the directory.
4. Give the ^ECREATE command with the KILL subcommand for the directory. (The TOPS-20 Operator's Guide provides additional procedures for deleting a single directory if the ^ECREATE command with the KILL subcommand is unsuccessful.)
5. Using your DUMPER backup tapes, first restore the files from the last FULL-INCREMENTAL DUMPER operation. Then, restore files from each INCREMENTAL tape until the time when the files were lost. Be sure the operator gives the CREATE command to DUMPER to restore the directory parameters.

If the directory contains unreproducible information and it is not backed up on tape, call Digital Software Services for assistance. It may be possible to reconstruct the directory without losing the valid information in it.

9.3 RESTORING <ROOT-DIRECTORY>

Each structure has its own <ROOT-DIRECTORY> and a backup <ROOT-DIRECTORY> that is used by the system if the primary <ROOT-DIRECTORY> is bad. The directory <ROOT-DIRECTORY> contains a pointer to each first-level directory on a structure, as well as several important system files. (Chapter 5 illustrates how <ROOT-DIRECTORY> points to directories.) If <ROOT-DIRECTORY> is lost on the system structure, users cannot access any files in the system. If <ROOT-DIRECTORY> is lost on a mountable structure, users cannot access files on that structure.

You can tell that the <ROOT-DIRECTORY> on the system structure is bad if the operator's console prints any one of the BUGHLTs listed in Table 9-1. When a BUGHLT appears on the console, the system stops. A BUGHLT appears in the form:

```
**********
*BUGHLT name AT dd-mm-yy hh:mm:ss
*JOB:n, USER:  user-name
*ADDITIONAL DATA:  data,data,data
**********
```

The lines beginning with JOB: or ADDITIONAL DATA: may not appear.

Table 9-1: <ROOT-DIRECTORY> BUGHLTS

BUGHLT	Meaning
BADROT	<ROOT-DIRECTORY> is invalid
FILIRD	The system could not initialize <ROOT-DIRECTORY>
FILMAP	The system could not map <ROOT-DIRECTORY> into memory
BADXT1	The index table is missing and cannot be created

<ROOT-DIRECTORY> may also be bad if you get the BOOT error ?FIL NOT FND. If this error appears, be sure you have mounted all devices correctly.

To recover from a bad <ROOT-DIRECTORY>, first determine which structure contains the bad directory. If the bad <ROOT-DIRECTORY> is on a mountable structure, you can run CHECKD with RECONSTRUCT ROOT-DIRECTORY and specify the proper structure. The TOPS-20 Operator's Guide details the procedure for determining the structure and using CHECKD for reconstructing <ROOT-DIRECTORY> on mountable structures.

If the bad <ROOT-DIRECTORY> is on the system structure, you can instruct the system to use the backup system structure <ROOT-DIRECTORY> and rebuild this directory. Section 9.3.1 describes this procedure.

NOTE

If your first attempt to rebuild <ROOT-DIRECTORY> fails, call your DIGITAL Field Service Representative. NEVER try to rebuild this directory twice on any structure.

9.3.1 Rebuilding the System Structure <ROOT-DIRECTORY>

To rebuild <ROOT-DIRECTORY> on the system structure, halt the central processor and perform Steps 7 through 21 in Chapter 2 of the TOPS-20 KL10 Model B Installation Guide. The steps are shown below for reference.

1. Type CTRL/backslash on the operator's console; the system prints PAR>.

2. Type SHUTDOWN and press the RETURN key; the system prints a few message lines.

```
PAR>SHUTDOWN<RET>
**HALTED**

%DECSYSTEM-20 NOT RUNNING
```

3. Mount System Floppy A in drive 0 (Step 7).

4. Mount System Floppy B in drive 1 (Step 8).

5. Mount the TOPS-20 Installation Tape on MTA0: (Step 9).

   If the TOPS-20 Installation Tape is not your most recent system backup tape, mount your SYSTEM BACKUP TAPE that contains the monitor, TOPS-20 Command Processor, DLUSER, DLUSER data, DUMPER, and save sets containing <SYSTEM> and <SUBSYS>. (Refer to Section 3.2 for a description of special system directories.)

6. Place the front-end HALT switch in the ENABLE position (Step 10).

7. Set the front-end switch register to 000007 (octal) (Step 11).

8. Press the ENABLE and SWITCH REGISTER switches simultaneously (Step 12).

```
RSX-20F VB16-00 8:55 1-JAN-88

[SY0:  REDIRECTED TO DX0:]
[DX0:  MOUNTED]
[DX1:  MOUNTED]
KLI -- VERSION VB1600 RUNNING
KLI -- ENTER DIALOG [NO,YES,EXIT,BOOT]?
KLI>
```

NOTE

The version and edit numbers in this manual may differ from the numbers printed on your console. The numbers on your console must be equal to or greater than the numbers in this manual.

9. Type YES and press the RETURN key (Step 13).

```
KLI>YES<RET>
KLI -- KL10 S/N:  2136., MODEL B, 60 HERTZ
KLI -- KL10 HARDWARE ENVIRONMENT
         MOS MASTER OSCILLATOR
         EXTENDED ADDRESSING
         INTERNAL CHANNELS
         CACHE
KLI -- RELOAD MICROCODE [YES,VERIFY,FIX,NO]?
KLI>
```

10. Type YES KLX and press the RETURN key (Step 14).

```
KLI>YES KLX<RET>
KLI -- MICROCODE VERSION 2.0 [407] LOADED
```

NOTE

If your system has cache memory, the following question appears previous to the question in Step 11. (Step 15.)

```
KLI -- RECONFIGURE CACHE (FILE,ALL,YES,NO)?
```

Type ALL and press the RETURN key (Step 16).

```
KLI>ALL<RET>
KLI -- ALL CACHES ENABLED
KLI -- CONFIGURE KL MEMORY [FILE, ALL, REVERSE,  FORCE,  YES,
NO]?
KLI>
```

11. Type ALL and press the RETURN key (Step 17).

```
KLI -- CONFIGURE KL MEMORY [FILE, ALL, REVERSE,  FORCE,  YES,
NO]?
KLI>ALL<RET>
LOGICAL MEMORY CONFIGURATION
.
.
.
KLI -- LOAD KL BOOTSTRAP [YES, NO, FILENAME]?
KLI>
```

12. Type MTBOOT and press the RETURN key (Steps 18 and 19).

```
KLI>MTBOOT<RET>
KLI -- CONFIGURATION FILE ALTERED
KLI -- WRITE CONFIGURATION FILE [YES,NO]?
KLI>NO<RET>
BOOTSTRAP LOADED AND STARTED

BOOT V11.0(315)

MTBOOT>
```

13. Type /L and press the RETURN key (Step 20).

```
MTBOOT> /L<RET>
[BOOT:  STARTING CHN:n DX20x:0 MICROCODE Vn(n)][OK]
[BOOT:  LOADING][OK]
MTBOOT>
```

NOTE

The message concerning the DX20 microcode is printed only if you are installing the TOPS-20 software on a DECSYSTEM-20 with a DX20 tape or disk controller.

14. Type /G143 and press the RETURN key (Step 21).

```
MTBOOT> /G143<RET>

[FOR ADDITIONAL INFORMATION TYPE "?" TO ANY OF THE
FOLLOWING QUESTIONS.]
DO YOU WANT TO REPLACE THE FILE SYSTEM ON THE SYSTEM
STRUCTURE?
```

NOTE

Read Step 15 carefully before answering this question.

15. Type N and press the RETURN key. You DO NOT want to clear all the information on the disk packs. If you want to retain all the information in the file system, always type N.

```
DO YOU WANT TO REPLACE THE FILE SYSTEM ON
THE SYSTEM STRUCTURE?  N<RET>

[PS MOUNTED]

RECONSTRUCT ROOT-DIRECTORY?
```

16. Type Y and press the RETURN key. This causes the backup copy of <ROOT-DIRECTORY> to be used.

```
RECONSTRUCT ROOT-DIRECTORY?  Y<RET>

[RECONSTRUCTION PHASE 1 COMPLETED]

%%NO SETSPD

System restarting, wait...
ENTER CURRENT DATE AND TIME:
```

The system restarts and runs CHECKD to reconstruct the bit table. The bit table contains one bit for every page in the file system. If the bit is on, the page is available; if the bit is off, the page is in use.

17. Type the date and time and press the RETURN key. Type Y and press the RETURN key to confirm the date and time.

```
ENTER CURRENT DATE AND TIME:  1-JAN-81 0931<RET>

YOU HAVE ENTERED THURSDAY, 01-JANUARY-1981 9:31AM,
IS THIS CORRECT (Y,N) Y<RET>
WHY RELOAD?
```

18. Type OTHER and press the RETURN key.

```
WHY RELOAD?  OTHER<RET>
[REBUILDING BIT TABLE]
```

NOTE

You should type a response to WHY RELOAD?, that reminds you of why you did this procedure. This response is stored in the <SYSTEM-ERROR>ERROR.SYS file. Refer to the TOPS-20 KL10 Model B Installation Guide for a complete list of valid abbreviations.

19. The system prints some standard messages, the output from CHECKD, RUNNING DDMP, and the output from SYSJOB and PTYCON.

    [REBUILDING BIT TABLE]

    [WORKING ON STRUCTURE - PS:]

    output from CHECKD

    RUNNING DDMP

    output from SYSJOB and PTYCON

    Refer to the TOPS-20 Operator's Guide for samples of the output from CHECKD, SYSJOB and PTYCON.

20. Log in as user OPERATOR.

```
 TOPS-20 SYSTEM, TOPS-20 Monitor 7(21017)
@LOGIN (USER) OPERATOR (PASSWORD) --- (ACCOUNT) OPERATOR<RET>
 Job 1 On TTY1 3-MAY-88 10:33:32
```

9.4 RESTORING THE ENTIRE FILE SYSTEM

If you are still receiving random errors and cannot use the system, you may have to restore the entire file system on the system structure. Before doing this, you should contact your software specialist to ensure that resorting to this procedure is necessary. The procedure requires shutting down the system and reinstalling the file system.

9.4.1 Re-creating the File System on the System Structure

The following steps outline the procedure for restoring the file system on the system structure.

1. Type CTRL/backslash to start the front-end command parser.

   CTRL/backslash

   PAR>

2. Type SHUTDOWN to stop the central processor; the system prints a few messages.

```
PAR>SHUTDOWN<RET>
**HALTED**
%DECSYSTEM-20 NOT RUNNING
```


3. Start at Step 9 in Chapter 2 of the TOPS-20 KL Model B Installation Guide and follow all the steps through Step 60.

   In Step 9, instead of mounting the TOPS-20 Installation Tape, mount your system structure SYSTEM BACKUP TAPE that contains the monitor, TOPS-20 Command Processor, DLUSER, DLUSER data, DUMPER, and save sets containing <SYSTEM> and <SUBSYS>.

4. After performing Step 60, restore your entire file system using DUMPER. First run DUMPER, then mount your first reel of the most recent full DUMPER tape and follow the procedures in the TOPS-20 Operator's Guide.

5. Re-create the front-end file system by following the directions in Chapter 4 of the TOPS-20 KL Model B Installation Guide.

6. Finally, restart the system by following the directions in Chapter 5 of the TOPS-20 KL Model B Installation Guide.

NOTE

Restore the incremental saves to obtain the most recently saved files. Before restoring each incremental tape, type DUMPER>CREATE to restore all the directories that were created since the last time you ran the DLUSER program.

9.4.2 Re-creating Mountable Structures

If, after your efforts to correct errors on a structure other than the system structure (using the command RECONSTRUCT ROOT-DIRECTORY to CHECKD), you still cannot use the structure, you can re-create that structure and restore all the directories and files. You can restore structures other than the system structure during timesharing.

To restore a mountable structure you must:

1. Give the SET STRUCTURE IGNORED command to the OPR program to prevent other users from mounting the structure while you are re-creating it.

2. Run CHECKD to create the structure.

3. Run DLUSER to restore directory parameters for the structure if you previously used the program to save the parameters.

4. Run DUMPER using the backup tapes for this structure. Give the CREATE and RESTORE commands to DUMPER to restore all the directories and files.

The TOPS-20 Operator's Guide details the procedures for running CHECKD and DUMPER to re-create a structure.

9.5 POWER FAILURES

Unfortunately, power failures and brown-outs sometimes occur at installations. You should be aware of the immediate steps to perform and transmit this information to your operations people. The kind of attention you should direct to this type of problem depends on the type of outage you have. These steps can protect your system from physical damage and perhaps unnecessary loss of files.
If your system experiences a total power failure, you should:

1. Immediately power-off all components of the system.

2. Inform your DIGITAL Field Service representative as to when you expect to resume power. The field service representative may ask to be present while you bring your system back up.

If your system experiences a short instance of a power failure or brown-out, the system may recover on its own. You may not have any problems and, usually, all your data remains intact. If you notice a problem, call your Field Service representative.

High-performance computer systems are sensitive to the quality of the electrical power supply. An investment in a power conditioner or an uninterruptible power supply may more than repay itself in improved system reliability and availability. Your Field Service representative may be able to assist you in evaluating the need for such equipment.

9.6 REMOTE DIAGNOSTIC LINK (KLINIK)

You may occasionally have a problem with your system and cannot determine the cause. The remote diagnostic link, available on all systems, allows a DIGITAL Field Service engineer to access your system from a remote location and run diagnostic programs. This capability is called KLINIK. The DIGITAL engineer accesses KLINIK through a terminal and telephone line at the DIGITAL Service Center. The TOPS-20 Operator's Guide describes when and how to use the KLINIK capability.

9.7 MAKING THE CI UNAVAILABLE ON NON-CFS SYSTEMS

The CI, a computer-interconnect bus, is a key piece of hardware for connecting systems in a CFS-20 configuration and for connecting HSC50-based disks (RA81s and RA60s) to a system. Ordinarily, you need do nothing at all to operate the CI. However, you may need to disengage a system from the CI so Field Service personnel can correct problems with the CI20 or the HSC50. At those times, you should instruct the operator to make the CI unavailable by means of the SET PORT CI UNAVAILABLE command. (Refer to the TOPS-20 Operator's Guide for details.)

When the CI is unavailable to a system, users cannot access HSC50-based disks, which rely on the CI to transmit data. The procedure calls for the operator to dismount any structures that the system indicates are mounted on these disk drives.

To put the CI back in operation, the operator gives the command:

```
OPR>SET PORT CI AVAILABLE<RET>
```

Structures that were affected must then be remounted.

Refer to Chapter 12, THE COMMON FILE SYSTEM, for information on disengaging the CI on CFS systems.

9.8 MAKING THE NI UNAVAILABLE

A system may need to be disengaged from the NI so that field service personnel can diagnose problems with the NI or the NIA20. To make the NI unavailable to a system, the operator gives the command:

```
OPR>SET PORT NI UNAVAILABLE<RET>
```

This command prevents users of the system from using LAT terminal servers as well as any other software that uses the NI. If DECnet or TCP/IP software is installed, it cannot use the NI for data transfer between systems. To make the NI available again, the operator gives the command:

```
OPR>SET PORT NI AVAILABLE<RET>
```

9.9 OFFLINE DISKS

When a disk unit becomes unavailable to a system, all jobs accessing the disk "hang" until the disk becomes available again. When the disk comes back online, input/output activity continues from the point where it left off.

Also, some jobs may try to begin accessing the disk after it went offline but before it becomes available again. They hang in the same way that interrupted jobs, described above, hang. You can specify that offline disks be made unavailable to these new jobs by putting the following command in the n−CONFIG.CMD file:

ENABLE OFFLINE−STRUCTURES mm:ss

where:

mm:ss is the "structure timeout interval" in minutes:seconds format. The structure timeout interval is the length of time from when the system recognizes that a disk unit has gone offline to when it marks the structure as offline. The maximum value for mm:ss is 15 minutes; the minimum is 1 second.

It should be noted that if just one disk of a multidisk structure becomes unavailable, then the entire structure is marked offline. After a structure is marked offline, it becomes unavailable to new jobs requesting input/output service. The system sends an error message to any requestor trying to access the structure.

This command is in effect by default with a timeout interval of 5 seconds. To disable the feature, enter the following command in the configuration file:

DISABLE OFFLINE−STRUCTURES

A good reason to disable the feature is to prevent batch jobs from terminating when they receive the error message indicating that the desired structure has been set offline. You may want batch jobs to hang until the disk is restored for use.

9.9.1 Operator Procedures

Privileged commands corresponding to the ones above are available to the operator during timesharing:

$^ESET OFFLINE−STRUCTURES mm:ss

$^ESET NO OFFLINE−STRUCUTRES

Operators may want to set the timeout interval to a value higher than five seconds if they are able to correct the disk problem within the specified period. However, new user requests will then hang until the disk is marked offline (or the problem is corrected).

9.10 DUMPING ON NON−FATAL SYSTEM ERRORS

The monitor can dump its memory area to a disk file when BUGCHKs and BUGINFs occur. This feature, called DUMP−ON−BUGCHK, helps you debug the system of nonfatal, "continuable" errors by providing a dump file for examination.

9.10.1 Enabling DUMP−ON−BUGCHK

The DUMP−ON−BUGCHK feature is enabled in the n−CONFIG.CMD file with the following command:

ENABLE DUMP−ON−BUGCHK FACILITY

At least one of the following commands is further required to enable dumping of all or specific BUGCHKs or BUGINFs that can be dumped:

ENABLE DUMP−ON−BUGCHK ALL−BUGCHKS
ENABLE DUMP−ON−BUGCHK ALL−BUGINFS
ENABLE DUMP−ON−BUGCHK BUG bugname

where: bugname is the name of a BUGCHK or BUGINF

With the first two commands, each BUGCHK or BUGINF is dumped only once per loading of the system. The third command causes a dump to be taken each time that the specified BUGCHK or BUGINF occurs. Dumps are taken only as often as the DUMP−ON−BUGCHK timeout allows, which is 15 seconds by default. The following command overrides the timeout constraint and allows a dump to be taken as often as a specified bug occurs:

ENABLE DUMP−ON−BUGCHK BUG bugname IGNORE−DUMP−TIMEOUT

9.10.2 Disabling DUMP−ON−BUGCHK

You may want to disable this feature to save memory space that the monitor uses to implement it, or after you've looked at the considerations in Section 9.10.5. You could save the feature for times when the system is experiencing problems.

If the feature is disabled, the system produces only the "crash" dumps associated with fatal errors, but not "continuable" dumps. To disable DUMP−ON−BUGCHK, enter the following command in the n−CONFIG.CMD file:

DISABLE DUMP−ON−BUGCHK

The DOB−ON−BUGCHK facility is disabled by default.

9.10.3 "Dumpable Structures"

The operator can specify where continuable dumps are written with the command:

OPR>SET STRUCTURE str: DUMPABLE<RET>

where:

str: is the name of a structure that is to be considered "dumpable."

This command can be given for more than one structure. Note that the system structure is always dumpable.

The dumpable status of a structure remains intact across crashes and system reloads.

A continuable dump is written to the <SYSTEM>DUMP.EXE file on a dumpable structure. The system writes the file to the first structure that meets the following conditions:

- The structure is dumpable.
- The structure is not being initialized or dismounted.
- The structure is not offline, in maintenance mode, or write locked.
- The data in the monitor's structure data block for the structure appears to be uncorrupted.
- Any <SYSTEM>DUMP.EXE files on the structure have been copied (see the following section) and therefore will not be overwritten.

9.10.4 Copying the Dump File

After the dump, the n-SETSPD program scans all dumpable structures and copies any previously uncopied <SYSTEM>DUMP.EXE files to an area on DMP:, if this logical name is defined. Otherwise, it copies the files to other areas on the same structures. The new files have unique names of the form:

str:<DUMPS>DUMP-nnnn-bug.CPY.n

where:

str is the name of the structure

nnnn is the edit number of the monitor that was running at the time of the crash

bug is the name of the BUGCHK or BUGINF

n is the file generation number

An informational message similar to the following is sent to the CTY after each copy:

```
Copying system dump
       from:  ABC:<SYSTEM>DUMP.EXE.1
       to:    XYZ:<DUMPS>DUMP-21002-FSPOUT.CPY.1
```

9.10.5 Time Considerations

The DUMP-ON-BUGCHK facility (DOB) turns off timesharing while the system is being dumped. If DOB takes an extended period of time, LAT, DECnet, or INTERNET connections may be lost. Therefore, you should specify the fastest available device for DOB dumps. The following are sample DOB times for various disk types and memory sizes:

Device	Memory Size (megawords)	DOB Time (seconds)
RP07	1.5	06
RP07	4.0	16
RP06	1.5	11
RP06	4.0	32
RA60/81	1.5	21
RA60/81	4.0	56

Problems could occur in the following areas during extended dump periods:

- Data from Terminal Lines

  During continuable dumps, the system is not in sufficient communication with the front end to ensure that terminal input data gets processed. (The monitor enters "secondary protocol" -- refer to the RSX-20F System Reference Manual for details). This data could get lost. It is a problem particularly with lines doing high-speed input.

- DECnet

  If other nodes in a DECnet network have their timeout periods set at a value lower than the time it takes for a continuable dump on this system, inter-system links could timeout.

- LAT

  A LAT server has a default maximum timeout value of two minutes, which should suffice for continuable dumps on the host. However you can decrease the server timeout period. If you make it too short, all jobs on the host could become detached after the dump.

9.10.6 Controlling DUMP-ON-BUGCHK

For your convenience, there is an unsupported program on the tools tape that can help you control the DUMP-ON-BUGCHK facility. The tool is called DOBOPR. It includes documentation that you can access with the program's HELP command.

CHAPTER 10

SYSTEM PERFORMANCE

The configurations of systems running TOPS–20 and the mix of jobs on these systems vary from installation to installation. TOPS–20 is designed to provide better response to interactive users than to provide higher throughput to computational tasks. On systems with a typical mix of interactive and batch jobs, this design provides both good response and adequate throughput. Therefore, if your system has many timesharing users and an average amount of batch processing jobs, you will find that your system's response is good and most users are satisfied.

If you have a mix of jobs or a configuration different from a typical system, you may want to change the response of your system to provide better service to specific users. TOPS–20 provides several tuning mechanisms that allow you to experiment with and change the behavior of your system. One mechanism allows you to favor classes of users by allocating each class a specified percentage of the CPU. Another mechanism allows you to favor computational jobs over interactive jobs. A third mechanism allows you to disable features that normally provide better performance, but that may not be applicable at your installation.

This chapter describes the mechanisms for tuning your system's response and provides guidelines for using these mechanisms. Because the response of your system can be felt during actual use only, you can expect system tuning to be an experimental and iterative process. By analyzing the statistics from the WATCH program and by gathering inputs from your users, you can determine the best way to tune your system.

The tuning mechanisms include:

- The class scheduler, which allows you to divide the central processor (CPU) resource among groups of users
- Assigning low priority to batch jobs for installations that do not use the class scheduler

- Bias control, which allows you to favor either interactive or compute–bound jobs on your system
- The program name cache for improving the startup time of frequently used programs
- Reinitializing disk packs in heavily used structures for improving file processing time

Each mechanism can be used independently. Unless otherwise noted, there is no interrelationship among them.

10.1 THE CLASS SCHEDULER

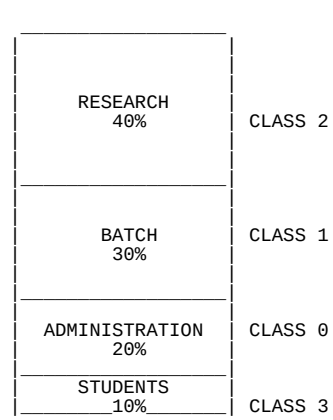
Occasionally, certain jobs monopolize the central processor's time while other more critical jobs wait for the CPU. You may want to control the amount of CPU time a job receives. You can use the class scheduler to provide an even distribution of CPU time or to provide more CPU time to critical jobs on the system. Some system managers may want to allocate a larger amount of processor time to special users than to other system users. Sections 10.1.1 through 10.1.9 describe:

- an overview of the class scheduler
- who should use the class scheduler
- how to begin using the class scheduler
- turning on the class scheduler
- changing class percentages
- disabling the class scheduler
- getting information about class scheduler status
- a sample session using class scheduler commands
- an alternative to using the class scheduler with accounts.

10.1.1 Overview

The class scheduler allows you to allocate percentages of the central processor's time to individual classes of users. Each job in a class receives a portion of the class percentage. Therefore, by using the class scheduler, you can provide a consistent service to predefined groups of users.

The diagram below illustrates the concept of classes and percentages of CPU time allocated to each class. Note that you can set up a class to include all batch jobs. This allows you to control batch jobs separately from timesharing jobs. The batch class is given a high percentage or a low percentage. Now, each time a user submits a batch job, the scheduler uses the batch class and not the user's class. Section 10.1.4, Step 5, describes how to create a batch class.



You can define class memberships by using the TOPS-20 accounting facility or by writing your own access control program. Section 10.1.9 provides an overview of using an access control program to define classes. The following description applies to using the class scheduler with the accounting facility.

Using the accounting facility, you can associate a class number with each account on the system. Each account has only one class associated with it. Therefore, only a user who has access to more than one account can have access to more than one class. The user changes classes by changing accounts. To use this method, you must enable account validation so that users are required to use a valid account. Section 10.1.4 describes the accounting file entries.

The scheduler periodically computes the time used by classes and individual jobs within a class. The scheduler's first concern is that a class receive its share of the CPU time. The scheduler gives a greater percentage of time to the class that is the farthest away from its target (the total percentage allocated to the class). The scheduler's second concern is that each job within a class receive its fair share. Periodically, the scheduler computes the average amount

of CPU time that a job has accrued. This quantity determines the job's priority within the class. A job that is requesting the CPU and is farthest way from receiving its fair share within the class receives a greater percentage of time within that class.

Generally, the CPU has unused time. This unused time is called windfall. Windfall occurs if one or more classes has no logged-in jobs, or if one or more classes has an insufficient demand for the CPU to use all of its class percentage.

You can handle windfall in one of two ways: allocate it or withhold it. Allocating windfall means that the excess CPU time is awarded proportionately to the active classes. (In this discussion, the term active class refers to a class that has logged-in users requesting CPU time.) Higher percentage classes receive proportionately more of the available windfall than lower percentage classes. This means classes can receive slightly more than their allowed percentages when there is windfall available. Note that windfall is distributed proportionately to each class and not to each job within a class.

Withholding windfall means that the excess CPU time is idle time and it is not distributed to the active classes. It also means that classes do not receive more than the percentage allowed for them. Usually, it is better to allocate windfall, and not withhold it so you don't throw away valuable computer time.

10.1.2 Who Should Use the Class Scheduler?

As mentioned earlier, the class scheduler allows you to divide the user community into defined classes and allocate a percentage of CPU time to each class. This intended use may not be beneficial or practical for some systems. Tradeoffs do exist and some installations do not require class scheduling. Therefore, read the following paragraphs and determine if your system needs this type of control. Several instances that can benefit from using the class scheduler are:

- You can elect to sell portions of CPU time. For example, customer X can purchase some percentage of computer time at a proportional cost. You should, in this case, invite customer X to your installation and provide an environment that simulates the kind of response this customer can expect at this percentage. Because it is impossible to create the environment exactly as the customer will experience it at all times, customer X may decide later to change the purchase to a different percentage.

- If you have a natural division among the users of the system, you can divide these users into classes, then distribute CPU time to these classes with respect to their importance on the system. For example, in an educational environment, you may have administrative users (doing payroll, grades, etc.), faculty users (perhaps working on a funded research project), and student users (who are computer science majors). Using the class scheduler, you can establish three classes and distribute the CPU time according to the importance or pay rate of each class.

- If you wish to give preference to batch jobs, you can use the class scheduler to give the batch class a high percentage of the CPU.

Conversely, you may not want to use the class scheduler for some of the following reasons:

- If you have not realized a need in the past for class scheduling, or if you have a new system and do not see an immediate requirement for class scheduling, you should not set it up.

- Systems with minimal amounts of memory may experience an increase in swapping. This additional overhead may offset any advantage gained by using the class scheduler.

- In addition to other tasks, the scheduler constantly maintains usage values for each class and job. Because of this extra overhead, the overall system throughput decreases. Therefore, use the class scheduler if allocating percentages to individually defined classes outweighs the throughput depreciation.

- To give batch jobs a low priority on the system, you do not have to use the class scheduler. Section 10.2 describes placing the BATCH-BACKGROUND command in the n-CONFIG.CMD file to place all batch jobs in a low priority (or background) queue.

10.1.3 How to Begin Using the Class Scheduler

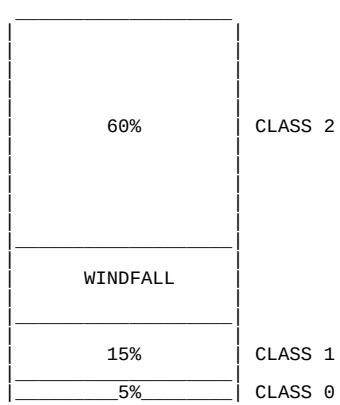
If you elect to use the class scheduler, first divide the system users into classes. Next, determine the amount of CPU time each class should receive. Percentages given to classes are typically in units of 5, that is, 5%, 10%, 15%, ...100%. The sum of the percentages given to all classes cannot exceed 100 percent. The result of these two steps depends on the reason you elected to use the class scheduler, and how you plan to use it at your installation. Because of this system dependency, step-by-step procedures for selecting classes and allocating the CPU cannot be given. The following discussion provides you with guidelines for setting up the class scheduler to meet your system's needs.

- Start with as few classes as possible, three or perhaps four. The class scheduler allows eight. Later, you can divide classes further, if necessary.

- Determine the number of logged-in users you expect in each class. Be sure that you do not overload a class, making it impossible to give a sufficient percentage of the CPU to each job in the class. Also, you can limit the number of users in a class, but you cannot limit the number of jobs. For example, the SUBMIT command creates an additional job. Therefore, consider the type of work that users in a class perform.

- Estimate the percentage of CPU time (or time purchased) each class should receive. This is a difficult step; however, you can experiment with and later alter the percentages you choose. (Section 10.1.5 describes how you can change class percentages.) If a class consists of a large number of users who are generally logged in at the same time, be sure to give this class sufficient CPU time. For example, using the diagram below, class 2 has 60 members. It also has 60 percent of the CPU. This means that if approximately 60 jobs in class 2 are demanding the CPU, each job will receive approximately 1 percent. (It may be slightly greater than 1 percent if you allocate windfall.) This also means that on a per-job basis, users in class 2, with the higher percentage, may not get as much of the CPU as users in class 0 or class 1.

- In order to avoid possible performance problems, no class should be allocated less than 5% of the CPU.



- The above diagram illustrates that you can allocate less than 100 percent of the CPU to classes. However, the total percentage for all classes cannot exceed 100 percent.

- The system uses class 0 as a default for any account that does not have a valid class assigned to it. Therefore, you can divide a portion of the system into well-defined classes and have all other users receive the percentage given to the default class.

- You can set up the class scheduler so that one class, perhaps the default class, receives only windfall. For example, suppose you allow some users to log into an account that is not yet assigned a class. These users are assigned to the default class 0. Or, suppose several users log in infrequently, read mail, and perform little computing. These users may log into an account that is not assigned a class (and, therefore, are assigned the default class, 0), or an account that is associated with class 0. You subsequently assign a zero percentage to class 0. These users may receive enough windfall to get their job done. However, they are not allocated CPU time and can have periods of extremely slow or no response. You may find, after experimenting with this type of procedure, that it is better to associate each account with a class and give the class a very low percentage of the CPU.

- Situations may arise that affect the scheduler's ability to give a class its percentage of the CPU. For example, the members of a class cannot use the amount of CPU time allocated to the class. Also, a situation may arise where the demands of the class exceed the percentage of CPU time given the class, and windfall is available but not allocated. In either case, you should reevaluate both the percentages given to classes, and whether or not you want to continue withholding or allocating windfall.

10.1.4 Procedures to Turn On the Class Scheduler

After dividing users into classes and estimating the amount of CPU time, follow these steps to set up classes and turn on the class scheduler.

Step	Procedure
1.	Edit the <ACCOUNTS>ACCOUNTS.CMD file on the system structure (and the subaccount files, if any, that contain all your accounting data). Follow the procedure in Chapter 6, Creating Accounts, for modifying or creating each account with the /CLASS:n switch. For example, ACCOUNT COMP1/CLASS:1 USERS KOHN,HOLLAND,MILLER means that when users Kohn, Holland, and Miller log into or change their account to COMP1, they are placed in class 1. Each account has only one class associated with it. Therefore, only the user who has access to more than one account can have access to more than one class. A job can be in only one class at any one time.
2.	Run the ACTGEN program. Give the TAKE command and specify the <ACCOUNTS>ACCOUNTS.CMD file (or the file containing the accounting data).
3.	When ACTGEN returns its prompt, give the INSTALL command. This command updates the ACCOUNTS-TABLE.BIN file that the system uses to validate accounts.
4.	Edit the n-CONFIG.CMD file to include the CREATE commands that specify the classes. Be sure account validation is enabled. That is, check that the DISABLE ACCOUNT VALIDATION command is NOT in the file. (The system default is ENABLED.) If account validation is disabled, users can use any account and therefore any class they choose. Enter the following commands in the order shown below.

The CREATE command defines the class number and the percentage of CPU time the class should receive. For example,

CREATE 0 .40

means that the default class, 0, should receive 40 percent of the CPU. Enter the CREATE command for each of the classes that you defined in the ACCOUNTS.CMD file. For example,

```
CREATE 0 .40
CREATE 1 .20
CREATE 2 .30
```

NOTE

Remember that class 0 is the default class. Any user whose account (this includes subaccounts) is not associated with a specific class is placed in class 0.

5. Next, if you have decided to place all batch jobs in a separate class, enter the BATCH-CLASS command. For example,

BATCH-CLASS 2

means that all batch jobs will be placed in class 2 regardless of the user's associated class. You must use the CREATE command to define the percentage of CPU time that the batch class should receive. Note that timesharing users can be in the same class as well, if the account they use is associated with that class. The CREATE command in step 4 defines class 2 to receive 30 percent of the processor. If you do not place a BATCH-CLASS command in the n-CONFIG.CMD file, the scheduler treats all batch jobs the same as timesharing jobs.

6. Finally, enter the ENABLE CLASS-SCHEDULING command. Be sure this is the last command entered. If you reverse the order, that is, you enter ENABLE CLASS-SCHEDULING before the CREATE commands and BATCH command, all users will receive zero percent of the CPU.

The ENABLE CLASS-SCHEDULING command turns on the class scheduler at the next system reload. It also specifies if you are using accounting or an access control program (policy program) for class scheduling and if you are allocating or withholding the windfall. The format of this command is:


```
ENABLE CLASS-SCHEDULING ACCOUNTS         WITHHELD
                        POLICY-PROGRAM   ALLOCATED
```

For example,

ENABLE CLASS-SCHEDULING ACCOUNTS ALLOCATED

means turn on the class scheduler, use the accounting method, and allocate the windfall. Always allocate the windfall. Do not withhold windfall unless you have a very good reason to do so, and you are sure that your class scheme works.

The entry,

ENABLE CLASS-SCHEDULING POLICY-PROGRAM ALLOCATED

means turn on the class scheduler, use the policy (access control) program, and allocate the windfall.

7. At the next system reload (the system is brought down and back up again) the new commands in the n-CONFIG.CMD file take effect.

10.1.5 Changing Class Percentages During Timesharing

During timesharing, you can change the percentage that a class receives by using the SET command to OPR. This change lasts until either the next system reload, or you make another change using the SET command. The format of the SET command appears:

SET SCHEDULER CLASS (number) m (to percent) n

n can be from 0-99 inclusive. Note that the decimal point required in the CREATE command is not allowed in this command.

To make the percentage that a class receives permanent, edit the CREATE commands in the n-CONFIG.CMD file. For example, you can edit the commands

```
CREATE O .60
CREATE 1 .30
CREATE 2 .10
```

to

```
CREATE 0 .10
CREATE 1 .05
CREATE 2 .80
```

Next, reload the system to process the commands in the n-CONFIG.CMD file. The classes that received a low percentage before you made the change now receive a higher percentage of CPU time.

Changing class percentage may be useful if you decide that users in one or two classes should receive a greater percentage of CPU time than other classes during the day. However, these high-percentage classes may not require this time during the evening shift. If you have a recurring need for such changes, you may want to put the appropriate commands into files for use with the TAKE command in OPR.

10.1.6 Disabling the Class Scheduler During Timesharing

You can turn the class scheduler off and back on during timesharing by using the DISABLE and ENABLE commands to OPR. The class scheduler remembers the class percentages that were in effect before the DISABLE command was given. For example, suppose you use the SET command to change the percentage that class 1 should receive from 05 to 15 percent. Then, you give the DISABLE CLASS-SCHEDULER command, and later give the ENABLE CLASS-SCHEDULER command. The class scheduler uses 15 percent for class 1. If you reload the system after disabling the class scheduler, the class scheduler is enabled again and uses the percentages given in the n-CONFIG.CMD file.

10.1.7 Getting Information About Class Scheduler Status

Several TOPS-20 commands provide information about different class scheduler statistics.

The INFORMATION (ABOUT) SYSTEM-STATUS command informs you if:

- the class scheduler is enabled or disabled
- the accounting method or an access control program is being used
- windfall is allocated or withheld
- the system's batch jobs are in a separate class.

For example

```
@INFORMATION (ABOUT) SYSTEM-STATUS<RET>

 CLASS SCHEDULING BY ACCOUNTS ENABLED, WINDFALL ALLOCATED, BATCH CLASS 1.
```


The INFORMATION (ABOUT) MONITOR-STATISTICS command provides a table that shows each active class, its target use of the CPU, its current use of the CPU, and the load averages for that class. For example,

```
@INFORMATION (ABOUT) MONITOR-STATISTICS<RET>

 Class Share   Use   Loads

     0  0.80  0.79    6.56   4.36   3.57
     1  0.15  0.21    5.46   1.38    .95
     2  0.05  0.00    0.00   0.00   0.00
```

The SYSTAT command outputs load averages for either the entire system or a specific class. When the class scheduler is disabled, these averages represent the load of the entire system. For example,

```
@SYSTAT<RET>
 Wed 11-Jul-88 10:17:09  Up 11:38:12
 52+16 Jobs   Load av  5.30  4.03   4.86
```

The last three numbers following Load av indicate the average number of runnable processes over a period of one minute, five minutes, and fifteen minutes. These numbers start at zero. The higher the numbers the longer a job has to wait for CPU time on the average. Using the example, over a fifteen minute period, a given job demanding the CPU waits approximately 4.86 times longer to run than it would if it were the only job running on the system.

When the class scheduler is enabled, these load averages represent the status of the job doing the SYSTAT command and not the entire system. For example,

```
@SYSTAT<RET>
 Wed 11-Jul-88 10:28:07  Up  11:49:12
 52+10 Jobs   Load av (class 1)   1.79   2.36    2.88
```

The last three numbers following Load av indicate the load averages of the class that the job giving the SYSTAT command is in. The SYSTAT command with the CLASS argument provides a breakdown of each job on the system. This breakdown includes the class each job is in, the average share of the class percentage that this job can receive, and how much CPU time the job is currently using. The TOPS-20 Commands Reference Manual describes these commands in detail.

10.1.8 A Sample Session

The following examples show:

- The ACCOUNTS.CMD file after associating classes with accounts.
- The procedures that you follow after editing all your accounting files.
- A sample of the class scheduling commands that are placed in the 7-CONFIG.CMD file.

```
!The <ACCOUNTS>ACCOUNTS.CMD file has been edited.

$TYPE (FILE) MAIN:<ACCOUNTS>ACCOUNTS.CMD<RET>

ACCOUNT MYBANK/CLASS:2
USERS BLOUNT,KONEN,ENGEL

ACCOUNT TRUST/CLASS:1
USERS BRAITHWAITE,HURLEY,HALL,CRISS

ACCOUNT OVERHEAD
USERS SAMBERG,BERKOWITZ,TAYLOR

ACCOUNT PROG/CLASS:3
USERS BLOUNT,KOHN,HOLLAND

$

!Notice that account OVERHEAD uses the default class, 0.

!Next, run the ACTGEN program.

$RUN ACTGEN<RET>
ACTGEN>TAKE (COMMANDS FROM) MAIN:<ACCOUNTS>ACCOUNTS.CMD<RET>

!After ACTGEN finishes and returns its prompt, give the
!INSTALL command to create a new version of the
!<SYSTEM>ACCOUNTS-TABLE.BIN file

ACTGEN>INSTALL<RET>

!After ACTGEN finishes and returns its prompt, give the
!EXIT command

ACTGEN>EXIT<RET>
$
```



```
$TYPE SYSTEM:7-CONFIG.CMD<RET>

!Terminal Speeds
TERMINAL 1 SPEED 2400
TERMINAL 2 SPEED 9600
TERMINAL 3 SPEED 2400
TERMINAL 4 SPEED 2400
TERMINAL 5 SPEED 300
TERMINAL 6 SPEED 300
DEFINE SYS: MAIN:<SUBSYS>
DEFINE SYSTEM: MAIN:<SYSTEM>
DEFINE NEW: MAIN:<NEW>,SYS:
DEFINE OLD: MAIN:<OLD>,SYS:
DEFINE HLP: MAIN:<OLD>,SYS:
MAGTAPE 0 40672 TU72
MAGTAPE 1 37719 TU72
PRINTER 0 VFU SYS:NORMAL.VFU
PRINTER 0 LOWERCASE RAM SYS:LP96.RAM
TIMEZONE 6

!Commands for the class scheduler

CREATE 0 .05
CREATE 1 .45
CREATE 2 .25
CREATE 3 .15
BATCH-CLASS 3
ENABLE CLASS-SCHEDULING ACCOUNTS ALLOCATED

$
```

To start the Class Scheduler, either reload the system, or give the ENABLE CLASS-SCHEDULER command to OPR. If you edited the n-CONFIG.CMD file to remove the DISABLE ACCOUNT VALIDATION command, you must reload the system so you can start validating accounts.

10.1.9 An Alternative to Using Accounts

Chapter 11 describes the access control program (policy program) that is used to grant and restrict access to various system hardware and software. This same program also includes the appropriate monitor calls to handle class scheduler decisions. Among the requests it can define are classes at login.

NOTE

You CANNOT run the access control program to implement class scheduler decisions at the same time you use the accounting method. You must use one or the other.

10.2 SCHEDULING LOW PRIORITY TO BATCH JOBS

The decision to favor batch jobs or to run them as background tasks depends on the type of batch environment you have at your installation. For example, if users submit batch jobs that are long and/or that do not require completion immediately, you can give batch jobs a low priority. Conversely, if batch jobs are the primary jobs on the system, you can give them a high priority.

Section 10.1 describes how to place batch jobs in either a high or low percentage class by including the BATCH n command in the n-CONFIG.CMD file. When a user submits a batch job, the scheduler uses the batch class and not the user's class.

You must use the class scheduler to give batch jobs a high priority. If you are not using the class scheduler, you can give batch jobs a low priority. Do this in one of two ways:

- Enter the BATCH-BACKGROUND command in the n-CONFIG.CMD file. This command specifies that all batch jobs run on the lowest priority queue, also known as the background queue. This means that after processing all interactive jobs, the scheduler selects and runs batch jobs waiting in the queue. They receive left-over CPU time. You can enter the BATCH-BACKGROUND command into the n-CONFIG.CMD file. The command takes effect the next time you reload the system.

  NOTE

  The BATCH-BACKGROUND command is intended for those who are not using the class scheduler on their system but want to give the batch jobs a low priority. You should not use this command when you enable the class scheduler.

- The operator can issue the SET SCHEDULER BATCH-CLASS n command at the OPR> prompt, where n can be BACKGROUND (batch jobs are run in the lowest priority queue or NONE (batch jobs receive no CPU time).

10.3 FAVORING INTERACTIVE VERSUS COMPUTE-BOUND PROGRAMS

This section describes how you can influence the scheduler to favor either interactive or compute-bound programs. You do this by using the bias control, which is analogous to turning a knob over a range of settings (from 1 to 20). When you select a lower number, the scheduler favors users running interactive programs. When you select a higher number, the scheduler favors users running computational programs.

NOTE

You can use bias control with or without the class scheduler enabled. However, the effect of the bias control is less noticeable when used with the class scheduler.

After you install TOPS-20 software, the system uses the default bias control number 11. This setting distributes the scheduler's attention evenly to interactive and compute-bound programs.

New users of TOPS-20 can use the default setting until they need to favor particular types of programs. Previous users of TOPS-20 may want to experiment with new control settings. For example, the bias controls can serve as a good tool for favoring different types of users at different times of the day. For instance, you can set the bias to a low number during the day to favor on-line users and to a high number during the evening to favor batch users. The response you receive from your user community should determine the appropriateness of the selected settings.

Setting the bias toward interactive programs gives better response to the terminal user. Also, this setting may produce a higher system overhead, because the scheduler swaps jobs and switches from different tasks more often in its effort to favor interactive programs. Generally, setting the bias toward interactive programs is beneficial only if you have sufficient memory. Both the swapping rate and the scheduler overhead are likely to increase in small systems with too little memory. If your system has adequate memory, the scheduler overhead should be fairly constant over most of the bias settings.

Setting the bias toward computational programs should reduce system overhead and increase the total system throughput. However, the response to the terminal user may decrease slightly. In some cases, the improvement in system throughput more than compensates for the lessened response time, and users are more satisfied.

As you experiment with the bias settings, remember that setting the bias control to the extremes can prevent certain types of programs from running for long time periods.

After you select a bias control number that fits your system's operation, enter the BIAS CONTROL command in the n-CONFIG.CMD file or use the SET SCHEDULER BIAS-CONTROL command to the OPR program.

The format of the command in the n–CONFIG.CMD file is:

```
BIAS CONTROL m
```

The format of the command to OPR is:

```
SET SCHEDULER BIAS–CONTROL (TO) m
```

where m is the bias control number. You can change the bias setting during timesharing by using the SET SCHEDULER BIAS–CONTROL command. However, when you reload the system, the bias setting in the n–CONFIG.CMD file takes effect. If you want your change to be permanent, edit the n–CONFIG.CMD file at a convenient time before you reload the system.

Any time you are unsure of the current setting of the bias control, give the INFORMATION (ABOUT) SYSTEM–STATUS command to determine its setting.

10.4 IMPROVING PROGRAM STARTUP TIME

Some of the programs on your system are run frequently. This involves constant searching for the same file on disk, bringing the program pages into memory, and allocating swapping space when the program is swapped out of memory. By storing these programs in an easy–to–access area, some of the startup time is saved. To improve the startup time of the frequently used system programs, TOPS–20 keeps a program name cache.

The <SYSTEM>PROGRAM–NAME–CACHE.TXT file is placed in the directory <SYSTEM> on the system structure automatically at installation time, and contains a list of programs and files to be copied into the program name cache. These are:

```
SYS:PA1050.EXE
SYS:MACRO.EXE
SYS:EDIT.EXE
SYS:TV.EXE
SYS:LINK.EXE
SYSTEM:ERRMES.BIN
```

Each time you reload the system, the <SUBSYS>MAPPER.EXE program runs under the SYSJOB program at the operator's console. <SUBSYS>MAPPER.EXE reads the <SYSTEM>PROGRAM–NAME–CACHE.TXT file and loads the program name cache. Now, when requests are made for these programs, the system looks first in the program name cache to see if it can retrieve the required pages quickly.

To further improve system performance, you can edit the PROGRAM–NAME–CACHE.TXT file and add the filenames of your own frequently accessed programs. For example, if your system uses the FORTRAN language, you may want to add the files:

```
SYS:FORTRA.EXE
SYS:FOROTS.EXE
SYS:FORLIB.REL
```

Your list can contain the names of up to 16 executable files. Therefore, select the files to be placed in the program name cache carefully. You should consider only the executable files that are started frequently by a large number of users. You can also add library files to the program name cache, for example, SYS:FORLIB.REL. However, these types of files use up swapping space. If you have too many or very large files, you may create a detrimental effect on your system's performance. The total library file pages that you cache should be no greater than 200 to 300. Give the VDIRECTORY command for the library files you want to cache and check the number of pages in each file.

If you edit the cache file, the revised file takes effect (MAPPER creates a new version of the cache) the next time you reload the system. Alternatively, you can create a new version of the cache immediately by entering the following commands at the operator's console.

```
^ESPEAK                     !talk to SYSJOB
KILL MAPPER                 !kill old version of cache
RUN SYS:MAPPER.EXE          !read new file and create
                            !new version of cache
^Z                          !exit
```

10.5 REINITIALIZING DISK PACKS

After many files are created, they may no longer be contiguous on the structure (disk(s)). This scattering of files may increase the time it takes to process them. You may decrease processing time by reinitializing the disk packs in your heavily used structures. For example, the system structure may be a likely candidate for reinitialization. This procedure places files into a contiguous format and can be scheduled as part of a backup procedure. How often you reinitialize your packs depends on the work load of the system and whether you notice a difference in system performance after following this procedure.

Reinitializing disk packs requires that you dump all the directories and files to tape, re-create the structure (and, in the case of the system structure, reinstall the system), and restore all the directories and files. Therefore, if you do not notice any appreciable difference in your system performance after doing this, don't schedule it on a regular basis, if at all.

The TOPS-20 Operator's Guide describes re-creating the system structure and other structures. Have the operator use this procedure for reinitializing disk packs. If possible, have the operator re-create the structure and restore the files to a different pack (or set of packs) from the structure that you dumped. This ensures that you do not lose your files should you have problems reading the tape back to disk. That is, you still have the original structure intact and can run DUMPER again to copy the files to another tape.

10.6 DYNAMIC DUAL PORTING

Dynamic dual porting refers to a disk drive that is dual-ported to one system only. When one of the channels is busy transferring data for another disk unit, input/output is automatically switched through the other disk channel. Dynamic dual porting improves the performance of input and output operations. It is activated automatically after field service properly sets up the RP06 or RP07 disk drive(s), and the operator places the drives' port switches in the A/B position.

Dynamic dual porting is not supported for RP20 disks. If it is tried, only one path to the RP20 is used.

The DECSYSTEM-20 Technical Summary describes the hardware associated with input and output operations.

CHAPTER 11

ACCESS CONTROLS

TOPS−20 provides control mechanisms that help you:

- Govern the access to many of your system's resources and services
- Reduce or prevent unauthorized access to the system
- Provide an audit trail to help investigate occurrences of unauthorized access

The following sections discuss these topics.

11.1 ACCESS CONTROL PROGRAM

Previous chapters deal with administrative policies for allocating resources. For example, Chapter 10 describes the policy decisions you can make regarding the scheduler, and Chapter 8 describes the policy decisions you can make regarding tape drive allocation and labeled tape support.

In addition, you can make policy decisions that govern the access to specific system resources. For instance, TOPS−20 allows a user to change the speed of a terminal, assign a device, enable capabilities at any time of day, mount a magnetic tape, and mount a disk structure. However, you may want to restrict or disallow use of some of these facilities. You may want only specified users at specified times of the day and, perhaps, at specified terminals, to use certain facilities. A particular mechanism lets you control the access to such resources and services. With it, you have an additional means for collecting accounting or other information.

The following sections describe how to use this access control mechanism through the TOPS−20 access control program (ACJ). This program carries out your policy decisions in many areas. It can control scheduling classes, the bias control, batch background queue, logging in, use of physical resources (tape drives, terminals, structures), and enabling capabilities. When a user requests a resource (like ASSIGN TTY34:), your program identifies the user, the user's controlling terminal, and the type of request being made. The program can merely log this information in a file, or make a decision and tell the monitor to either grant or deny the request.

11.1.1 Starting the ACJ

You can start the ACJ in three ways:

1. $^ESET SYSTEM−ACCESS−CONTROL−JOB

   The advantage of starting the ACJ with ^ESET is that there is no corresponding ^ESET NO command to disable the ACJ. Starting the ACJ this way or by way of the configuration file is the most secure.

2. ENABLE SYSTEM−ACCESS−CONTROL−JOB command in the configuration file

3. $ESPEAK command:

   ```
   $^ESPEAK
   [Please type SYSJOB commands − end with ^Z]
   RUN SYSTEM:ACJ.EXE
   ^Z
   ```

   The disadvantage of using the ^ESPEAK command is that privileged users can disable the ACJ:

   ```
   $^ESPEAK
   [Please type SYSJOB commands − end with ^Z]
   KILL ACJ
   ^Z
   ```

11.1.2 Defining the ACJ Environment

To tailor the ACJ environment, run the ACJDEC program:

```
@RUN ACJDEC
ACJDEC>HELP
ACJ 7(105) commands:
DISABLE (function) ALL|name
ENABLE (function) ALL|name [profile]
HELP (message)
SAVE (program in) ACJ.EXE
SET (mode) keywords
SHOW ALL|FUNCTION [f]|SETTING [s]|USER [u]
TAKE (commands from) ACJPROFILE.CMD
USER name [profile]
WRITE (commands to) ACJPROFILE.CMD
```

This program WRITEs your tailored settings (according to your ENABLE/DISABLE, SET, and USER commands) into the ACJPROFILE.CMD file. The ACJ executable image is later created with the SAVE command and implements the policy decisions specified in the profile file. Note that the TAKE command resets any of your previous ACJ settings and should be issued as the first command in your ACJDEC session.

The following sections describe the ACJDEC commands.

11.1.3 ENABLE and DISABLE Commands

The ENABLE and DISABLE commands enable the ACJ to receive requests and disable it from receiving them for a particular user function. The command format is:

```
ACJDEC>ENABLE function qualifier
ACJDEC>DISABLE function
```

Where:

- **function** is one of the following:

ACCESS	ARPANET-ACCESS	ASSIGN-DEVICE
ASSIGN-DUE-TO-OPENF	ATTACH-JOB	CAPABILITIES
CLASS-ASSIGNMENT	CLASS-SET-AT-LOGIN	CREATE-DIRECTORY
CREATE-FORK	CREATE-JOB	CREATE-LOGICAL-NAME
CTERM	DECNET-ACCESS	DETACH
ENQ-QUOTA	GETAB	HSYS
INFO	LATOP	LOGIN
LOGOUT	MDDT	MTA-ACCESS
SECURE-CHFDB	SECURE-DELF	SECURE-OPENF
SECURE-RNAMF	SYSGT	TERMINAL-SPEED
TLINK	TTMSG	USER-TEST

NOTE

It is recommended that you enable the following functions:

ASSIGN-DEVICE	LOGIN
ATTACH-JOB	LOGOUT
CAPABILITIES	MDDT
CREATE-DIRECTORY	SMON

Also, if your site is using "secure" files (see Section 11.7), enable all the SECURE functions.

Section 11.1.3.1 describes the ENABLE/DISABLE command functions.

- **qualifier** is one of the following:

[NO] CONSOLE	[NO] DENY-BATCH	[NO] DENY-CTY
[NO] DENY-DECNET	[NO] DENY-DETACHED	[NO] DENY-LAT
[NO] DENY-LOCAL	[NO] DENY-PTY	[NO] DENY-TCP
[NO] LOG	[NO] POLICY	

NOTE

The following qualifiers are frequently used:

[NO]LOG [NO]POLICY

The defaults are LOG and POLICY.

Section 11.1.3.2 describes the ENABLE/DISABLE command function qualifiers.

11.1.3.1 ENABLE/DISABLE Command Functions –

NOTE

You can specify ALL to enable or disable all functions.

- ENABLE ACCESS

  This function controls the ACCESS, CONNECT, and END-ACCESS user commands. It is recommended that this function be enabled with the POLICY and LOG qualifiers so that an audit trail of failed accesses and user connections to their "owned" subdirectories is created. (Owned subdirectories are <username*> directories on any file structure that is DOMESTIC.) Directory owners are always allowed to connect to their "owned" subdirectories.

- ENABLE ARPANET-ACCESS

  This function controls access to TCP/IP. The ACJ is activated each time a user tries to initiate an outgoing TCP/IP connection. The implemented policy allows access for users with SC%ANA capability only. To log all TCP/IP access and allow all users TCP/IP access, enable this function with NO POLICY.

- ENABLE ASSIGN-DEVICE and ENABLE ASSIGN-DUE-TO-OPENF

  These identical functions help prevent two users on separate systems from accessing the same tape drive at the same time, if the two systems are not in the same CFS-20 cluster. This is important when there are shared tape drives between two or more systems.

  Only users with enabled WHEEL or OPERATOR privileges are allowed to assign magtape devices. The functions prevent non-enabled users from assigning magtape devices that are set UNAVAILABLE with the OPR command. All other devices are allowed.

  It is recommended that you enable this function with the POLICY and NOLOG qualifiers.

- ENABLE ATTACH-JOB

  Like LOGIN, the ATTACH function controls user access to the system. The attach is denied if:

  The job is a batch job.

  The target job's user profile indicates that the target user cannot log in to the source's line type.

  WHEEL-only logins are set for the source line type and the target job is not a WHEEL user.

  The target job is a batch job and the source job is not an enabled WHEEL.

  The source job has OPERATOR capabilities enabled and the target job is a user with WHEEL.

  The source job is controlled by a user with OPERATOR capability and the target job is a user with WHEEL capabilities.

  It is recommended that you enable the ATTACH function with the LOG and POLICY qualifiers.

- ENABLE CAPABILITIES

  This function controls the enabling of the capabilities described in Section 5.9.

  To allow all users to enable at any time and to log all capability changes, enable the CAPABILITIES function with the NO POLICY qualifier. Issue the DISABLE CAPABILITIES command to allow enabling at any time with no logging of activity.

  Setting WHEEL or OPERATOR capabilities is disallowed during non-prime time unless you have issued the USER ENABLE-NON-PRIME-TIME command (described in Section 11.1.4).

  If you use the USER ENABLE-NON-PRIME-TIME command (described in Section 11.1.4), it is recommended that you also use the LOG and POLICY qualifiers.

  The CAPABILITIES function provides an audit trail of all users who enable capabilities.

- ENABLE CLASS-ASSIGNMENT

  The ACJ is activated for all attempts to set a job's scheduler class using the SKED% monitor call. An example of this is the OPR command SET JOB x SCHEDULER-CLASS y. However, if a user is not WHEEL or OPERATOR and is trying to set another job's class, the ACJ is not consulted and the user receives a CAPX1 error.

- ENABLE CLASS-SET-AT-LOGIN

  The ACJ is activated each time a job logs in only if the class scheduler is on and class assignments are set by the policy program.

  The policy implemented is to use a SKED% monitor call to set the job in a particular class as specified by the user profile CLASS-AT-LOGIN (see Section 11.1.4). Because class zero is the default class, no SKED% monitor call is done if the CLASS-AT-LOGIN in the user profile is set to zero or if there is no user profile associated with the user logging in. If the job cannot be set in the proper class, the ACJ log file entry is marked as "unusual."

- ENABLE CREATE-DIRECTORY

  This function prevents unauthorized manipulation of directories. It provides an audit trail of changes to directories and attempted break-ins since the last new username was created or a capability was changed.

  This function has the following effects:

  Users with OPERATOR capabilities are not allowed to make new usernames or changes to existing usernames on DOMESTIC structures. With TOPS-20, the boot structure and the public structure are always DOMESTIC and cannot be made FOREIGN.

  Setting passwords or user or directory groups on any <ROOT-DIRECTORY> is not allowed.

  Only enabled WHEELs can change the following directory attributes:

  SECURE
  FILES-ONLY
  WHEEL
  OPERATOR or SEMI-OPERATOR

  Nonprivileged users must create subdirectories with the FILES-ONLY and NO SECURE attributes.

  It is recommended that you enable the CREATE-DIRECTORY function with the LOG and POLICY qualifiers.

- ENABLE CREATE-FORK

  This function controls the ability to create job forks. Each time a user runs a program, a process (fork) is created. The CFORK% monitor call is used to create forks. The ACJ is activated for each CFORK% monitor call that creates more than FKCNT forks. (FKCNT is a monitor cell that system programmers can patch; it is defaulted to 5.) The ACJ always allows the CFORK% monitor call.

- ENABLE CREATE-JOB

  This function controls access to the CRJOB monitor call. The ACJ allows only enabled WHEELs or OPERATORs to perform the CRJOB%. To log all CRJOB-created jobs but let any user use CRJOB, enable this function with NO POLICY.

- ENABLE CREATE-LOGICAL-NAME

  This function controls the ability to create systemwide logical names. The ACJ is activated for each of the following CRLNM% monitor call functions when the user is not a WHEEL or OPERATOR: CLNS1, .CLNSA, or .CLNSY. The LOG qualifier provides an audit trail of logical name activity.

- ENABLE CTERM

  This function enables incoming CTERM connections by way of the SET HOST command. It also creates an audit trail, which provides the origin of the connecting user. (The @SYSTAT command displays user origins.)

  It is recommended that you enable this function with the LOG qualifier.

- ENABLE DECNET-ACCESS

  This function allows access to the network for users with DECNET-ACCESS capability.

  It is recommended that you enable this function with the LOG qualifier to provide an audit trail of users who are accessing other systems through DECnet connections.

- ENABLE DETACH

  This function controls the ability to detach jobs by way of the DETACH command. The ACJ always allows users to detach jobs. The LOG qualifier provides an audit trail.

- ENABLE ENQ-QUOTA

  This function controls user setting of ENQ quota. The ACJ allows only a WHEEL or OPERATOR to perform the function.

- ENABLE GET-DIRECTORY

  The GTDIR% monitor call provides information about a directory.

  The ACJ always allows the GTDIR% monitor call to succeed. This function is primarily to assure an audit trail whenever a user gets information about a directory (INFORMATION DIRECTORY command).

- ENABLE GETAB

  This function controls use of the GETAB% monitor call (SYSTAT and INFORMATION MONITOR commands) to get information from the monitor. The ACJ is activated for each GETAB% when the previous context is not monitor. The ACJ always allows the GETAB%.

  You should enable this function with caution, because it increases system overhead and slows system response, especially when the activity is logged.

- ENABLE HSYS

  This function allows only users with enabled WHEEL, OPERATOR, or MAINTENANCE privileges to shut down the system with the ^ECEASE command.

- ENABLE INFO

  This function controls access to the INFO% monitor call to get information about systems in a CFS-20 cluster (SYSTAT NODE command) or to send clusterwide notices.

  You should enable this function with caution, because it increases system overhead and slows system response, especially when the activity is logged.

- ENABLE LATOP

  This function allows enabled wheels and operators to implement .LARHC (request host connect) functions in the LATOP% monitor call. To log all LATOP% .LARHC functions and allow all users access to LATOP%'s .LARHC functions, enable this function with NO POLICY. These functions are performed regularly by LPTSPL for LAT printers.

- ENABLE LOGIN

  The LOGIN function controls access to the system. A login request is denied if:

  The user tries to log in to <ROOT-DIRECTORY>.

  The user is over quota on the ps:<username> directory.

  You specified with the USER command (described in Section 11.1.4) that the user cannot log in to this terminal line type.

  WHEEL-only logins are set and the user does not have WHEEL capability.

The login attempt is on a non−batch PTY.

The user has OPERATOR capability and is trying to log in to a directory with WHEEL capability.

It is important to enable the LOGIN function so that there is an audit trail for tracking "unusual" logins.

It is recommended that you enable the LOGIN function with the LOG and POLICY qualifiers.

- ENABLE LOGOUT

  The ACJ is consulted each time a user attempts to log out. The logout request is denied if the user is over quota on the ps:<username> directory.

  The LOGOUT function helps capture a user session in an audit trail by showing when the user logged out.

  It is recommended that you enable the LOGOUT function with the LOG and POLICY qualifiers.

- ENABLE MDDT

  This function allows access to internal monitor data structures.

  All non−WHEEL users are prevented from entering MDDT, because through MDDT, users can violate system security or crash the system. It is important to enable this function with the LOG and POLICY qualifiers so that an audit trail of MDDT entry is available in case of security problems.

- ENABLE MTA−ACCESS

  This function controls access to magnetic tapes. The ACJ is activated for labeled MTA access in the following situations:

  - The label type is TOPS−20 and access is by non−owner and there is a protection failure.
  - ANSI labels are used and volume accessibility is not "full."
  - EBCDIC labels are used and the accessibility byte is from 1 to 3 inclusive.

  The ACJ always allows the access.

- ENABLE SECURE

  Four ENABLE/DISABLE command functions control access to "secure" files. These functions are described below. Refer to Section 11.7 for details on secure files.

  ENABLE SECURE CHFDB% allows a request to set or clear a file SECURE, based on settings in the ACCESS.CONTROL file. If there is no ACCESS.CONTROL file in the same directory as the file to be set secure, the request is allowed and logged as "unusual."

  ENABLE SECURE DELF% allows the request to delete a secure file based on settings in the ACCESS.CONTROL file. If there is no ACCESS.CONTROL file in the same directory as the file to be deleted, the request is allowed and logged as "unusual."

  SECURE OPENF% allows the request to open a secure file based on settings in the ACCESS.CONTROL file. If a read, write, or append access is attempted to a secure file and no ACCESS.CONTROL file is in the same directory as the secure file, the access is allowed and logged as "unusual."

  SECURE RNAMF% allows the request to rename a secure file based on settings in the ACCESS.CONTROL file. If rename access is attempted to a secure file and there is no ACCESS.CONTROL file in the same directory, the access is allowed and logged as "unusual."

- ENABLE SET−TIME

  The STAD% monitor call sets the system time. The ACJ always allows the monitor call to succeed. This function is primarily to assure an audit trail whenever the system date and time are changed.

- ENABLE SMON

  The SMON% monitor call sets or clears operating system features (such as account validation, logins allowed,...) and is usually invoked with commands in SYSTEM:7−CONFIG and the ^ESET command. The ACJ is activated for each SMON%.

  The ACJ allows only enabled WHEELs or OPERATORs to use the monitor call.

  It is recommended that you enable the function with the LOG and POLICY qualifiers, because SMON% changes TOPS−20 operating system parameters.

- ENABLE STRUCTURE−MOUNT

  This function controls mounting of disk structures. The ACJ is activated for every MSTR% monitor call "increment mount count" function (.MSIMC function). A job must have a "regulated" file structure mounted in order to use it.

  The ACJ always allows the increment of the mount count and lets normal file and directory protection handle security.

- ENABLE SYSGT

  The SYSGT% monitor call extracts information from the monitor (SYSTAT and INFORMATION MONITOR commands). The ACJ is activated for each SYSGT% when the previous context is not monitor. The ACJ always allows the SYSGT%.

  You should enable this function with caution, because it increases system overhead and slows system response, especially when the activity is logged.

- ENABLE TERMINAL−SPEED

  This function disallows the setting of terminal baud rate with the SET TERMINAL command unless the WHEEL or OPERATOR privilege is enabled.

- ENABLE TLINK

  The ACJ is activated when the following commands are invoked from outside the monitor: ADVISE, TALK, BREAK, and REFUSE. The ACJ always allows these functions; however users can override them with the REFUSE commands.

  If the user is set SPY−ON (see Section 11.1.4), the log entry will be marked as "unusual."

- ENABLE TTMSG

  The ACJ is activated when the ^ESEND and SEND commands are invoked from outside the monitor. The ACJ allows only enabled WHEELs or OPERATORs to use these commands and the associated monitor call (TTMSG%).

- ENABLE USER−TEST

  This function is reserved for customer use. System programmers should refer to function code 400000 of the GETOK monitor call.

11.1.3.2 ENABLE/DISABLE Function Qualifiers −

The following are qualifiers that you can specify with the ENABLE and DISABLE commands described in Section 11.1.3.1.

- [NO]POLICY

  Causes the enforcement of the ACJ policy on a particular ENABLE function. The default is POLICY. The NO POLICY qualifier is often combined with the LOG qualifier; without this qualifier, the effect is similar to the DISABLE function.

- [NO]LOG

  Creates an entry in the ACJ log file that describes the function. The default is LOG.

- [NO] CONSOLE

  The CONSOLE keyword enables display of the logging string to the console terminal. The default is NO CONSOLE.

- [NO] DENY−BATCH

  The DENY−BATCH keyword causes a request for this function from any batch job to always be denied. The default is NO DENY−BATCH.

- [NO] DENY−CTY

  The DENY−CTY keyword causes a request by a job logged into the system’s console terminal (the CTY) to always be denied. The default is NO DENY−CTY.

- [NO] DENY−DECNET

  The DENY−DECNET keyword causes a request by a job which is attached to the system through DECnet (NRT or CTERM protocol) to always be denied. The default is NO DENY−DECNET.

- [NO] DENY−DETACHED

  The DENY−DETACHED keyword causes a request by a detached job to always be denied. The default is NO DENY−DETACHED.

- [NO] DENY−LAT

  The DENY−LAT keyword causes a request by a job attached to the system through a LAT terminal to always be denied. The default is NO DENY−LAT.

- [NO] DENY–LOCAL

  The DENY–LOCAL keyword causes a request by a job logged into any terminal local to the system to always be denied. The default is NO DENY–LOCAL.

- [NO] DENY–PTY

  The DENY–PTY keyword causes a request by a job attached to a PTY that is not a batch job to always be denied. The default is NO DENY–PTY.

- [NO] DENY–TCP

  The DENY–TCP keyword causes a request by a job attached to the system through TCP/IP (TELNET protocol) to always be denied. The default is NO DENY–TCP.

11.1.4 USER Command

You can make ACJ profile entries for users:

```
ACJDEC>USER username keyword
```

Where:

- username is a particular user or all users (*)
- keyword is one of the following:

CLASS–AT–LOGIN n

Sets scheduler class "n" at login. The default is CLASS–AT–LOGIN 0

[NO] ENABLE–NON–PRIME–TIME

Lets users enable capabilities at times other than prime time (see Section 11.1.5). If you want to let anyone enable at any time, enable the CAPABILITIES function with the NO POLICY qualifier, or issue the DISABLE CAPABILITIES command. The default is NO ENABLE–NON–PRIME–TIME.

[NO] LOGIN–BATCH

Lets batch jobs log in. The default is NO LOGIN–BATCH. Note that the LOGIN–PTY keyword controls non–batch PTY logins.

[NO] LOGIN–CTY

Lets a job log in to the system through the system's console terminal (the CTY). The default is LOGIN–CTY.

[NO] LOGIN–DECNET

Lets a job log in to the system through DECnet using the NRT or CTERM protocol. Do not confuse this keyword with remote DECnet access using the RMSFAL (which is controlled by the LOGIN–DETACHED keyword). The default is LOGIN–DECNET.

[NO] LOGIN–DETACHED

Lets a job log in detached. Digital–supplied software that does detached logins is limited to DIU (Data Interchange Utility) and RMSFAL (RMS File Access Listener).

[NO] LOGIN–LAT

Lets a job log in to the system through a LAT terminal.

[NO] LOGIN–LOCAL

Lets a job log in to terminal lines connected to the DECSYSTEM–20 (RSX20F console front end) that are not configured as REMOTE. LOCAL lines are normally connected directly to terminals rather than to communications equipment such as modems or port selecters.

[NO] LOGIN–PTY

Lets a job log in to a PTY that is not a batch job.

[NO] LOGIN–REMOTE

Lets a job log in to terminal lines connected to the DECSYSTEM–20 (RSX20F console front end) that are configured as REMOTE. REMOTE lines are normally connected to modems or port–selecting equipment and not directly to terminals.

[NO] LOGIN–TCP

Lets a job log in to the system through a TCP/IP terminal (TELNET protocol).

[NO] SPY–ON

Users will be spied on when logging in or attaching to jobs. The default is NO SPY–ON.

EXAMPLE

USER command keywords are normally combined. For example, if a user should be able to log in only from a batch job and for DECnet access using RMSFAL, you should set the keywords for that user as follows:

```
NO LOGIN-CTY
NO LOGIN-DECNET
NO LOGIN-LAT
NO LOGIN-LOCAL
NO LOGIN-PTY NO
LOGIN-REMOTE
NO LOGIN-TCP
```

In the example above, only LOGIN-BATCH and LOGIN-DETACHED are allowed.

11.1.5 SET Command

You can tailor the ACJ environment with the SET command:

```
ACJDEC>SET keyword
```

Where keyword is one of the following:

ACCESS-LOG-FILE	PRIME-TIME-BEGIN
PRIME-TIME-END	SPY-CHECK-INTERVAL
SPY-LOG-DIRECTORY	LOG-FILE-CACHE-SWEEP-INTERVAL

The SET command keywords are described below:

ACCESS-LOG-FILE filespec

Sets the log file specification. The default is SYSTEM:LOGFILE.LOG.

LOG-FILE-CACHE-SWEEP-INTERVAL n

Sets the log file cache sweep interval. The log file is updated every few seconds (or whenever it is about to be read or renamed). The log file cache sweep interval defaults to 30 seconds. The cache can be disabled by setting the log file cache sweep interval to zero; this forces the log file to be updated each time a line is written to it.

PRIME-TIME-BEGIN time

Sets the time at which prime time begins on each weekday. This time is used when you issue the USER ENABLE-PRIME-TIME command (described in Section 11.1.4) with the ENABLE CAPABILITIES command. The default beginning time is 07:00 (7 AM).

PRIME-TIME-END time

Sets the time at which prime time ends on each weekday. This time is used when you issue the USER ENABLE-PRIME-TIME command with the ENABLE CAPABILITIES command. The default ending time is 18:00 (6 PM).

SPY-CHECK-INTERVAL seconds

Sets the time between TLINK% monitor calls to jobs that are being spied on. Shorter times increase overhead but lower the risk that spying data will be lost. The default is 10 seconds.

SPY-LOG-DIRECTORY string

Sets the initial part of the specification for spy log files. The string will have "-nodename.username" appended to it to complete the file specification. The default is SYSTEM:ACJ-SPY, so spy log files are of the form "SYSTEM:ACJ-SPY-nodename.username."

The ACJ always makes spy log files secure.

11.1.6 SHOW Command

The SHOW command displays the items changed with the ENABLE, DISABLE, SET, and USER commands. The SHOW command is followed by one of the following keywords:

- ALL

  Shows all information.

- FUNCTION ALL keyword

  Shows the profile of all functions or of a particular function.

- SETTINGS ALL keyword

  Shows all items or a particular item changed by the SET command.

- USER ALL * or username

  Shows all user profiles or a particular user profile.

11.1.7 WRITE Command

When you have finished issuing ACJDEC commands, write the ACJ settings to the profile file:

```
ACJDEC>WRITE filename
```

Where filename is the name of the profile file. The default filename is ACJPROFILE.CMD.

The following is a sample profile file:

```
Set PRIME-TIME-BEGIN 07:30
Set PRIME-TIME-END 18:00
Enable ACCESS
Enable ARPANET-ACCESS NO POLICY
Enable ASSIGN-DEVICE NO LOG
Enable ASSIGN-DUE-TO-OPENF NO LOG
Enable ATTACH-JOB
Enable CAPABILITIES DENY-DECNET DENY-TCP
Enable CREATE-DIRECTORY
Enable CTERM
Enable DECNET-ACCESS NO POLICY
Enable LOGIN
Enable LOGOUT
Enable MDDT DENY-BATCH DENY-DETACHED
Enable SECURE-CHFDB
Enable SECURE-DELF
Enable SECURE-OPENF
Enable SECURE-RNAMF
User FS ENABLE-NON-PRIME-TIME
```

11.1.8 SAVE Command

When you have written the profile file, create the ACJ:

```
ACJDEC>SAVE filename
```

Where filename is the name of the access control program. The default filename is ACJ.EXE.

11.1.9 Summary

Follow these steps to implement the ACJ:

1. Run the ACJDEC.EXE program:

```
@RUN ACJDEC.EXE

ACJDEC>
```

2. Set up the ACJ and user environments:

```
ACJDEC>TAKE ACJPROFILE.CMD

ACJDEC>Set PRIME-TIME-BEGIN 07:30

ACJDEC>Enable ACCESS

ACJDEC>User FS ENABLE-NON-PRIME-TIME
.
.
.
```

3. Write the commands into the ACJPROFILE.CMD command file:

```
ACJDEC>WRITE
```

4. Create the ACJ.EXE policy program from the current settings:

```
ACJDEC>SAVE SYSTEM:

ACJ.EXE.1 Saved
```

5. Start the ACJ policy program:

```
$^ESET SYSTEM-ACCESS-CONTROL-JOB
```

11.1.10 Reviewing the Log Files

You should review daily the log files produced by the ACJ for possible unusual activity. Careful monitoring results in greater system security, because repeat penetrations cause the most severe security problems. The log file is always made "secure" (see Section 11.7) to prevent unauthorized access. A new generation of the log file is created each time the ACJ is run and each night at midnight.

11.1.10.1 Log File Format –

The log file is split into pages with a header on each page. The first line of the header contains the ACJ version, system nodename, current date/time, and page number. The second and third lines summarize the activity so far: number of requests allowed, denied, and failed; CPU time used; and "uptime" of the ACJ.

Each activity results in one line of text in the log file. The first item is the time of day. The second item is the username requesting the operator. (For the LOGIN function, the username that is trying to log in is in this field.) The next field is the access control function. Following the function, there is information about the job. This information consists of the job number, controlling job if the job is being run on a PTY, "batch" if it is a batch job, terminal number, network origin string if the job originates from a network terminal, and the job's current program name. Finally, the enabled capabilities of the process attempting access are displayed.

Following that fixed information is a comma and information that is specific to this request type. For example, if a user is enabling capabilities, the desired capabilities are shown.

At the end of the line one of three strings may appear.

- [Denied] indicates that the ACJ denied the request. For example, a user tried to log in but is not allowed to log in to that terminal type.
- [Failed] indicates that the ACJ allowed the request but that the action failed. For example, a user tried to log in but gave a bad password.
- [Unusual] indicates that the ACJ allowed the request but it is in some way unusual. For example, a user logged in when that user is set SPY–ON. The conditions under which the request is considered unusual are documented in Section 11.1.3, ENABLE and DISABLE commands.

11.1.10.2 Log File Examples –

This section shows sample entries from a typical ACJ log file.

The following is a sample ACJ log file header. It shows that the log file was started at midnight and that the ACJ has been up about 12 hours, processing 790 requests.

```
     ACJ 7(100) on GARK, Thursday, February 2, 1989 00:00:00, page 1
     Allowed 782 requests, denied 8 requests, 5 requests failed
     Used 1:09.32 in 11:57:56.69
```

The following log file entries show a batch job execution. User OPERATOR running BATCON opened a PTY, assigned it to itself, and created a job, which logged in to user SCHMITT. That user enabled capabilities, and then the batch job ran to completion. The job running BATCON then logged out the batch job.

```
     00:00:47 OPERATOR Open–assign job 194 ctrl 193 TTY233 GALAXY opr
     ana, device PTY7
     00:00:47 OPERATOR Assign job 194 ctrl 193 TTY233 GALAXY opr ana,
     TTY241
     00:00:48 OPERATOR CRJOB job 194 ctrl 193 TTY233 GALAXY opr ana
     00:00:48 SCHMITT Login job 206 batch TTY241 EXEC, by OPERATOR job
     194 ctrl 193 TTY233 GALAXY
     00:00:49 SCHMITT Caps job 206 batch TTY241 ENABLE, desired whl
     00:01:36 OPERATOR Logout job 194 ctrl 193 TTY233 GALAXY opr ana,
     target SCHMITT job 206 batch TTY241 EXEC
```

The next two entries show an incoming CTERM connection from user SGAGNE on system GIDNEY, who proceeds to log in to this system.

```
     08:35:49 OPERATOR Cterm job 0 Det SYSJOB, from GIDNEY::SGAGNE
     08:35:55 SGAGNE Login job 214 TTY364 GIDNEY::SGAGNE(CTM) LOGIN
```

Below, user GAS tried to attach to his detached job and apparently typed an incorrect password (see [Failed]). On the second try, he succeeded and attached to his job.

```
     09:38:48 not–logged–in Attach job 214 TTY444 LAT1(LAT) ATTACH,
     target GAS job 207 Det DETACH [Failed]
     09:38:58 not–logged–in Attach job 214 TTY444 LAT1(LAT) ATTACH,
     target GAS job 207 Det DETACH
```

Below, the first entry shows the OPERATOR job running MX connecting to system VAXVLN using DECnet. The second entry shows the MX job delivering mail to user APUCHRIK, who has a "secure" MAIL.TXT file. The third entry shows this same user accessing his mail file with MS.

```
09:39:32 OPERATOR DECnet job 198 ctrl 193 TTY237 MX opr ana, to
VAXVLN
09:47:11 OPERATOR Secure-OPENF job 198 ctrl 193 TTY237 MX opr
ana, read write PUBLIC:<APUCHRIK>MAIL.TXT.1
09:49:15 APUCHRIK Secure-OPENF job 199 TTY435 LAT1(LAT) MS, read
write preserve-dates PUBLIC:<APUCHRIK>MAIL.TXT.1
```

The first line below shows a user trying to set TTY3's terminal speed. The request is denied. The second and third lines show OPERATOR jobs running DUMPER to save the system. The second line shows that user RASPUZZI did not allow OPERATOR read access to a file; the access failed and the file is not backed up. The third line shows that a SECURE file was opened for reading, but there was no ACCESS.CONTROL file present in that directory. The request is marked as unusual, and access is allowed.

```
19:17:56 JWONG Terminal-speed job 216 TTY3 EXEC, TTY3 input 2400
output 2400 [Denied]
19:38:00 OPERATOR Secure-OPENF job 215 ctrl 206 TTY243 DUMPER opr
ana, read preserve-dates RANDOM:<RASPUZZI.MAIL>JAN89MAIL.TXT.1
[Denied]
19:38:24 OPERATOR Secure-OPENF job 211 ctrl 206 TTY241 DUMPER opr
ana, read preserve-dates WORK:<DEVANS.POOF>BOBBIE.MAIL.1
[Unusual]
```

11.2 PASSWORD ENCRYPTION

One way to violate system security is through unauthorized use of directory passwords. Having acquired someone's password, an intruder could log in or gain access to restricted system resources. The password encryption facility in TOPS-20, however, makes it harder to steal passwords.

With encryption enabled, passwords entered into the system are translated to an indecipherable cyphertext format before they are stored or otherwise used. Nowhere in the system is the original plaintext form of a password kept. As a further security measure, no current TOPS-20 utility converts the cyphertext to plaintext.

NOTE

Password encryption is irreversible. Therefore, before enabling encryption, be sure you will never need to revert back to an earlier version of the operating system.

To enable password encryption, use the CHECKD program. You can do this during or after system installation. (Refer to the TOPS-20 KL Model B Installation Guide for details.) With CHECKD, password encryption is enabled on a structure-by-structure basis; after the procedure, all passwords for a particular structure are encrypted as previously described. If you enable encryption after installation, run the KRYPTN program after CHECKD to convert existing plaintext passwords on a structure to cyphertext. The KRYPTN program is located on the tools tape, which is part of the TOPS-20 software installation package.

Encryption should be enabled for all structures except those that will be used on a TOPS-20 pre-Version 6 system. (Section 11.2.1 discusses this topic.)

You can add your own encryption algorithm to the system if you choose not to use TOPS-20's algorithm. Refer to Section 11.2.2.

Because the encryption algorithm is irreversible, care is required in the following areas:

- Remembering one's password
- Working in a multiple-system environment
- Adding new algorithms to the system
- Using DUMPER

Mistakes in these areas could invalidate passwords so that they may need to be respecified with BUILD or ^ECREATE. These interrelated topics are discussed in the following sections.

11.2.1 Moving Structures Among Systems

If you are in a multiple-system environment, you may need to move structures from one system to another. Problems could arise, however, if some systems are running TOPS-20 pre-Version 6 software and others are running TOPS-20 Version 6. For example, when a structure containing encrypted passwords is taken to a TOPS-20 pre-Version 6 system, any access to files on the pack that requires a password to be supplied fails, because, in validating a password, the older monitor simply compares the entered plaintext to the cyphertext stored on disk. The older monitor is unfamiliar with the encryption process.

To avoid this problem, you should postpone encryption for relevant structures until all systems are upgraded.

Any TOPS-20 system can correctly handle unencrypted structures.

You could also encounter problems in moving structures to other systems if you use your own encryption algorithm. This topic is discussed below.

11.2.2 Adding Encryption Algorithms to the System

You can use one or more of your own encryption algorithms exclusively or in addition to TOPS-20's algorithm. For a description of the procedures involved, refer to the monitor module, STG.MAC.

Each time a password is encrypted and stored in a directory, the version number of the algorithm used to encrypt it is also stored. This allows new encryption algorithms to be added to the system with no impact on currently encrypted passwords, provided the old algorithms have not been removed from the monitor. Only passwords created since the installation of the new (current) algorithm will be encrypted with that algorithm. Older passwords invoke the appropriate algorithms during password-required accesses.

If you also want existing passwords to use the new algorithm, the operator must individually respecify the passwords with BUILD or ^ECREATE. The operator does this after the new algorithm is installed. Note that KRYPTN cannot be used here to convert existing cyphertext to new cyphertext.

In using your own encryption algorithms, be aware that directories on structures and on DUMPER-created tapes include passwords that may be unusable at other sites. Other TOPS-20 monitors could consider the passwords' algorithm version numbers to be invalid. For example, these monitors may acknowledge only the standard TOPS-20 algorithm. Even if a site accepts the version numbers, its corresponding algorithms may be different from the algorithms at your site. Thus, on attempted password use, the cyphertext produced at this other site could never match the stored cyphertext.

To address these problems, you could:

- Warn sites about the nature of the passwords on a tape or structure. A site could then avoid using certain directories or respecify a password with BUILD or ^ECREATE if necessary.
- Refrain from saving directory information on tapes bound for other sites. That is, do not use DUMPER's CREATE command when creating such tapes.
- Ship only directories that have null passwords. These "passwords" are considered unencrypted, and should cause no problem on any system.

11.2.3 Using DUMPER

Section 11.2.2 addressed using DUMPER with nonstandard algorithms. This section continues the discussion of DUMPER.

Care must be taken when restoring directories that were saved with DUMPER's CREATE command. Incompatible versions of tapes, DUMPER, and TOPS-20, when combined, can produce a number of password-related problems. Note the system's behavior during tape restoration for the combinations in Table 11-1. In the table, tape Version 4 is the version of tape that DUMPER Version 4.1 creates. Tape Version 5 is the version of tape that DUMPER Version 5 creates.

Table 11–1: DUMPER Directory Restorations

Tape Version	DUMPER	MONITOR	Result
5	5	6	OK
4	5	6	OK (N1)
5	4.1	6	E1
4	4.1	6	OK (N1)
5	5	5	E2
4	5	5	OK
5	4.1	5	E3
4	4.1	5	OK

Legend:

N1 Passwords are correctly encrypted for the first time using the monitor's current encryption algorithm.

E1 The tape version number is incompatible with this DUMPER. DUMPER reports this fact before restoring the tape data. If directories are restored from this tape, encrypted passwords are re–encrypted, causing all uses of these passwords to fail. The passwords will then have to be individually respecified with BUILD or ^ECREATE. However, if a password is unencrypted on the tape, then it is encrypted for the first time, and will be usable.

Any files on this tape are restored correctly.

E2 Pre–version 6 monitors have no logic to handle encryption–related data that the tape may contain. Therefore, restored encrypted passwords are unusable and must be respecified with BUILD or ^ECREATE. Note that directory blocks on the tape may contain password descriptor information, such as the encryption version number. This descriptor data is not restored.

Any files on this tape are restored correctly.

E3 The tape version number is incompatible with this DUMPER. DUMPER reports this fact before restoring the tape data. This incompatibility results in the same situation as E2.

If these problems occur often, users and operators could refrain from saving directory information on tapes, or they could use null passwords for directories that are to be saved. Null passwords are considered to be unencrypted and should cause no access problems.

11.3 PASSWORD MANAGEMENT

Encryption is just one part of password management. You should also make sure that users at your site do not choose passwords that are too short or that are otherwise easy to guess, such as one's name or initials.

11.3.1 Setting Password Length

You can set the minimum password length in the n–CONFIG.CMD file with the command:

ENABLE MINIMUM–PASSWORD–LENGTH n

where:

n is the minimum number of characters allowed for passwords, from 1 to 39 characters. A common value for n is 8 characters.

Or, you can issue the command:

$^ESET [NO] MINIMUM–PASSWORD–LENGTH n

The @INFORMATION SYSTEM command reports the minimum password length set for your system.

11.3.2 Changing Passwords Regularly

It would also be helpful for users to change their passwords regularly. You can enforce this through the ^ESET command and in the n–CONFIGURATION file:

$^ESET [NO] PASSWORD–EXPIRATION (TO) N

In the n–CONFIGURATION file:

ENABLE PASSWORD–EXPIRATION n
DISABLE PASSWORD–EXPIRATION

Where:

n is the date after which passwords expire

n is the number of days a password is valid since the time it was last changed. This value for n is a number from 1 through 366. The default number of days is 30. The @INFORMATION SYSTEM command reports the number of days a password is valid.

After a password has expired, the user will be allowed to log in but must change the password immediately.

11.3.3 Disallowing Certain Passwords

You may want to make certain combinations of characters illegal for use as passwords. Perhaps they would be too easily guessed by an intruder. You can place such words in the file SYSTEM:PASSWORD.DICTIONARY and have them automatically matched against newly supplied passwords. If a match occurs, the password is denied, and the user must choose a new one.

The following is a sample password dictionary file. Note that words must be placed in the file alphabetically.

```
ABORT, S, ED, ING
ABUSE, S, D, \ING
ACQUIT, S, "ED, "ING
ADMIRAL, S, 'S
```

The first line shows the word ABORT. Other entries, separated by commas, indicate suffixes that are added to ABORT. So, the first line, in effect, contains the following words that cannot be used as passwords: ABORT, ABORTS, ABORTED, or ABORTING.

In the second line, the backslash character (\) removes the last character from the base word and adds the suffix. The base word is ABUSE. After the final E is removed and ING is added, the word ABUSING is formed and cannot be used as a password.

In the third line, the quote character (") duplicates the last letter of the word before adding the suffix; ACQUIT becomes ACQUITTED.

In the final line, the apostrophe is taken as is. The base word ADMIRAL becomes ADMIRAL'S.

You can enable and disable the dictionary–checking feature through the ^ESET command or in the n–CONFIG.CMD file:

$^ESET [NO] PASSWORD–DICTIONARY

In the n–CONFIG.CMD file:

ENABLE PASSWORD–DICTIONARY
DISABLE PASSWORD–DICTIONARY

The @INFORMATION SYSTEM command reports whether or not the password dictionary is enabled.

11.4 LAST LOGIN INFORMATION

When users log into the system, the dates and times they last logged in are displayed on their terminals. This information helps alert them to instances of illegal system or account entry. For example, if users keep track of their actual login times, they can compare those times to the ones displayed. Then, if there is a discrepancy, they will know the exact time that someone else logged into their directory using their password.

The system also provides login–failure information.

Information is provided for both interactive and noninteractive logins. An example of a noninteractive login is one for a batch job.

11.5 PREVENTING FAST LOGINS

By using the /FAST switch with the LOGIN command, users can bypass processing of the system's and their own LOGIN.CMD and COMAND.CMD files. Managers sometimes set up these command files to limit users' computing environments, however. For example, sets of users may be allowed only to read mail or run some other computer application. You can prevent fast logins at your site by entering the following command in the n-CONFIG.CMD file:

DISABLE FAST-LOGIN-OPTION

The ENABLE FAST-LOGIN-OPTION command is in effect by default. You can also enable and disable fast logins with the ^ESET privileged command.

Refer to the TOPS-20 User's Guide for information on the LOGIN.CMD and COMAND.CMD files. The TOPS-20 Commands Reference Manual describes the LOGIN command.

Also, with fast logins, system mail is not displayed, nor is notice of new mail given.

11.6 PREVENTING NOT-LOGGED-IN SYSTAT

You can prevent people who are not logged in to a system from finding out the usernames of people who are logged in. If your site has public or network access, you should disable not-logged-in SYSTAT. Enter the following command in the n-CONFIG.CMD file:

DISABLE NOT-LOGGED-IN-SYSTAT

This increases security by making it harder for intruders who have connected to a system but not yet logged in to find out valid usernames for the system. To allow the SYSTAT command when users are not logged in, enter the following command in the configuration file:

ENABLE NOT-LOGGED-IN-SYSTAT

11.7 SECURING FILES

You and users can restrict access to files and directories according to the type of access or the particular user who attempts access: The command @SET FILE [NO] SECURE sets a file "secure" or not secure, and the commands @SET DIRECTORY and @BUILD cause all new files created in the directory to be set secure. (Refer to the TOPS-20 Commands Reference Manual for details on these commands.)

When a request is made to access a "secure" file, the monitor asks the ACJ for permission to open the file before the request is granted or denied. The ACJ bases its decision on the information contained in the ACCESS.CONTROL file, which you or users create and place in the same directory as files that are marked as secure. This file contains a list of filenames, access keywords, and usernames that have unrestricted access. It is important that the ACCESS.CONTROL file be set secure so that it cannot be changed.

The format of the ACCESS.CONTROL file is:

```
filename keyword user, keyword user,...,-
```

Where:

- o filename is the name of the file to be set secure
- o keyword determines the type of access granted to users of secure files:

  ALL (all access)
  APPEND (append access)
  DELETE (delete access)
  NOSECURE (permission to clear the SECURE bit)
  READ (read access)
  RENAME (permission to rename the file from or to filename)
  SECURE (permission to set a file secure)
  WRITE (write access)

- o user is the user who is granted access

The following is a sample ACCESS.CONTROL file:

```
ACCESS.CONTROL.*  READ Operator Staff.Mike Staff.Greg,-
SECURE Operator Staff.Mike Staff.Greg,-
WRITE Staff.Mike Staff.Greg,-
RENAME Staff.Mike Staff.Greg

ACJ.EXE.*   READ Operator Staff.Greg,-
SECURE Operator Staff.Greg,-
WRITE Staff.Greg,-
NOSECURE Staff.Greg

*.*.*   READ *,-
WRITE Staff.* Operator,-
SECURE Staff.* Operator,-
RENAME Staff.*,-
ALL Staff.Mike Staff.Greg Staff.Dave
```

11.7.1 Secure Files and the ACJ

To implement the secure-file feature, enable the SECURE policies in the ACJ, as described in Section 11.1.3.1. Refer to the descriptions of ENABLE SECURE-CHFDB, ENABLE SECURE-DELF, ENABLE SECURE-OPENF, and ENABLE SECURE-RNAMF.

11.7.2 Securing Important Files

It is recommended that you set secure those files that are sensitive to the operation and health of the system. Such files include MONITR.EXE. The ACCESS.CONTROL files should also be set secure.

11.8 SECURITY HINTS

Listed below are some general tips for enhancing the security of your TOPS-20 system.

- **Usernames**: No username should be used by more that one person at a time. It is important that each person accessing the system do so with a username uniquely assigned to that person. Sharing usernames hampers or, in some cases, prevents auditing of security problems.

- **Capabilities**: Since a user with WHEEL or OPERATOR capability has the power to change many aspects of system operation, the WHEEL and OPERATOR capabilities should be granted to a minimum number of usernames on the system.

  A need to access files should be handled by the directory and user groups structure but not by granting OPERATOR capability.

  Note that it is a particularly BAD idea to have one username with WHEEL or OPERATOR that a number of persons use.

- **Backups**: A "system tape" (containing the monitor, exec, user information, and system files from the boot structure) should be created once a week. Not only is this useful for hardware failures, but it can also provide a known secure copy of the monitor and all other software running on the system with enabled capabilities.

- **FILES-ONLY directories**: Directories on the system's PS: structure that are not normally used for usernames should be set FILES-ONLY. This prevents usernames from being created, which can lead to security problems.

- **OPERATOR username**: Prevent the OPERATOR username from logging in interactively by removing or expiring its password. OPERATOR should be used to run jobs needed for system operator (for example, jobs under SYSJOB or PTYCON that are logged in when the system is reloaded).

  A password does not have to be set for OPERATOR in order for jobs to log in to OPERATOR under SYSJOB or PTYCON. Operators should use their own usernames to back up the system areas and run OPR. (The OPERATOR user can run DUMPER under PTYCON to save the system areas.)

- **Local Changes**: It is important not to underestimate the positive effect that a few minor site-specific changes have on security. Anything that makes your system just a little different will slow down attempted break-ins. For example, change the ACJ log filename from the default.

CHAPTER 12

THE COMMON FILE SYSTEM

12.1 OVERVIEW

The Common File System (CFS) is a feature of TOPS−20 that allows users from more than one system to simultaneously access files. Any structure in the CFS configuration can be made available to any user for reading or writing.

Each TOPS−20 system in the CFS configuration has its own operating system, main memory, system structure, console, unit−record devices, and processes to be scheduled and run. But the systems are linked through a shared file system. This unified file system can be composed of all the disk structures on all systems. These structures appear to users as local to their own systems.

The main features of CFS are:

- It increases file accessibility. For example, if a system is down for maintenance, users can log onto another system and still access all files that do not depend on the down system for access.
- It lets you adjust loads on systems by reassigning users as loads require. (Or, users themselves may be allowed to switch systems as they see fit.) These changes need not result in file−access limitations.
- It lets you reduce the time that would be involved in maintaining duplicate sets of files.
- It lets you save disk space by minimizing duplication of files on different systems.

CFS (with Cluster GALAXY software) also lets users send jobs to printers connected to any system in the configuration.

12.1.1 CFS HARDWARE

The following are typical CFS configurations:

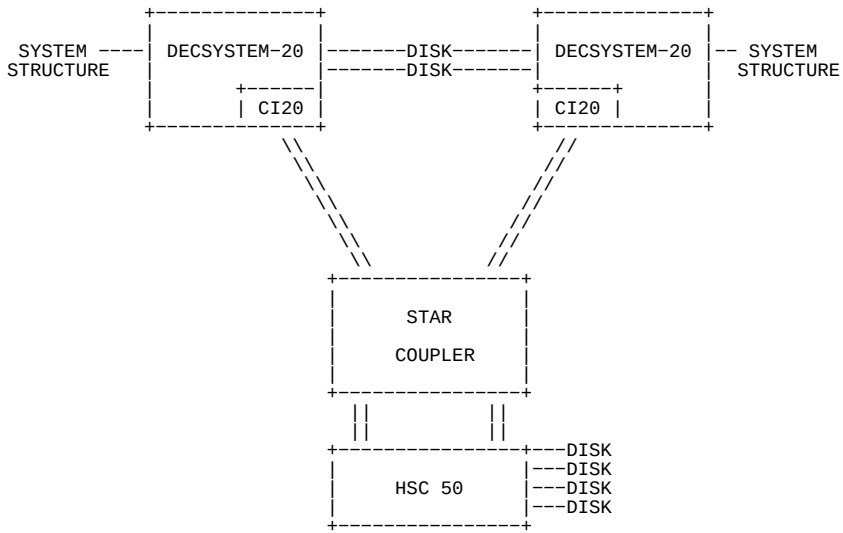


Figure 12−1: Two Systems with Massbus Disks and HSC50−based Disks

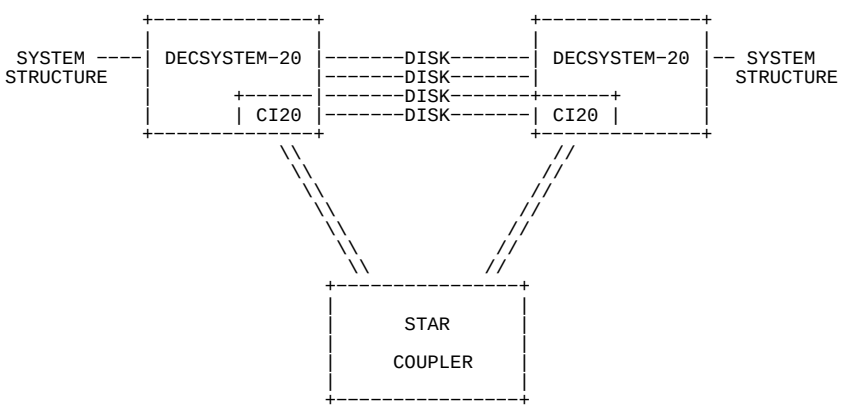


Figure 12-2: Two Systems with Massbus Disks

Star Coupler

The star coupler provides the physical interconnection for the CI cables among DECSYSTEM-20s and HSC50s. The maximum distance between a system and the star coupler is 45 meters.

A DECSYSTEM-20 can be connected to just one star coupler. That is, it can be part of only one CFS cluster.

CI

The Computer Interconnect (CI) bus is the communications link used by CFS. It also connects systems to HSC50-based disks (RA60s and RA81s). In addition, it provides access to massbus disks for systems without a direct connection to those disks, for example, to another system's system structure.

Each system has four communications links to the star coupler. Two of them are for transmitting data and the other two are for receiving data. The redundant CI connections are used for increased availability and performance. When one of the connections has failed or is in use, the CI microcode chooses the other one for data transmission. At start-up, TOPS-20 verifies that at least one set of transmit and receive connections is working.

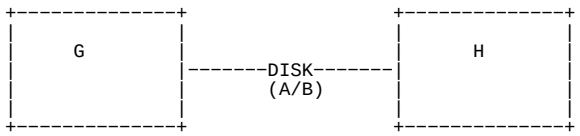
CI20

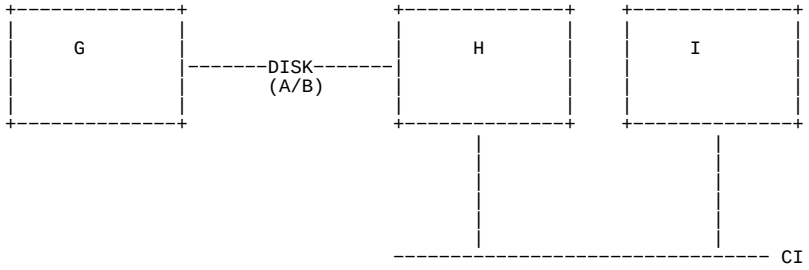
The CI20 port adapter provides the interface between the DECSYSTEM-20 and the CI bus. Only one CI20 is allowed per system.

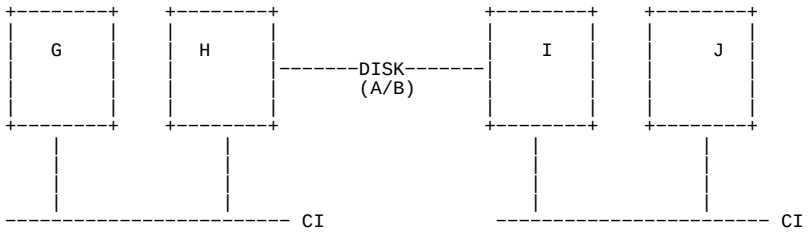
Massbus Disks

Multisystem access may be granted to all massbus disks.

It is recommended that massbus disks intended to be shared be dual-ported between two DECSYSTEM-20s (drive port switches placed in the A/B position). With a two-system CFS cluster, this avoids the overhead involved in file-server activity, as described later in this section. However, the systems must be able to communicate with each other over the CI; they must be connected to the same star coupler. Otherwise, neither system will be allowed access to the disk. Thus, the following configurations are not supported:







In the first two figures, systems G and H are not joined in a CFS configuration. The same applies to systems H and I in the third figure. TOPS-20 maintains the integrity of data on shared disks by ensuring that the systems can, over the CI, coordinate accesses to those disks.

Massbus disks not directly connected to a system are called "served disks" because TOPS-20's MSCP (Mass Storage Control Protocol) file-served facility makes this "outside" access possible. To enable an outside path to a massbus disk, that is, to make it a served disk, enter an ALLOW command in the n-CONFIG.CMD file, on a system to which the disk drive is connected, in the form:

ALLOW <drive type> serial number

The drive type is one of the following: RP06, RP07, or RP20. You can obtain the serial number with the command:

OPR>SHOW CONFIGURATION DISK-DRIVE<RET>

Note that TOPS-20 creates an RP20 serial number by adding 8000 to the disk drive unit number. Therefore, RP20 unit numbers should be unique among CFS systems.

To disallow access to a served disk that was allowed access, enter the following command in the n-CONFIG.CMD file:

RESTRICT <drive type> serial number

Disks are RESTRICTED by default if you do not specify ALLOW commands.

NOTE

Disks that make up the system structure must not be dual ported to another TOPS-20 system.

12.1.2 CFS SOFTWARE

Intersystem communication is an integral part of CFS. When TOPS-20 starts up, it makes a CFS connection with each TOPS-20 system that is already running. This establishes the contact necessary for intersystem file-system management.

In reality, only one system writes to a 256K section of a file at a time. When a system needs write access to a file section, it broadcasts a request for that resource to all systems it has established contact with. If another system already owns the desired write access, that system will respond negatively. Clearance will be granted to the requesting system only after the other system has completed the write operation by writing the data back to disk from its memory. Thus, systems negotiate for write access to files and keep each other informed of the state of the disks that they share. This ensures the integrity of data on those disks.

Because intersystem communication is vital to CFS operations, the systems stay alert to CI problems and to other indications that they may have lost contact with each other. Section 12.11.1, Communication Problems, discusses the actions that systems take when there is a breakdown in communications.

The INFORMATION CLUSTER command displays the names of HSC50s and CFS systems that are currently accessible.

DATE and TIME

When a CFS system starts up, it takes the date and time from the systems that are already running. The operator is not prompted for this information. Instead, the system types a message similar to the following on the operator's terminal:

The date and time is: Wednesday, 11-MAY-1988 9:38AM

This typeout serves as a check on the date and time. If no other system is running, the operator is prompted for the information.

When the date and time are changed on any CFS system, such as with the ^ESET command, all other systems are notified so that they can re-synchronize. This synchronization ensures that the creation date and time of files written from one system are consistent with the other CFS systems. Otherwise, many programs that use this information could malfunction.

12.1.3 CFS USERS

CFS is transparent to users:

- Users are normally unaware that someone from another system may be accessing a file at the same time that they are, except in such cases as the following. A file being read on system "A" will prevent its being renamed on system "B."
- Users are not required to know about the CFS configuration. Specifically, they do not need to know how massbus disks are ported. To access files, all they need to know are structure names, as on non-CFS systems.

The INFORMATION CLUSTER command lets users know what HSC50s and TOPS-20 systems are currently accessible to their systems.

12.1.4 CFS and DECnet

A CFS configuration differs from a DECnet network. Although a CFS configuration comprises multiple independent systems, the systems share a unified file system and cooperate in its operation. They function more as a single system than as systems merely communicating. If the optional DECnet-20 software is installed, each CFS system running DECnet is a DECnet network node with its own node name.

The files in CFS disk structures may be accessible to remote systems by way of such DECnet facilities as NFT. However, a node name is needed to access files in this way. CFS users, on the other hand, do not need to specify node names.

All systems in a CFS configuration must be TOPS-20 systems. In a DECnet network, however, other systems that support DECnet can be included.

DECnet on a system allows access to other CFS clusters as well as DECnet communication between systems in a cluster (for example, with the SET HOST command).

Table 12-1: Comparison of CFS and DECnet

Characteristic	CFS	DECnet
Multiple systems	X	X
TOPS-20 systems only	X	
One file system	X	
Node name in file spec		X
DECnet software		X
CI	X	X
NI		X

12.1.5 CFS and TIGHTLY-COUPLED SYSTEMS

A CFS cluster also differs from tightly-coupled multiprocessing environments. Each CFS system has its own main memory, which is not shared with another system. It also has its own system structure for booting and swapping and may have its own public structure for logging in. Also, CFS systems do not perform automatic load balancing. That is, the CPUs do not relieve each other of processing during high job loads. All jobs, including batch jobs, run only on the computer that the user logs onto.

12.1.6 Limitations

CFS does not coordinate use of the following facilities across systems: IPCF and OPENF OF%DUD. As an example, a DBMS application cannot span multiple systems, because DBMS uses the OPENF OF%DUD facility. Therefore, such applications should be restricted to a single system. Attempts to cross systems using these facilities will generate error messages.

CFS allows for shared disk files and line printers. However, it does not provide for shared magnetic tapes.

12.1.7 "Cluster Data Gathering"

The "cluster data gathering" system application (CLUDGR) is enabled by default in the n−CONFIG.CMD file. This "SYSAP" collects cluster−related data so that, for example:

- Users can obtain information on remote systems in the cluster by way of the SYSTAT command.
- Users can send messages throughout the cluster with the SEND command.
- Operators can obtain scheduling information on remote systems (SHOW SCHEDULER), receive structure status information from system responses during remote structure dismounts, and send messages to users throughout the cluster (^ESEND and SEND).
- System programmers can use the INFO% monitor call to obtain information on remote cluster systems. (As described in Section 11.1, you can control access to the INFO% monitor call through the access control program.)

You can disable and enable user, operator, and programmer CLUDGR SYSAP functions in the n−CONFIG.CMD file with the following commands:

```
DISABLE CLUSTER−INFORMATION
DISABLE CLUSTER−SENDALLS

ENABLE CLUSTER−INFORMATION
ENABLE CLUSTER−SENDALLS
```

NOTE

The CLUDGR SYSAP functions cannot be disabled for the GALAXY components.

During timesharing, the operator can disable and enable these same functions with the following privileged commands:

```
^ESET [NO] CLUSTER−INFORMATION
^ESET [NO] CLUSTER−SENDALLS
```

12.1.8 Cluster GALAXY

GALAXY is the TOPS−20 batch and spooling subsystem. In a cluster, it lets operators:

- Dismount a structure from a single terminal in a cluster even if the structure is mounted on more than one system in the cluster. The dismount with removal process is automated.

- Mount structures on a remote system in the cluster.
- Set a structure exclusive from a single terminal in a cluster even if the structure has been mounted on more than one system in the cluster. This process is automated.
- Send messages to all users on remote systems in the cluster.
- Control cluster printers
- Obtain remote information through most of the SHOW commands.
- Obtain the status of inter−system GALAXY DECnet connections.

Cluster GALAXY lets users:

1. Send jobs to cluster printers
2. Receive information on remote print requests
3. Cancel remote print requests
4. Receive notification of remote print job queuing and completion

"Cluster GALAXY" requires DECnet, TOPS−20 version 7, and GALAXY version 6.

You can disable cluster GALAXY on one or more systems by way of the GALGEN dialog. This dialogue can be run during or after system installation, and then GALAXY must be rebuilt. However, with the feature disabled, none of the remote functions listed above are available; the operating environment is as it was in GALAXY version 5, with TOPS−20 version 6.1. For example, to dismount a shared structure, the operator had to give commands from a terminal on each system on which the structure was mounted, and there was not a great deal of remote information to help the operator with this activity.

This chapter assumes that the feature is enabled.

12.2 PLACEMENT OF FILES

This section offers guidelines for arranging files on CFS systems for maximum performance and efficiency.

12.2.1 Update Files

Simultaneous shared writing to a file from multiple systems incurs the most overhead of any CFS file access operation. This is because systems involved in shared writing spend time seeking and granting write permission and coordinating their moves in other ways. Therefore, you might want to place the involved users on the same system.

12.2.2 Files on Served Disks

For optimum performance, you should not place on served disks files that require frequent access from multiple systems. This applies to both reads and writes. MSCP file-server operations incur considerable overhead, because the system with the direct connection acts as a disk controller for the accessing system. Therefore, such files should reside on HSC50 disks or, in a two-system CFS configuration, on massbus disks dual ported between systems.

12.2.3 Mail Files

By default, users' mail files are created and updated in their logged-in directories on the public structure. To access this mail, users log in and issue appropriate mail commands. They may have to go through this login procedure for every system that contains mail for them. You can change this default arrangement and simplify matters for the CFS user who has accounts on multiple systems. By redefining the systemwide logical name POBOX:, as described in Section 3.3.9, you can establish a central location on a sharable structure for all mail files in the CFS configuration. Then, no matter where users log in, the mail facility sees an accumulation of mail that could have been addressed to them at any system in the configuration. Mail is no longer isolated on individual public structures.

An added advantage to redefining POBOX: is that public structures do not fill up with mail files.

You must create a directory on the structure defined by POBOX: for every user in the CFS configuration who is to receive mail.

12.2.4 Sharing System Files

Most of the files that normally reside on system structures can be moved to a shared structure. Rather than duplicate files in such areas as SYS: and HLP: across systems, you can keep one set of these files on a shared structure. This saves disk space and eases the task of maintaining the files. Also, time and tape are saved during DUMPER backup and restore operations. Because system files are primarily read and not often updated, system performance does not suffer because of this file sharing, provided the structure is not on a server disk. If you consolidate system files, remember to include in the definitions for the systemwide logical names the structures that contain the files. For example, if the SYS: files reside on the structure COMBO:, the definition for SYS: would be:

```
DEFINE SYS: (AS) COMBO:<NEW-SUBSYS>, COMBO:<SUBSYS>,-<RET>
MAIN:<NEW-SUBSYS>, MAIN:<SUBSYS><RET>
```

where:

MAIN: is the name of a system structure

You should define structures in this way on all the systems, giving the appropriate system structure name. Make sure that the shared structure or structures are mounted UNREGULATED so that users will be able to access the files without having to give a MOUNT command.

The drawback to sharing system files is that if there is trouble with the shared structure, users on all systems suffer.

Most of the SYSTEM: files must remain on the system structures, so sharing these files is not recommended.

START-UP FILES

Certain files must remain on each system structure. These files are involved in system start-up and are required before a non-system structure is made available to a system. The following files should remain in each <SYSTEM> area:

```
7-CONFIG.CMD
7-PTYCON.ATO
7-SETSPD.EXE
7-SYSJOB.EXE
7-SYSJOB.RUN
ACCOUNTS-TABLE.BIN
CHECKD.EXE
DEVICE-STATUS.BIN
DUMP.EXE
ERRMES.BIN
EXEC.EXE
```

HOSTS.TXT
IPADMP.EXE
IPALOD.EXE
KNILDR.EXE
MONITR.EXE
MONNAM.TXT
TGHA.EXE

In addition, all the GALAXY files should remain in each <SUBSYS> area. These files come from the GALAXY saveset on the TOPS-20 Distribution Tape. (Refer to the TOPS-20 KL Model B Installation Guide.)

Command files that are used at your installation during start-up also should be kept on separate system structures. These files include SYSTEM.CMD and NETWORK.CMD.

12.3 LOAD BALANCING

This section discusses the distribution of jobs across CFS systems.

12.3.1 Dedicating Systems

One way to balance loads is to establish the types of jobs that will run on particular systems. For example, you might relegate batch jobs to one system, freeing other systems to run interactive jobs unimpeded. To encourage users to adopt this arrangement, you could give batch jobs the lowest priority on all but the batch-designated system. Users will have to wait a relatively long time for completion of batch jobs on non-batch systems. Refer to Section 10.2, SCHEDULING LOW PRIORITY TO BATCH JOBS, for further information.

Conversely, on the batch system, you could accord batch jobs the highest priority. Refer to Section 10.1.4, Procedures to Turn On the Class Scheduler, for details. Dedicating a system in this manner is especially useful when there are many long-running batch jobs at an installation.

Another suggestion is to put software development jobs on one system and production jobs on another. Also, you may want to keep one system lightly loaded for critical jobs.

DBMS applications and programming applications requiring IPCF facilities must be confined to one system. These are other items to consider if you choose to establish certain uses for particular systems.

Keep in mind that users must log onto the systems that are to run their particular jobs. This applies to batch jobs also (without

DECnet). Batch jobs must be submitted by a user logged in on the system where they are to run. The control and log files may reside on shared disks.

12.3.2 Assigning Users to Systems

In the CFS environment, much of the load balancing is expected to be performed by users. The systems, for example, do not detect that one CPU is overburdened and that another one is underutilized and, accordingly, reassign users' jobs. Instead, users themselves could determine whether or not they should log off a system and log onto another one when system response is slow. Such user tools as the SYSTAT and INFORMATION SYSTEM commands and the CTRL/T function can help users in this area. These tools report on the current state of a system. Among the items reported are the number of jobs running on a system, load averages, the current setting of the bias control "knob," and whether batch jobs are assigned to a special class. This information can be obtained for all systems in the configuration, not just for the user's logged-in system.

If you choose this load balancing scheme, you should create directories for all users on all the system structures in the CFS configuration. Also, directory usernames should be unique throughout the configuration, as described below. Then, users can log onto any system with no problem.

USERNAMES

Directory usernames should be unique throughout the CFS configuration. For example, there should be only one user with the username <BROWN> at an installation. This lets users access system resources without encountering password-related obstacles or causing security breaches.

If two users on different systems have the same usernames but different passwords, their passwords will be invalid when they switch systems. If these same users should by chance have the same passwords, they will have complete access to each other's files when they switch systems. Also, if a structure is mounted on both systems as domestic, neither user will have to give a password when accessing the directory on that structure that has their username. (Refer to Section 4.5.7, Mounting Structures from Another Installation, for a discussion of foreign and domestic structures.)

DIRECTORY AND USER GROUPS

To facilitate user access to CFS files, you could make directory and user group numbers consistent on all structures. That way, users could change structures or systems and their access attempts would have predictable outcomes.

12.4 STRUCTURE NAMES

Because the structures on all systems are part of a unified file system, structure names must be unique throughout the CFS configuration.

If it is necessary to mount structures with duplicate names, the operator should mount one of the structures using an alias. (Refer to Section 4.5.2, Mounting Structures Having the Same Name.) The system recognizes a structure by its alias, which is the same as the permanent structure identification name, unless otherwise specified. Note that everyone throughout the CFS configuration must refer to a structure by the same alias.

12.5 SYSTEM LOGICAL NAMES

Logical names are implemented differently from structure names and their aliases. Logical names are local definitions that need not be unique nor consistent throughout the CFS configuration. Thus, the same logical name on two different systems can refer to two completely different disk areas. However, because users are likely to be mobile, systemwide logical names should be consistent across systems. This will avoid confusion for users who switch systems.

Refer to Section 3.3, SYSTEM-LOGICAL NAMES, for further information.

12.6 SHARING STRUCTURES AMONG SYSTEMS

By default, all structures in the CFS configuration are accessible to all systems, provided outside paths have been established for massbus disks where necessary, using the ALLOW command (refer to Section 12.1.1, CFS HARDWARE). It is necessary to "mount" a structure on any system that is to access files on it, however. That is, the operator or a user on that system must issue a MOUNT command for the structure. (There can be up to 64 structures online on one system.) After a structure is mounted on a system, users can access it as on non-CFS systems. Users have automatic access to their public structure files, as on non-CFS systems.

If a structure has been restricted to a system through previous use of the operator command, SET STRUCTURE str: EXCLUSIVE, it can be made sharable again with the SET STRUCTURE str: SHARED command. The operator issues this command from a terminal running OPR on the system that has exclusive use of the structure. Then, MOUNT commands can be issued for the structure that has been made sharable. The default setting for structures is sharable.

The operator command, SHOW STATUS STRUCTURE, indicates the shared or exclusive status for all structures known to a system.

STRUCTURE ATTRIBUTES

The operator specifies attributes for a structure with the SET STRUCTURE command, as described in the TOPS-20 Operator's Command Language Reference Manual. They are permanent settings that do not revert to default values after system crashes and reloads.

Note that all systems need not have the same attributes in effect for a structure. For example, one system can have a structure mounted as foreign and regulated, and another system can have the same structure mounted as domestic and unregulated. Except for SHARED and EXCLUSIVE, attributes are on a single-system basis only.

12.6.1 Sharing System Structures

Bear in mind that when system structures are shared, privileged users can create privileged accounts on any system structure, with the ^ECREATE command. This may or may not be desirable.

12.6.2 Sharing the Login Structure

In a CFS-20 cluster, it may be advantageous to set up a homogeneous environment where all user accounts reside on a shared "login structure". Then, you do not need to maintain an account on every system to which a user has access.

12.6.2.1 Creating the Login Structure - To create this shared structure, give the following command to the CHECKD program for every system in the cluster:

CHECKD>ENABLE LOGIN-STRUCTURE(FOR STRUCTURE)str:(FOR CPU)cpu<RET>

where: str is the name of the login structure and cpu is the system's CPU serial number. The default serial number is the current system's.

This command adds the CPUs' serial numbers to the login structure's home blocks. The following command displays all the serial numbers that were entered into the blocks:

CHECKD>SHOW (INFORMATION FOR) LOGIN-SERIAL-NUMBERS (FOR STRUCTURE) str:<RET>

where: str is the name of the login structure

You should create a directory on this structure for each user in the cluster. (You can also put any other kind of directory on the login structure.) Users' LOGIN.CMD files and their .INIT files for various system programs should reside in these directories.

12.6.2.2 Enabling "Login Structure" – The system looks for user accounts on the login structure rather than on the boot structure when you enter the following command in the n-CONFIG.CMD file:

ENABLE LOGIN-STRUCTURE

12.6.2.3 Disabling "Login Structure" – You can disable the "login structure" feature with the following commands:

n-CONFIG.CMD file:

DISABLE LOGIN-STRUCTURE

CHECKD program:

CHECKD>DISABLE LOGIN-STRUCTURE(FOR STRUCTURE)str:(FOR CPU)cpu<RET>

where: str is the name of the shared structure and cpu is the system's CPU serial number. The default serial number is the current system's.

These commands cause the system to look for user accounts on the boot structure, which is the default condition.

12.6.2.4 PS: and BS: Directories – Before you enable the "login structure" feature, the public structure (PS:) is the boot structure (BS:), and is also known as the system structure.

After you enable the feature, the system considers PS: to be the login structure, the structure that contains all the user login directories.

The special system directories, described in Section 3.2, must remain on BS:, although you may choose to move many of their files to other directories. However, the files listed in Section 12.2.4 must remain on the boot structure in <SYSTEM>:

Also, the GALAXY components write files to SPOOL:, which the system defines at startup to be BS:<SPOOL>.

NOTE

Except where noted, this manual assumes that you have not enabled the "login structure" feature.

12.7 RESTRICTING STRUCTURES TO ONE SYSTEM

There may be times when you want to restrict use of a structure to a particular system. Such a structure might be used for DBMS applications (refer to Section 12.1.6, Limitations), or security measures may call for restricted use. For whatever reason, the operator restricts a structure with the following command:

OPR> SET STRUCTURE str: EXCLUSIVE<RET>

When the operator gives this command, the system first checks to see that the structure is not in use on other systems. If it is, the operator is given a list of those systems and asked whether or not this system should proceed with an automatic remote dismount of the structure from those systems (with the NO-REMOVAL option). This information and automatic dismount requires cluster GALAXY to be enabled. Ideally, the operator should beforehand follow the normal dismount procedure of making the structure unavailable to new users and notifying existing users of the pending dismount. The structure should be kept unavailable for all systems except the exclusive one so that the structure will not be inadvertently shared when the owning system crashes.

After a structure has been dismounted from other systems, the SET STRUCTURE EXCLUSIVE command can take effect. It remains in effect on the system, as do all SET STRUCTURE specifications, throughout crashes and reloads. If users give the MOUNT command for a structure that is exclusive to another system, an error message will be issued, indicating that the structure is unavailable.

Note that any system can have exclusive use of any sharable structure except another system's system structure.

Refer to the TOPS-20 Operator's Guide for details on setting structures exclusive.

12.8 DISMOUNTING STRUCTURES

When issuing a DISMOUNT command for a structure, operators have the option of specifying that the structure be physically removed from a disk drive. In the CFS environment, however, the system first ensures that the structure is not in use on other systems. If a structure is mounted on another system, the operator is notified and must go through the normal procedure of dismounting the structure (with the NO-REMOVAL option) from that system.

Note that with cluster GALAXY enabled, the operator can dismount structures remotely by performing all activities from an OPR terminal on the local system. The operator does not need to log onto any other system.

Throughout the dismount process, the operator receives various informational messages as well as error messages if, for example, the system cannot get an exclusive lock on the structure by way of the ENQ% monitor call or communicate with nodes on which the structure is mounted. (The system cannot communicate with nodes that have the cluster GALAXY feature disabled.)

Refer to the TOPS-20 Operator's Guide for details on dismounting structures.

The default setting on CFS systems is for a structure to be dismounted with the no-removal option.

Sometimes the system instructs the operator to dismount structures. This can occur for the following reasons:

- The operator attempts to shut down a system.
- The operator attempts to make the CI unavailable to a system.
- A system has been granted exclusive use of a structure.
- A structure has been physically dismounted from another system.

Refer to Sections 12.7, 12.9, and 12.12 for details.

These dismount instructions appear if you have included the ENABLE JOB0-CTY-OUTPUT command in the n-CONFIG.CMD file.

12.9 MAKING THE CI UNAVAILABLE TO A SYSTEM

Ordinarily, you need do nothing at all to operate the CI. However, you may need to disengage a system from the CI so Field Service personnel can diagnose and/or correct problems with the CI20 or the HSC50. Or, you may wish to remove a system from the CFS configuration. At those times, you should instruct the operator to make the CI unavailable by means of the SET PORT CI UNAVAILABLE command. (Refer to the TOPS-20 Operator's Guide for details.)

When the CI is unavailable to a system, users cannot access multi-access disks (dual-ported disks, HSC50-based disks, or served disks on other systems). These disks rely on the CI to coordinate accesses and/or to transmit data. Served disks on the system disengaging from the CI will be unavailable to other systems. Dual-ported massbus disks in the A/B position will have to be powered down and switched to one system.

When the operator gives the SET PORT CI UNAVAILABLE command, the system indicates the structures that need to be dismounted and the disk drives that need to be made unavailable. The operator is advised to follow the normal procedures of forewarning users before dismounting structures and making disk drives unavailable. The command option to forcibly disengage a system from the CI should be reserved for emergencies. If the operator determines that disengaging from the CI will be too disruptive to users, the operator has the option of aborting the procedure.

To put the CI back in operation, the operator gives the command:

OPR>SET PORT CI AVAILABLE<RET>

The operator is then asked if any other TOPS-20 system is running on the CI. If yes, the system rejoining the CFS configuration must be rebooted. If no, the CI20 will be reloaded and started. If the operator answers "no" and another TOPS-20 system is found after the CI20 has started, a CFRECN BUGHLT is issued on processors with lower serial numbers than the system joining the cluster and on this processor also, if there's a system in the cluster with a higher serial number. (See Section 12.11.1 for details.) After the system rejoins the configuration, structures that were affected when the CI was made unavailable will need to be remounted.

12.10 USING DUMPER

CFS offers operators and users flexibility in saving and restoring disk files. The only restriction is that DUMPER must be running on a system to which tape drives are attached. Tape drives are not served through CFS.

12.11 ERRORS

This section discusses the actions you or the operator take when errors occur in the CFS environment. It also describes how CFS systems react to various errors. Note that there is no single hardware or software point that can disable the whole configuration. For example, systems can start up or crash with little impact on other systems.

12.11.1 Communication Problems

CFS systems are sensitive to breaks in communication, whether they are caused by CI20 errors or system crashes. Because the data integrity of shared structures depends on unbroken intersystem contact, the systems take quick action to prevent data corruption. Therefore, you may observe any of the following when systems lose contact with each other. These should be rare occurrences.

- For a period of time calculated as 5 seconds per node (hosts and HSCs), no system in the configuration can access any multi-access disks (dual-ported disks, HSC50-based disks, served disks on other systems).

  This interval allows each system to check that its own CI20 and segment of the CI bus are working. Most likely, some system's CI20 microcode has stopped and is automatically reloaded during the interval, or a system has crashed. (There may be other, unpredictable reasons for CI disruption.) Jobs that were accessing multi-access disks are suspended until data integrity is assured.

  If the CI20 and CI bus are working before the end of the interval, the system can resume accessing all multi-access disks except server disks on a crashed system.

- A system crashes with a KLPNRL BUGHLT. This happens if the CI20 microcode takes longer to reload than 10 seconds. This BUGHLT is expected to occur rarely, because the microcode should be reloaded within a couple of seconds.

- If communication resumes after the interval mentioned at the beginning of this section, without the faulty system having crashed and restarted, the system with the lower serial number crashes with a CFRECN BUGHLT message as the faulty system tries to establish contact with each running system. That is, a system joining the cluster illegally will crash any system already in the cluster with a lower CPU serial number. The node itself will crash if there is already a node in the cluster with a higher CPU serial number. For example, this occurs when the SET PORT CI AVAILABLE command has caused communication to resume incorrectly due to operator error, as described in Section 12.9, MAKING THE CI UNAVAILABLE TO A SYSTEM.

  With such a delayed reconnection, a system is likely to contain old, invalid information about the status of multi-access disks. This is because other systems are allowed to access the disks after the interval, believing a faulty system is no longer running. Therefore, systems are selected to crash so that a fresh database can be established for the disks when the systems restart.

  EXAMPLE

  There are four systems in a cluster with serial numbers one through four:

System	Serial Number
A	1
B	2
C	3
D	4

  System B leaves the cluster and then tries to rejoin after the delay allowance has expired. System A crashes because its serial number is lower than system B's. System B crashes when it tries to establish contact with either systems C or D, whose serial numbers are higher than B's. Systems C and D remain running.

AT STARTUP

Sometimes, communication problems begin at system startup. A system that has just started up tries to communicate with each TOPS-20 system and HSC that is already running. After the SETSPD program sets the systemwide defaults, a system joining the CFS cluster checks to make sure that:

- Its own CI20 is working. If there is any malfunction, the "CFS joining" process is aborted. The system makes no further attempts to communicate with other CFS nodes and remains outside the CFS cluster.
- Its segment of the CI bus is working.

If the areas above are satisfactory, the system starting up then checks to see if, for each cluster node:

- The CI20 at the remote node is in maintenance mode (TOPS-20 nodes only, not HSCs). If so, the system knows that it cannot communicate with that node and tries to establish contact with the next node.
- Its own CI20 driver has created a system block for the remote node. The driver creates this block when the remote system responds to its request for recognition. The system block allows for a virtual circuit to be established between the two systems, over which inter-node data and messages are sent on the CI. If the block does not yet exist, the system sends a message to the CTY so that the operator can take appropriate action on the remote node. This situation usually indicates a hardware problem with the remote system's CI20.

If a system block has been created for a remote TOPS-20 node, the system tries to establish a CFS connection with that node by way of the virtual circuit. It is through CFS connections that systems communicate in order to coordinate access to shared disks. If the attempt fails, a message is sent to the CTY for operator action. This situation usually indicates a software problem, most likely that TOPS-20 is not running at the remote node.

| If the attempt at communication is successful, a confirming BUGINF is
| sent to the starting system's CTY.

A system starting up makes these communication checks for every other node in the cluster.

Refer to the TOPS-20 Operator's Guide for details on the operator
| information and error messages.

12.11.2 Massbus Problems with Dual-Ported Disk Drives

Dual-ported disk drives are accessed by each system through the massbus hardware connections. However, if for some reason a massbus path becomes unavailable to a system, the other system, with working massbus connections, can provide access to the drives affected, with the MSCP file server. The disks become "served."

The operator enables this facility by powering down the disks and flipping the drive port switches from the A/B position to the position that corresponds to the servicing system. Then the operator must reboot the system with the faulty massbus link. These procedures are required because a running system will never invoke the MSCP server after identifying a massbus path for a disk. It is assumed that an ALLOW command has been entered in the n-CONFIG.CMD file for the disk drives, as described in Section 12.1.1, CFS Hardware.

The operator returns the switches to the A/B position when the massbus problem is corrected. The PHYTPD BUGINF is then issued to confirm that the massbus will now be used for data transmission.

12.12 SHUTTING DOWN A CFS SYSTEM

When an operator issues the ^ECEASE command to shut down a system, outside jobs that may be accessing the system's served disks do not hang, with the following procedure. If any served disks have been mounted from other CFS systems, the operator is warned to check those systems for possible structure dismounting instructions.

At the other systems, meantime, if any served disks are mounted on the system shutting down, the operator is warned of the pending shutdown and is advised to dismount the structures listed.

In a CFS-20 cluster, a shutdown on one system causes "system going down" messages to be transmitted to all systems in the cluster at the sixty-minute, five-minute, and one-minute marks. For example, if SYSA is shutting down, the following messages appear clusterwide:

```
[System SYSA going down in 60 minutes at 1-Dec-87 16:29:22]

[System SYSA going down in 5 minutes at 1-Dec-87 16:29:22]

[System SYSA going down in one minute!!]
```

CHAPTER 13

LAT TERMINAL SERVERS

13.1 OVERVIEW

A local area network is a collection of computers and their resources that are linked together in a small geographic area, such as a college campus or a large building. Local Area Transport (LAT) software enables special-purpose computers to be used as terminal servers in such a network. With LAT software, user and application terminals that would otherwise be individually wired to host systems (DECSYSTEM-20s, for example) are instead connected directly to a LAT terminal server. The server, in turn, is linked to one or more hosts by way of the Ethernet Network Interconnect cable (NI), as shown below:

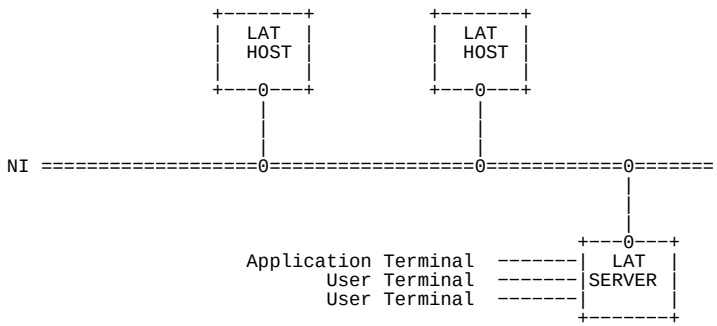


Figure 13-1: A LAT Network

A LAT host might be a TOPS-20, VMS, or RSX-11 system. The LAT server

is any NI-based terminal server.

An application terminal is a device under the control of an application process on the host. A printer is an example of an application terminal. An application process such as LPTSPL requests a connection to the LAT server when users want access to the device. The type of application terminal referred to in this manual is a LAT printer. Section 3.7 contains further information on LAT printers.

The main features of LAT servers are:

- Users can connect to any NI host that supports the LAT protocol, and it appears to them that their terminals have direct connections to those hosts. Connections between LAT terminal users and hosts are called sessions. There can be up to 128 sessions on a TOPS-20 host.

- By requesting multiple host connections, a LAT terminal user can enable multiple sessions at one host or at several hosts simultaneously. With one keystroke, such users can switch between their jobs on these systems.

- LAT servers can balance user loads among hosts according to a rating system that you set up.

- Users throughout the network can share LAT application terminals. An LN03 printer on a LAT server, for example, is not restricted to local host use.

Note that LAT software runs on host systems as well as on LAT servers. This chapter discusses how to control LAT host software from a DECSYSTEM-20. You can also issue commands directly to a LAT terminal server. Refer to the documentation provided with your LAT terminal server hardware for details on those commands.

13.2 LAT SOFTWARE

Each host that supports the LAT protocol for interactive terminal service periodically broadcasts this fact to all LAT terminal servers. Each server maintains a list of hosts that have sent such messages. A terminal user can issue a command to the server to display this list to see which hosts are accessible.

The logical path between a LAT host and server is called a LAT virtual circuit. There is one virtual circuit for each server/host pair that has at least one active terminal session. When a terminal user requests a connection to a host, the server creates a virtual circuit to that host if none exists. Otherwise an already existing circuit is used for data transmission. As system manager, you can decide the maximum number of virtual circuits that can exist at a host. The

number that you decide upon can affect system performance. This topic is discussed in Section 13.4.

Virtual circuit messages are transmitted between host and server at periodic intervals determined by a circuit timer maintained in the LAT server. When the timer expires, the server assembles into a single virtual circuit message any terminal input received during the past interval for a particular host and transmits it to the host. You can set this timer only at the server. It has a recommended value of 80 milliseconds, which is the default.

The data associated with a particular terminal are grouped in a virtual circuit message in units called slots. A virtual circuit message may contain slots from many terminals and may contain more than one slot from a single terminal. The maximum size of a slot is another parameter that you can set.

Having received a virtual circuit message, the host assembles as much terminal output data as possible into a single message and transmits it to the server. If no output is waiting for any terminal at that server, a message is sent with no slots. The host's response message acknowledges the previously received message from the server. This message will be acknowledged by the server message transmitted at the next tick of the circuit timer.

To reduce NI utilization and load on the host, the server transmits a message only when there is something to send. It could happen that when the server has no data to transmit, there is more terminal output data waiting at the host. But because the last host message remains unacknowledged, output flow from the host would stop. For this reason, the host is permitted to transmit one "unsolicited" message to the server. (Refer to the description of the HOST RETRANSMIT TIMER dynamic parameter in Section 13.4 for additional information.) The host sets a flag in the message, which forces the server to respond at the next tick of the circuit timer. This "forced" response acknowledges all the host's previously transmitted messages and returns all transmit buffers so that they may be used again.

The host maintains a circuit timer with a default interval of 1 second. (You can set it up to 2 seconds.) The function of the host's circuit timer differs from that of the server's circuit timer: it is used solely as a retransmit timer. When the host sends an "unsolicited" message, as described above, it starts its circuit timer. Because the server's circuit timer is shorter than the host's, the server should have acknowledged all outstanding host messages well before the host's timer expires. If this does not happen, the host retransmits all unacknowledged messages. This continues until either the server responds with an acknowledgement, or the retransmit limit is reached. If the retransmit limit is reached, the host assumes the connection to the server is no longer useable and detaches all LAT jobs from that server. (Refer to the description of the HOST RETRANSMIT TIMER and the HOST RETRANSMIT LIMIT dynamic parameters in

Section 13.4.)

13.3 DECNET

LAT, for the most part, is independent of DECnet. It does not require DECnet for general operations. However, DECnet is required on at least one host for start–up of LAT servers. Software from a host is loaded into the server's memory by DECnet's Network Management. This "down–line loading" of LAT servers occurs when the operator either physically boots the server or triggers a boot of the server by issuing a DECnet command from a host running DECnet. In either case, any host that has DECnet and the appropriate load files may respond to the boot, and then be selected by the server to load its memory. DECnet is also needed for "dumping" files of LAT memory images to a host for problem diagnosis on the host. Refer to the DECnet–20/PSI–20 System Manager's Guide for the operator procedures involved in loading and dumping LAT servers.

If a system supports DECnet, its node name and number are taken to be the LAT name and number also. You do not need to respecify them in the n–CONFIG.CMD file. Refer to the discussion of static parameters in Section 13.4.

13.4 CONTROLLING LAT FROM THE HOST

There are three ways for you or the operator or a system programmer to control LAT from a host:

- By rebuilding the monitor with new permanent parameters.
- By changing static parameters in the n–CONFIG.CMD file
- By changing dynamic parameters with the LAT Control Program subsystem of the OPR program

The lists below briefly describe these parameters. Many are discussed more fully later.

Note that you can also control LAT from the server itself. Refer to the documentation that came with your LAT server for details on server commands.

Permanent Parameters

Normally, you do not need to change permanent parameters. To change them, a system programmer familiar with monitor internals must rebuild the monitor.

- FRAME SIZE – The host or server guarantees that it can receive NI datagrams of at least this size. Usually three (2 transmit and 1 receive) buffers of this size are used for each LAT circuit. The default and recommended value is 1504 bytes.

- MAXIMUM HOST SERVICES – Sets the maximum number of services that this host can offer. You can specify up to 254 services. The default and recommended maximum is 8. Refer to Section 13.7, HOST SERVICES.

- MAXIMUM SLOTS – Sets the maximum number of slots that may be entered into a virtual circuit datagram for a given circuit. It is agreed upon by the server and host when the virtual circuit between them is established. The default and recommended value is 64.

- MAXIMUM SLOT SIZE – Sets the maximum number of bytes of data (not including the slot header) that the host can accept from the server in any slot. The slot size can range from 1 to 255. The default and recommended value is 40.

- MAXIMUM SERVER CACHE – Sets the maximum number of servers whose characteristics are kept in memory. The LCP command, SHOW SERVER, displays these characteristics. (Refer to Section 13.8.3, Displaying Server Information.) The default and recommended value is 20.

Static Parameters

- DEFAULT LAT ACCESS STATE – Determines LAT accessibility to and from the host. Values for the state are ON (the default) and OFF. The LAT access state can be changed dynamically with a LAT Control Program command. When the system is reloaded, however, the value for this static parameter again takes effect.

  Refer to Section 13.5, STARTING AND STOPPING LAT, for additional information.

- HOST NAME – Uniquely identifies the host within the local area network. It may contain up to a combination of six alphabetic and numeric characters, with at least one alphabetic character. The default host name is the DECnet node name, if the system supports DECnet. If the system does not support DECnet, you must specify a name. For related information, refer also to Section 13.7. That section discusses host service names.

- HOST NUMBER – Uniquely identifies the host within the local area network. The number can range from 0 to 65535. It is

  passed to the server when the virtual circuit between host and server is established. The default number is the DECnet node number, if the system supports DECnet. This is an optional parameter for use by one of your system programmers.

The n–CONFIG.CMD file commands that set these static parameters are:

– NODE hostname hostnumber

– LAT–STATE default LAT access state

where the default LAT access state is ON or OFF

Refer to the DECnet–20 PSI–20 System Manager's Guide for DECnet–related SETSPD commands, if applicable.

Dynamic Parameters

As system manager, you will probably be most concerned with dynamic parameters:

- HOST GROUP CODES – Defines the group codes for the host. They can range from 0 to 255. Code 0 is enabled by default. You can define any number of codes for a host. A LAT server will not connect to the host unless both server and host have at least one group code defined in common. Refer to Section 13.6, LAT GROUPS.

- HOST IDENTIFICATION – Supplies information about the host. It appears in various displays requested by managers and users. The identification may contain up to 64 printable characters. The default identification is the TOPS–20 banner message from SYSTEM:MONNAM.TXT.

- HOST NUMBER – Uniquely identifies the host within the local area network. The number can range from 0 to 65535. It is passed to the server when the virtual circuit between host and server is established. The default number is the DECnet node number if the system supports DECnet. This is an optional parameter for use by one of your system programmers.

- HOST MULTICAST TIMER – Determines the interval at which the host announces to all servers that its LAT terminal service is available. The default value is 60 seconds.

- HOST RETRANSMIT LIMIT – Sets the maximum number of times that the host retransmits a message at the expiration of HOST CIRCUIT TIMER. The virtual circuit is closed and all jobs associated with the virtual circuit become detached when this limit is exceeded. The default value is 30 times.

- HOST RETRANSMIT TIMER − Determines the interval at which the host retransmits any unacknowledged messages to the server. It is started when the host sends an "unsolicited" message and is stopped when the host receives any message from the server that acknowledges all outstanding messages. The default value is 1000 milliseconds (one second). It can be set from 1000 to 2000 milliseconds, however. Refer to the description of the HOST RETRANSMIT LIMIT parameter for related information.

- HOST SERVICE NAME − Specifies the name of a service that the host provides for users. It may contain a combination of up to 16 alphabetic and numeric characters as well as the dollar sign ($),dash (−), and underscore (_). The default service name is the host name. Refer to Section 13.7, HOST SERVICES.

- HOST SERVICE NAME RATING − Assigns a rating to a service name. It can be a fixed number in the range of 0 to 255 or it can be DYNAMIC. Refer to Section 13.7.1, Service Ratings.

- LAT ACCESS STATE − Determines LAT accessibility to and from the host. The default state allows LAT access. This dynamic parameter is affected by the LCP START and STOP commands.

- MAXIMUM ACTIVE CIRCUITS − Sets the maximum number of LAT virtual circuits that can exist simultaneously at the host. The default value is 20.

- MAXIMUM SESSIONS − Sets the maximum number of terminal sessions that may be active at the host simultaneously. The default value is equivalent to the number of terminals allowed in your system configuration. Section 13.1 discusses sessions.

The dynamic parameters are changed with the LAT Control Program (LCP). To use LCP, the operator enters the following from the OPR prompt:

1. For many LCP commands:

```
OPR> ENTER LCP<RET>
LCP> <LCP command><RET>
LCP> <LCP command><RET>
LCP> <LCP command><RET>
LCP> <LCP command><RET>
LCP> RETURN<RET>
OPR>
```

2. For a single command:

```
OPR> LCP <LCP command><RET>
OPR>
```


The LCP commands also let you start and stop operations, check parameter values and other information, and access counters maintained by the NI and servers. These LCP functions are discussed in later sections. The Operator's Command Language Reference Manual fully describes LCP.

The following is a summary of the LCP commands:

```
START
STOP
SET GROUPS (0,1−5,....255)
SET SERVICE−NAME service−name{/RATING:{nn|DYNAMIC}|
                             /IDENTIFICATION:"text"|
                              nothing}
SET IDENTIFICATION "text"
SET NUMBER number
SET MAXIMUM ACTIVE−CIRCUITS number
SET MAXIMUM SESSIONS number
SET MULTICAST−TIMER seconds
SET RETRANSMIT LIMIT number
SET RETRANSMIT TIMER number

CLEAR GROUPS (0,1−5,....255)
CLEAR SERVICE−NAME service−name
CLEAR IDENTIFICATION
CLEAR MAXIMUM CIRCUITS
CLEAR MAXIMUM SESSIONS
CLEAR MULTICAST−TIMER
CLEAR NUMBER
CLEAR RETRANSMIT LIMIT
CLEAR RETRANSMIT TIMER

SHOW CHARACTERISTICS
SHOW HOST−INITIATED−REQUESTS
SHOW PENDING−REQUESTS
SHOW SESSIONS
SHOW COUNTERS{/SERVER:server−name|
              nothing}
SHOW SERVER{/ALL|
            server−name|
            nothing}

ZERO COUNTERS{/SERVER:server−name|
              nothing}
```

13.5 STARTING AND STOPPING LAT

Support for LAT servers is enabled by default in TOPS−20. That is,the LAT access state is enabled unless specifically disabled in the n−CONFIG.CMD file. Disabling it is useful if you wish to establish

guidelines and set restrictions for LAT use before you enable it. You can set groups and the host identification, for example, before enabling LAT. The operator can enable and disable the LAT access state dynamically with the LCP START and STOP commands.

The host name and number are other parameters that you specify in the n-CONFIG.CMD file (unless DECnet is supported on the host). They are required items that uniquely identify the host in the local area network.

After the host name and number have been established, and the LAT access state has been enabled, the operator starts LAT by booting the server or by issuing a DECnet command from the host, as discussed in Section 13.3.

It is recommended that you disable the LAT access state if LAT is not intended to be used for an extended period of time. This will improve system performance.

13.6 LAT GROUPS

By using group codes, you can divide the local area network into smaller networks. That is, you can allow or prevent connections between specified servers and hosts. When users give a LAT command to display the names of available services, only those services corresponding to hosts within a server's "group" are displayed. (Refer to Section 13.7 for a discussion of services.) Codes in the range of 0 to 255 can be assigned to DECSYSTEM-20s and LAT servers. A LAT server will connect to a host only if it has at least one group code defined in common with the host. (Refer to the LAT documentation set for information on assigning codes to LAT servers.) Note that code 0, the default for hosts and servers, allows universal access. When servers and hosts implement the default, users have access to all hosts.

In the example below, an operator enables access between this DECSYSTEM-20 and any LAT server assigned a code in the range of 1 to 5. No connection is allowed with any other server.

```
LCP> SET GROUPS 1:5<RET>
LCP> CLEAR GROUPS 0<RET>
```

User terminals wired to servers in any one of these groups will be able to access the system.

Group settings stay in effect until reset with the CLEAR GROUPS command.

13.7 HOST SERVICES

Another name for a LAT host is a service node (a node being a system in a network). This refers to the fact that hosts provide computing services. Users access hosts in order to have some type of computing need served--to compile a program, update a file, or print a report, for example. LAT terminal servers deliver these services to users.

Users refer to hosts by the services that they offer, rather than by host name. When they request the LAT server to connect them to a host, they specify a service name that you have previously established for that host. The server maintains a list of host and service names and lets users display service names assigned to hosts in the user's group. (The server also displays any service identification text you specified.) With the SET SERVICE-NAME command, you can specify one or more services for a host:

```
LCP> SET SERVICE-NAME LASER-PRINTER/IDENTIFICATION:"OMEGA System"<RET>
LCP> SET SERVICE-NAME ARPA-GATEWAY<RET>
```

The default service name for a host is the host name. You may want to specify some identifying text for such services. For example, on the host ALPHA, you could give the command:

```
LCP> SET SERVICE-NAME ALPHA/IDENTIFICATION:"A DECnet System"<RET>
```

You can assign the same service name to multiple hosts. The following section discusses this topic.

13.7.1 Service Ratings

You can arrange for LAT servers to distribute users among host systems, according to a rating system that you set up. This is useful for hosts with service names in common. Perhaps several systems are part of a CFS cluster. You can assign the same service name, CFS, on each host but specify a unique service rating for each one:

```
LCP> SET SERVICE-NAME CFS/RATING:3
    /IDENTIFICATION: "ALPHA/BETA/OMEGA Cluster"<RET>
LCP> SET SERVICE-NAME CFS/RATING:9
    /IDENTIFICATION: "ALPHA/BETA/OMEGA Cluster"<RET>
```

If a user requests connection to CFS, the server will pick the host with the highest rating for that service. If, for some reason, that connection fails, the server will try the host with the next highest rating.

Ratings can be a fixed number from 0 to 255. Or they can be DYNAMIC. When hosts with common service names have DYNAMIC ratings for the service, the hosts compute ratings using an algorithm based on system load averages. Generally, the host with the lowest load average is given the highest rating. That host probably has the greatest available computing capacity.

You can use common service names for any collection of hosts offering the same function.

The default rating is 1 for the default service name, and 0 for services created with the SET SERVICE-NAME command.

Service Rating Example

Hosts SOLAR and LUNAR, in addition to providing a timesharing service as service names SOLAR and LUNAR, each have the service name CFS. SOLAR assigns a host rating to name CFS of 5; whereas LUNAR assigns 3. A LAT terminal user, when displaying the list of available LAT services, will see SOLAR, LUNAR, and CFS. If the user connects to CFS, the server will first attempt to access SOLAR. If that fails, it will try LUNAR.

13.8 MONITORING LAT FROM THE HOST

The following sections show various LAT informational displays.

13.8.1 Displaying User Information

The SHOW SESSIONS command displays information about connected LAT terminals:

```
LCP>SHOW SESSIONS<RET>
LCP>
10:52:26                      [LCP] Active LAT sessions

Job Line Program Server Name      Port Name        User
105  326 EXEC    COTTAGE          PORT7            DOPEY
114  327 LPRINT  PALACE          *LASER-PRINTER    THE_QUEEN
114  330 LPRINT  PALACE           ROYAL-TTY        THE_QUEEN
120  331 EXEC    COTTAGE          PORT3            GRUMPY
113  332 EMACS   PALACE           KINGS-TTY        THE_KING
 * indicates an Application Terminal
```

The TOPS-20 SYSTAT user command shows much of this same information.

13.8.2 Displaying Host Parameters

The SHOW CHARACTERISTICS command displays many of the LAT parameter settings:

```
LCP>SHOW CHARACTERISTICS<RET>

LCP>
09:20:41 [LCP]                                  -- Host Characteristics --

LAT Access State: ON
Host Name: ALPHA
Host id: ALPHA, TOPS-20 Development System, TOPS-20 Monitor 7(20753)
Host number: 140
Retransmit Limit: 60
Retransmit Timer: 1000
Multicast Timer: 30
Groups: 3:4,7,10,18,21:23,45,47

                     Current   Maximum
                     -------   -------
Allocated circuits       6        32
Active circuits          6        20
Sessions                 8       128

Service name   Rating        Identification
------------   ------   ------------------------
   ALPHA         1      ALPHA - More Networks per CPU
   SGROUP        D      Software Engineering Cluster
```

The D that appears for the service name rating indicates a dynamic rating.

13.8.3 Displaying Server Information

The SHOW SERVER command displays information about servers with connections to this host. The host tries to keep information in memory on all servers that have connected since the last monitor load. However, this could require a very large data base. Therefore, information is kept only for the number of servers specified in the MAXIMUM SERVER CACHE permanent parameter. (Refer to Section 13.4 for information on this parameter.) When this number is exceeded, the oldest inactive entry is deleted from the data base, making room for a new entry.

You can specify a single-line summary for each server or a detailed display for a single server:

```
LCP>SHOW SERVER/ALL<RET>
LCP>
14:55:32                        [LCP] Summary of all servers

Server Name(Number): Finance(8) Address: 08-00-2B-00-17-BA
Server Name(Number): Accounting(22) Address: 08-00-2B-02-08-C0
Server Name(Number): Payroll [LAT3](3) Address: AA-00-03-01-25-38
Server Name(Number): Development(2) Address: AA-00-03-01-06-AB


LCP>SHOW SERVER FINANCE<RET>
LCP>
14:56:12                        [LCP] Information about server Finance

Server Number: 8
Server Location: Near vending machines
Server Type: DECserver-100
Ethernet Address: 08-00-2B-00-17-BA
Server Status: Connected
Max Slots: 32
Data Link Size: 1518
Circuit Timer(ms): 80
Keep-alive Timer(s): 20
```

13.8.4 Displaying LAT Counters

The SHOW COUNTERS command displays counters kept by LAT software modules. These counters provide information such as the number of messages transmitted, the number of transmission errors, and so on. You can obtain these numbers for a single server, or you can obtain cumulative figures for all servers that have connected since the last monitor reload:

```
LCP>SHOW COUNTERS<RET>
LCP>
14:48:11                        [LCP] Counter totals for all servers

Messages received: 33955
Messages transmitted: 36413
Messages retransmitted: 0
Sequence errors received: 21
Illegal messages received: 0
Illegal slots received: 0
Resource failures: 0
```



```
LCP>SHOW COUNTERS/SERVER:PUBLICATIONS<RET>
LCP>
14:48:34                        [LCP] Counters for server PUBLICATIONS

Messages received: 28189
Messages transmitted: 30132
Messages retransmitted: 0
Sequence errors received: 0
Illegal messages received: 0
Illegal slots received: 0
Resource failures: 0
```

The single-server counts are available even after a server has disconnected from the host. However, this availability, as the information displayed with the SHOW SERVER command, depends on the limit set with the MAXIMUM SERVER CACHE permanent parameter.

The cumulative counters are incremented each time individual server counters are. It is possible that the sum of all server counters is not equal to the cumulative counts, because you can zero the counters for any server, as discussed below, or the data base may have been cleared according to the MAXIMUM SERVER CACHE parameter limitation.

When using the counters to monitor performance or to isolate hardware failures, it is often desirable to be able to reset (or zero) the counters. The ZERO COUNTERS command lets you do this. You can reset counters for one server or for all the servers. Resetting the cumulative counts does not affect the counts of the individual servers.

13.8.5 Displaying Pending Requests for LAT Application Terminals

The SHOW PENDING-REQUESTS command displays information on pending, rejected, and canceled requests for LAT application terminals:

```
LCP>SHOW PENDING-REQUESTS<RET>
LCP>
09:20:49 [LCP]                        -- Current Pending Connect Requests --

Job Status    Server Name     Service Name      Port Name     User
--- ------ ---------------- ---------------- ------------ ----------
146 QUE  1 LAT100                            LN03         WADDINGTON

QUE nn means request was queued; entry is nn requests into the queue

A total of 1 request was found
LCP>
```

LAT TERMINAL SERVERS

13.8.6 Displaying All Print Requests for LAT Application Terminals

The SHOW HOST−INITIATED−REQUESTS command displays information on all active and pending requests for LAT application terminals:

```
LCP>SHO HOST
LCP>
09:21:10 [LCP]                                  −− Current Host−Initiated Requests −−


Job Status    Server Name     Service Name      Port Name          User
−−− ------ ---------------- ---------------- ---------------- ---------------
130 TTY334 LAT100                           LN03             OPERATOR
146 QUE  1 LAT100                           LN03             WADDINGTON

TTYnnn means connect is active and terminal nnn was assigned
QUE nn means request was queued; entry is nn requests into the queue

A total of 2 requests were found
LCP>
```

–T–

–U–

–V–

–W–

–X–