

TOAD-1 System Installation and Operation Manual for TOPS-20



special purpose computer design, manufacturing, and sales

8420 154th Avenue NE
Redmond, Washington 98052
(206) 869-9050 FAX: (206) 861-7863

DISCLAIMER

This draft document is still changing. As of the date of publication below, the information included matches the way the system operates. Subsequent changes may invalidate parts of this manual.

All material contained herein is proprietary to XKL Systems Corporation.

Part Number 50103-00002

November 22, 1996

Copyright ©1995, 1996 XKL Systems Corporation.

This document contains information which is protected by copyright. All rights are reserved. Reproduction, adaptation, or translation without prior written permission is prohibited, except as allowed under the copyright laws.

Restricted Rights Legend. Use, duplication, or disclosure by the United States Government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013 for Department of Defense agencies, and subparagraphs (c)(1) and (c)(2) of the Commercial Computer Software Restricted Rights clause at FAR 52.227-19 for other agencies.

Use of this manual and any software supplied with it is restricted to this product only. Additional copies of the programs may be made for security and back-up purposes only. Resale of the programs, in their present form or with alterations, is expressly prohibited.

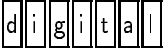
Warranty

The information in this publication is subject to change without notice. The information contained herein should not be construed as a commitment by XKL Systems Corporation.

XKL Systems Corporation makes no warranty of any kind with regard to this material, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose.

XKL Systems Corporation shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance, or use of this material.

This document may contain material previously published by Digital Equipment Corporation. Any such material is used here under license from Digital Equipment Corporation.

The following are trademarks of Digital Equipment Corporation: CI, DEC, DECnet, DECUS, DECSYSTEM-10, DECSYSTEM-20, DDT, HSC, HSC-50, MASSBUS, PDP, PDP-10, TOPS-10, TOPS-20, TOPS-20AN, UETP, and 

National Electrical Code and NEC are registered trademarks of the National Fire Protection Association

PostScript is a trademark of Adobe Systems Incorporated. UNIX is a trademark of some organization, probably Novell.

Notice:

The TOAD-1 System has been tested and found to comply with the limits for a Class A digital device, pursuant to part 15 of the (U.S.) FCC Rules. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment. This equipment generates, uses, and can radiate radio frequency energy and, if not installed and used in accordance with the instructional manual, may cause harmful interference to radio communications. Operation of this equipment in a residential area is likely to cause harmful interference in which case the user will be required to correct the interference at his own expense.

Preface

The TOAD-1 System Installation and Operation Manual for TOPS-20 describes the procedures for installing a new TOAD-1 System and customizing it to a particular network environment. This manual also briefly describes how to manage and operate the system, including such topics as generating tailored GALAXY software.

In several places in this document, nouns (e.g., boot) are used as verbs and vice versa. This is done for conciseness at the expense of Standard English usage. We apologize in advance to any offended readers.

Audience

This manual addresses TOPS-20 operators, system managers, and others who have the responsibility to:

- Install the new TOAD-1 System and customize the software for the particular TCP/IP environment found at the site.
- Test the success of the installation.
- Manage and operate the system.
- Generate a tailored batch and spooling (GALAXY) system.

Organization

- Chapter 1 **Site Preparation** describes the physical environment necessary to support the TOAD-1 System.
- Chapter 2 **Physical Setup** explains how to unpack the TOAD-1 System and connect it to the surrounding environment. This chapter also describes the indicator lamps found on the system and on the individual circuit boards.
- Chapter 3 **Initial Boot** describes the procedure for booting the system the first time. This chapter shows what you can expect to see when you turn on the TOAD-1 System for the first time. This chapter also explains some of TDBOOT, the bootstrap program. (A detailed summary of TDBOOT commands can be found in an appendix of the *TOAD-1 System Architecture Reference Manual*).
- Chapter 4 **System Administration** describes the most important aspects of customizing TOPS-20 to serve your user community. Topics covered include the configuration

of TOPS-20 for your network environment, the creation of login accounts for users, and the creation of file-storage directories. This chapter also discusses security considerations in the network environment, and the configuration of printers on the network. Additional materials are found in *TOPS-20 System Manager's Guide*.

- Chapter 5 **System Operation** describes some of the tasks that are required occasionally: shutting down timesharing, rebooting, rebooting for stand-alone operation; file system tasks: creating a file structure, creating directories, backing up a file structure, checking file system consistency, rebuilding a file system from a back up tape, and formatting a disk. Further, tape back up policies are described, as are a set of periodic tasks. Additional materials are found in *TOPS-20 Operator's Guide*.
- Chapter 6 **Running the TOPS-20 UETP Package** describes how to run the User Environment Test Package (UETP program). This program performs a cursory check of the system.
- Appendix A **7-CONFIG.COMD Commands** describes the system configuration command file, `7-CONFIG.COMD`, providing a detailed list of the configuration commands and their parameters. This appendix describes how to declare the system name, and change the system defaults¹ for system devices, accounting, performance improvements, scheduler controls, file archiving and migration, and control other system functions.
- Appendix B **OPR Commands** describes commands to the OPR program; these are covered in greater detail in *TOPS-20 Operators Command Language Reference Manual*.
- Appendix C **CheckD Commands** describes commands to the CHECKD program. This program is used to create and maintain file structures.
- Appendix D **TOPS-20 Start Up Operations** describes how TOPS-20 initializes itself, so that a programmer or administrator has a better understanding of how the configuration parameters interact. Appendix E also describes some TOPS-20 internal variables that affect system operation in a global sense; these are mostly of interest to programmers.
- Appendix E **Generating a Tailored Batch and Spooling System** contains the steps for generating a batch and spooling (GALAXY) system tailored for your particular site.

Related Documents

The initial software provided with the TOAD-1 System, on disk and on the DAT tape shipped with the system, includes a variety of documents as text or PostScript files.

The manuals and guides listed below are valuable as references while using this manual. Some of these have been prepared by XKL: they describe the TOAD-1 System specifically. These manuals are available as PostScript files in the directory `BS:<DOCUMENTATION.TOAD>`. Other manuals are precisely as Digital Equipment Corporation made them, circa 1990. These files were intended for the KL10-based DECSYSTEM-20. They are mentioned here because they do have some applicability to the TOPS-20 system provided on the TOAD-1 System. The text files from Digital are found in the directory `BS:<DOCUMENTATION>`.

Pertaining to all users:

¹The word “default” is used in its sense “in the absence of”. That is, the “default” action is what the program does in the absence of instructions that specify some other action.

TOPS-20 User's Guide provides an overview of the TOPS-20 system and its general capabilities. This guide is online in **USERS.MEM**.

TOPS-20 Commands Reference Manual documents the commands handled by the TOPS-20 EXEC. The online version is in **COMMANDS.MEM**.

TOPS-20 User Utilities Guide, which documents the the programs MAIL, RDMAIL, FILCOM, CREF, MAKLIB, DUMPER, and PLEASE. Of these, the documentation for DUMPER is most relevant to the procedures in this Installation Manual. The Utilities Guide is online in **USER_UTILS.MEM**.

For system managers and operators:

TOPS-20 Operator's Guide is online in **OP_GUIDE.MEM**.

TOPS-20 Operator's Command Language Reference Manual is the guide to the OPR program and its commands. It is online in the file **OPRCOM.MEM**.

TOPS-20 System Manager's Guide is online in **MGR_GUIDE.MEM**.

TOPS-20 Bug Messages Manual is derived from the file **<SYSTEM>BUG-DOC.TEX**. This manual contains the complete messages for each of the TOPS-20 BUGHLT, BUGCHK, and BUGINF error codes. The complete manual is in **BUGS.PSF**.

This manual, *TOPS-20 Installation and Operation Manual*, is available in hardcopy and as a PostScript file in **INSTALL.PSF**.

For systems programmers:

TOAD-1 System Architecture Reference Manual describes the PDP-10 instruction set, the privileged instructions used by the operating system, and the interconnection of the various system components. An appendix describes all the commands recognized by TDBOOT. This manual is online in **ARCHITECTURE.PSF**.

TOPS-20 Monitor Calls Users Guide provides an overview of the system calls. It is online in **JSYS_USERS.MEM**.

TOPS-20 Monitor Calls Reference Manual documents the system calls supported by TOPS-20. The online version is in **JSYS_REFERENCE.MEM**.

The JSYS program provides online lookup of JSYS documentation by the name of a particular system call. This program refers to the file **7-JSYS.MAN**.

Using Input Recognition

The TOPS-20 EXEC command processor (and many other TOPS-20 programs that read commands from terminals) offers a helpful feature to prompt you when you type a command. Simply type the first few characters, and then press the "Escape" key (this key is generally labeled "ESC"). Pressing the Escape key, signalled in this document by the mark ESC, tells TOPS-20 to examine the

characters you typed and to complete the command for you. If TOPS-20 recognizes the characters you typed, it completes as much of the command as it can; this is called *command recognition*. When it cannot print any more, or does not recognize the characters you typed as part of a valid command, TOPS-20 rings the terminal bell (nowdays, this is generally a beep or similar noise) and waits for you to type additional characters. If you type part of a command that means more than one thing to TOPS-20, it rings the terminal bell to indicate that it needs more information. By using the Escape key in these ways, you are employing the TOPS-20 feature called input recognition.

When you use input recognition, TOPS-20 may also print some words in parentheses. These words, called “guide words”, make the command more readable and also tell you what kind of *parameter* TOPS-20 expects you to type next.

In this manual, commands are shown as complete words followed by the appropriate parameters.² Most commands end (and are acted upon) when you press the “Return” key (sometimes labeled as “Enter”). The mark RET signifies a place where the Return key must be pressed. Also, the portion of the command that you type will be marked with underlining. For example:

```
!Build PS:<OPERATOR> RET
```

If you use input recognition for this command, it looks like this:

```
!BuilESC (DIRECTORY NAME) PS:<OPERATOR> RET
```

Input recognition completes the partial command name “Buil” by adding the letter “D”; it then provides the guide words “(DIRECTORY NAME)”.

Here is another example:

```
!COPY PS:<SYSTEM>FILE.CMD PS:<OPERATOR>FILE.CMD RET
```

Using input recognition, the command looks like this:

```
!COPYESC (FROM) PS:<SYSTEM>FILE.CMDESC (TO) PS:<OPERATOR>FILE.CMD RET
```

A further facility, related to input recognition, is invoked by the question mark character, “?”. When you type a question mark in the middle of a command, TOPS-20 will tell you what kind of parameter is required. For example:

```
!Build ? Old directory name being modified  
or, New directory name being created  
!Build  
    ~ the input pointer is left at this spot
```

²Command parameters are also called *arguments*.

This example shows that when you type “?”, TOPS-20 tells you what kind of parameter the program wants to see. It then retypes your command, up to the point where you typed the question mark, and it awaits your further input.

If you type accurately, the command lines shown in this manual will be accepted and interpreted correctly by TOPS-20. If you choose, you can use input recognition. In most cases, if you press the Escape key after you type a command or a parameter, TOPS-20 will provide guide words or instruct you to press the Return key.

Input recognition offers several advantages. First, you can double-check your typing, because the completed command as printed by TOPS-20 verifies that your typing has been interpreted correctly. Second, you can minimize typing, because you need only type enough characters to make a unique match with a command or parameter. And third, TOPS-20 prompts your next response, because it prints guide words in parentheses.

How to Use this Manual

You might find it helpful to be familiar with the information found in the *TOPS-20 User's Guide*, the *TOPS-20 System Manager's Guide*, and the *TOPS-20 Operator's Guide*. But you have to start somewhere; you can start with this manual, but be aware of the “Related Documents” listed above.

If you are installing a new system, read the entire manual. If you are generating a tailored GALAXY, read Appendix E.

Typographical conventions used in this manual are described in Table 0.1.

The commands you type are underlined in this manual. If you type accurately, all you have to do is perform the indicated function and verify that your actual terminal output resembles the sample output. Be sure to include the spaces shown in the command lines.

Remember that systems differ in memory size and peripherals, so steps that describe configuring memory, configuring network printers and disk drives, and assigning logical unit numbers to tape drives, require you to type the values that are correct for your system, and not the sample parameters shown in the examples.

If your terminal output does not resemble the sample output, first read **all** of the instructions after the word “Error” then choose the recovery procedure that corrects your problem.

Note

The version and edit numbers in this manual may differ from those printed on your terminal. The numbers printed on your terminal must be equal to, or greater than, the numbers in this manual.

Time estimates are included so that you know about how long a step takes. Times are estimated to the nearest minute, so don't worry if a step takes somewhat longer or shorter than this.

Table 0.1: Typographical Conventions

Symbol	Indicates
<code>RET</code>	The “Return” or “Enter” key.
<code>ESC</code>	The “Escape”, “ESC”, “ALT”, “ALTMODE”, or “PREFIX” key.
@	The TOPS-20 EXEC program’s prompt character as represented in this manual. Your actual prompt may differ, as the prompt can be set by each user. As delivered, your system will prompt with “Host@”.
@@	The EXEC’s subcommand prompt.
!	The EXEC’s Enabled prompt character as represented in this manual. Your actual prompt may be different.
!!	The EXEC’s Enabled subcommand prompt.
\$	In some programs, e.g., DDT, the echo of <code>ESC</code>
<CTRL/\>	Press and hold the CTRL (Control) key, then type the backslash key. When typed on the console terminal, this key summons the micro-command processor, which prompts by typing either <code>XKL-1></code> or <code>XKL-1%</code> ; the former signifies that a program is running, the latter signifies that the PDP-10 instruction-set processor is halted.
<CTRL/x>	Press and hold the CTRL (Control) key, then type the other indicated key, in this example “x”.
^X	The echo of <CTRL/x>. For example, pressing <CTRL/z> echoes as ^Z. Note that pressing <CTRL/x> does not always echo with ^x on your terminal. For example, <CTRL/\> has no echo at all.
Typewriter Font	Actual (or typical) output from the computer. In running text, this font also signifies the actual name of a command or the name of a file.
<u>Typewriter Font</u>	Actual input required in a command string. You must type such an element in full or with an abbreviation acceptable to the system. Generally, TOPS-20 commands and file names are not sensitive to any difference between uppercase and lowercase characters.
<u>^X</u>	Actual input of <CTRL/X>
<i>Italic Font</i>	An input variable in a command string (for example, name or number). These are variables that you must determine.
Spaces	Spaces are used to separate elements of a command. A space must be input where shown. Generally, TOPS-20 will tolerate multiple spaces where one space is shown.
SMALL CAPS	The names of programs typically appear in small capital letters.
{ X Y Z }	A list of alternatives within a command. You must specify one of the alternatives in your command string. Do not type the braces or the vertical bar in the actual command string.
:	This mark indicates that some of the lines in the example or figure are not shown.

Revision History

4 December 1995 to Present

- This revision list has been created and added to the manual.
- Class A warning has been added to the frontmatter and chapter 1.
- Language describing the emergency power off connection was changed.
- The installation procedure for the XTU-1 has been added.
- The chapter on GALAXY has been added.
- The frontmatter has been expanded.
- Figures depicting the machine and the 16-bit to 8-bit converter were added.

Contents

Preface	v
Revision History	xi
1 Site Preparation	1
1.1 Electrical	1
1.1.1 Power Connection	1
1.1.2 Emergency Power Off Connection	1
1.2 Air Conditioning	2
1.3 Network Environment	3
1.3.1 Terminal Server	3
1.3.2 Network Printer	3
1.4 Radio Interference	4
2 Physical Set Up	5
2.1 Unpacking	5
2.2 Assembly	6
2.2.1 Mounting	6
2.2.2 Battery Installation	6
2.3 Power Connections	8
2.3.1 Mains Power	8
2.3.2 Emergency Power Off	9
2.4 External Device Connections	9
2.4.1 Console Terminal	9
2.4.2 Auxiliary Console	9
2.4.3 SCSI Devices	10
2.4.3.1 Connectors	10
2.4.3.2 General SCSI Cabling	11
2.4.3.3 DAT Drive Cabling	11
2.4.3.4 XTU-1 Cabling	11
2.4.3.5 The SCSI “Standard”	13
2.4.3.6 SCSI Target ID Selection	13
2.4.4 Network	14
2.5 Indicator Lamps and Jumpers	14
2.5.1 XKL-1 Processor	15
2.5.2 XRH-1 SCSI Bus Controller	15

2.5.3	XNI-1 Ethernet Controller	16
2.5.4	XMG-1 Memory	16
3	Starting the System	17
3.1	Initial Boot	17
3.2	Other Boot Operations	22
3.3	Abnormal Conditions	23
3.3.1	Board Swapping	23
3.3.2	Backplane Swapping	23
3.3.3	Disk Structure Problems	24
4	System Administration	25
4.1	Configuring TOPS-20	25
4.1.1	System Definitions	25
4.1.2	Network Configuration	28
4.2	Users, Directories, and Accounts	30
4.2.1	Defining New Users	30
4.2.2	FINGER Data Base	31
4.2.3	Mail Aliases	31
4.2.4	Defining Directories	32
4.3	Console Terminal Operation	32
4.4	Security Considerations	33
4.4.1	Encrypted Passwords	33
4.4.2	Security Procedures	34
4.4.3	More Extreme Measures	34
4.5	Network Printers	35
4.5.1	Printing on a Printer Attached to a Terminal Server	37
4.5.2	Printing to a Remote Printer Queue or a Networked Printer	38
5	System Operation	39
5.1	Shutting Down the System	39
5.2	Booting the System	40
5.2.1	Rebooting the System for Stand-Alone Operation	40
5.3	File System	41
5.3.1	Creating Disk Structures	41
5.3.2	Creating Directories	43
5.3.3	File and Directory Protection	43
5.3.4	Managing Directory Access	44
5.3.5	File System Back Up	45
5.3.6	Periodic Tasks	46
5.3.7	Rebuilding the Boot Structure from Tape	46
5.3.7.1	Less Drastic Measures	47
5.3.7.2	Reinitializing the TOPS-20 File System from Tape	48
5.3.8	Formatting a Disk	57
5.4	Tape Backup	58
5.5	Tape Handling	59
5.6	The OPR Program	59

5.7	Other Periodic Tasks	60
5.7.1	Accounting Data	60
5.7.2	The <code>BS:<SPOOL></code> directory	60
5.7.3	The <code>SERR:ERROR.SYS</code> file	61
6	Running the TOPS-20 UETP Package	63
6.1	Obtaining a Tape Drive	63
6.1.1	Systems without Tape Drive Allocation	64
6.1.2	Systems with Tape Drive Allocation	64
6.1.3	Using a Logical Name for the Tape Drive	66
6.2	Running the Standard Tests	66
6.3	Running Optional Tests	69
6.4	Deleting Temporary Directories	71
6.5	Release the Tape Drive	71
	Appendices	75
A	7-CONFIG.COMD Commands	75
A.1	Defining System-Wide Logical Names	76
A.2	Selecting an Editor	76
A.3	Defining Magnetic Tape Logical Unit Numbers	77
A.4	Defining the Local Time Zone	79
A.5	Specifying Daylight Savings Time	79
A.6	Directory Parameter Setting	80
A.7	Account Validation	81
A.8	Performance Improvements	81
A.8.1	Working Set Swapping	81
A.9	Scheduler Controls	82
A.9.1	Bias Control	82
A.9.2	Traditional Scheduling	82
A.9.3	Class Scheduling	83
A.9.3.1	Class Scheduling by a Policy Program	83
A.9.3.2	Class Scheduling by Accounts	83
A.10	File Archiving and Migration Parameters	84
A.10.1	Archive Tape Recycle Period	85
A.10.2	Migration Tape Recycle Period	85
A.11	Tape Drive Allocation	85
A.12	Accounting Shift Changes	86
A.13	Controlling System Message Levels	86
A.14	Network Information	87
A.15	Designating Off-line Structures	87
A.16	The Login Structure Facility	88
A.17	Preventing Fast Login	89
A.18	Password Management	89
A.18.1	Designating Minimum Password Length	89
A.18.2	Changing Passwords Regularly	89
A.18.3	Disallowing Certain Passwords	90

A.19 Dumping on Non-fatal System Errors	90
A.20 Disabling the Console Terminal for Operator Output	91
A.21 Enabling the Access Control Program	91
A.22 Setting Console Terminal Parameters	91
A.23 Commands not Supported in the TOAD-1 System	92
A.23.1 Assigning MSCP-served Disk Drives	92
A.23.2 Cluster Data Gathering	93
A.23.2.1 Cluster Information	93
A.23.2.2 Cluster Sendalls	93
A.23.3 The Magtape Command	94
A.23.4 The Printer Command	94
A.24 Running SETSPD Interactively	94
B OPR Commands	97
C CheckD Commands	103
D TOPS-20 Start Up Operations	109
D.1 TOPS-20 Initialization	109
D.2 TOPS-20 System Variables	111
E Generating a Tailored Batch and Spooling System	113
E.1 Restoring the GALAXY Source Files	114
E.1.1 Restoring the Galaxy Sources	114
E.1.2 Verify	119
E.2 Running the GALGEN Program	120
E.3 Assembling the New GALAXY System	126
E.4 Shutting Down the Old GALAXY System	128
E.5 Starting up the New GALAXY System	131
E.6 Making Your New GALAXY System Permanent	134
F TDBOOT Command Summary	137
F.1 Macro-console commands	137
F.2 Micro-console commands	173
Index	181

Chapter 1

Site Preparation

This chapter describes the requirements at the location where the TOAD-1 System is to be installed.

1.1 Electrical

The two electrical requirements are that power be supplied to the system and, if the system is to be installed in a computer room, that a means be provided to disconnect electrical power (including battery power) from the system.

1.1.1 Power Connection

The TOAD-1 System can be powered from either 110 VAC 20A or 220 VAC 10A, 50-60 Hz. A separate circuit is recommended for this system, especially if it is powered from 110 VAC. The power inlet on the rear panel of the machine is an IEC-950 connector; mating power cords to plug into a variety of standard electrical outlets are available. We have attempted to supply a cable that will plug into the receptacle type common in the country to which the machine is shipped; other types may be acquired, either from XKL, or from local suppliers.

1.1.2 Emergency Power Off Connection

The TOAD-1 System contains a battery which provides power during interruptions of the regular AC source.

According to Article 645 of the 1996 (U.S.) National Electrical Code (NEC), some locations are classified as “Information Technology Equipment Rooms” i.e., computer rooms. Such locations require a “Disconnecting Means” (informally, an emergency power off button, or “EPO”) which shall disconnect power to all equipment in the room. The disconnecting means shall also disconnect

the supply and output circuits of Uninterruptible Power Supply (UPS) systems installed within the computer room; the disconnecting means shall disconnect the battery from its load¹.

When the TOAD-1 System is installed in a computer room, it must be attached to the Disconnecting Means. The back panel contains a connector for this purpose. That connector should be wired in parallel with any other devices that are connected to the EPO circuit (110/220VAC). The system is furnished with an EPO control cable and with a “T” connector. This connector accepts the EPO cable and it presents the EPO signal on two female connectors. Plug one of the female connectors into the connector on the backpanel; the second connector allows you to extend the EPO circuit to another system in daisy-chain fashion, so that all will shut down together when the Disconnecting Means is activated.² To avoid current overload in the control cables, no more than 10 systems should be attached to a single daisy-chain.

The site should be equipped with control connection to the EPO circuit, brought to an electrical box (on the wall, or as code permits, in the sub-floor). The connector is:

RKF-30 Female plug/female thread bulkhead connector
 Turck, Inc.
 3000 Campus Drive
 Minneapolis, MN 55441

The EPO control cable has three wires. The connector position nearest the notch is chassis ground. Facing the female connector from the connector side (not the wire side) and moving in a clockwise direction from the ground pin, the other two connector pins are EPO source (110/220 VAC) and return (neutral). Actuation of the Disconnecting Means should provide power on the EPO source wire for a sufficient duration so that the connected devices respond. (The circuits supplied by the Disconnecting Means should be powered from a source other than those that it intends to disconnect.)

The TOAD-1 System includes a 4 m EPO control cable, XKL part number 50024-00003, that can connect a system to the wall box, or it can connect one system to another. Cables can be extended by connecting one to another.

1.2 Air Conditioning

The TOAD-1 System dissipates up to 1500W, depending on the number of modules and I/O devices mounted in its cabinet. Adequate air circulation and cooling must be provided to maintain the inlet air temperature in the range 10° C—40° C (50° F—104° F). Airflow through the cabinet is rated at 350 ft³/min (10 m³/min).

If the inlet air temperature rises too high, or if the airflow is restricted, the system will protect itself by shutting down abruptly. An attempt has been made to give the operating system a warning in this case; in some multi-fault scenarios, the system could crash, leading to loss of data. Therefore, adequate environmental control is highly recommended.

¹ Consult local building code requirements.

² Some backpanels contain two EPO connectors; in this case, the male connector accepts the incoming EPO cable, and the female connector is available to extend the EPO circuit to another system.

1.3 Network Environment

1.3.1 Terminal Server

The TOAD-1 System is designed to be part of a networked computing environment. Consequently, (nearly) all terminal connections to the system are *virtual terminals*, or TCP connections. These may originate in several ways: from another computer that supports TCP/IP, from a network terminal, or from a terminal server which provides TCP/IP support for RS-232 devices that do not have that capability.

A network terminal server also will support a wide variety of devices that use an RS-232 connection, either as a command input, or as a status-reporting port. User programs running on the TOAD-1 System can form connections to devices through their terminal server, and can then use these connections to monitor or control the attached devices.

1.3.2 Network Printer

A printer is a special case of an RS-232 device that can be controlled through a TCP connection. If a printer is attached to a terminal server, then any number of computers (not necessarily being all the same type) can gain sequential access to it. The print spooling programs need only to open a TCP connection to the printer's port on the terminal server, pass the data, and close the connection when done. The terminal server then resolves competing requests for service, and effective sharing of the printer is automatic.

Alternately, a printer that supports TCP/IP internally can be attached directly to an Ethernet segment, so that it is available to several systems. In this case, the printer's TCP/IP service resolves competing connection requests, providing sequential access to that printer.

A third option is to equip a printer with an external box that provides the TCP/IP service. Such a device attaches to the Ethernet on one side; it accepts TCP connections there. The printer attaches to the other port; it operates solely on command from the adapter box. In this case, the adapter box resolves contending requests, and the printer is effectively shared.

Finally, print service may be handled by other networked computers (e.g., UNIX systems); the physical connection between a printer and such a machine can be anything supported by that machine's operating system.

1.4 Radio Interference

Notice:

The TOAD-1 System has been tested and found to comply with the limits for a Class A digital device, pursuant to part 15 of the (U.S.) FCC Rules. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment. This equipment generates, uses, and can radiate radio frequency energy and, if not installed and used in accordance with the instructional manual, may cause harmful interference to radio communications. Operation of this equipment in a residential area is likely to cause harmful interference in which case the user will be required to correct the interference at his own expense.

Chapter 2

Physical Set Up

This chapter describes how to unpack your TOAD-1 System and connect it to the world.

You should receive the following items, depending on what you have ordered:

- The system main chassis, in a crate affixed to a pallet.
- The battery, in a shipping carton.
- Software, Manuals, and sundries. This carton contains the software (one DAT tape), one blank DAT tape, manuals (including this one), the system power cord (appropriate to your country), and the EPO cable.
- If the XTU-1 nine-track tape was ordered, there will be two cartons. The large carton is the tape drive and its power cord. The second carton contains the 15 ft (5 m) 8-bit differential SCSI cable, an 18 in (.5 m) 16-bit SCSI differential cable, the 16-bit to 8-bit SCSI converter box, and an 8-bit differential SCSI terminator.

The following tools are required for installation:

- #2 Phillips-head screwdriver
- 1/2 inch socket & ratchet driver
- 7/16 inch nut driver
- 7/64 inch hex driver

2.1 Unpacking

The system is shipped in a crate affixed to a pallet. We recommend that the entire crate be placed near the installation site before unpacking. (If your system includes a 9-track tape drive, it will arrive in a separate box.)

To unpack the system, remove all 5/16 inch bolts (ratchet driver and 1/2 inch socket) that hold the top and sides of the exterior packing box together. This will expose the interior packing crate, resting on the shipping pallet. Slide the crate off the pallet. Disassemble the interior packing crate by cutting the plastic strapping. This will leave you with a nice, shiny TOAD-1 System. Move the system cabinet to its final position. You may retain the packing crate in storage, in case the system must be repacked and shipped to another location.

2.2 Assembly

2.2.1 Mounting

If the system is free-standing, no additional assembly is required. (The sides of the cabinet should be blue.) Place the system in the desired location, and go on to the next step.

If the system is to be mounted in a rack, this should be done first, along with any peripheral equipment that is to be installed in the same rack. (In this case, the sides of the box should be aluminum-colored, and rack-mount ears should be visible.)

This system is quite heavy when fully assembled. If it is to be mounted in a rack, XKL recommends that the system cabinet be supported on angle-bracket rails rated for a 150 pound (70 Kg) load. The rack-mount ears are not intended to bear the full load of the cabinet; they should be used to secure the system to the front rails of the rack.

Due to the weight of the system, some arrangements of equipment in a rack are susceptible to tipping over. Equipment mounted in a rack should be arranged so as to provide a low center of gravity. Special precautions should be taken to secure equipment located in seismically active regions.

2.2.2 Battery Installation

The battery is shipped in a separate package.

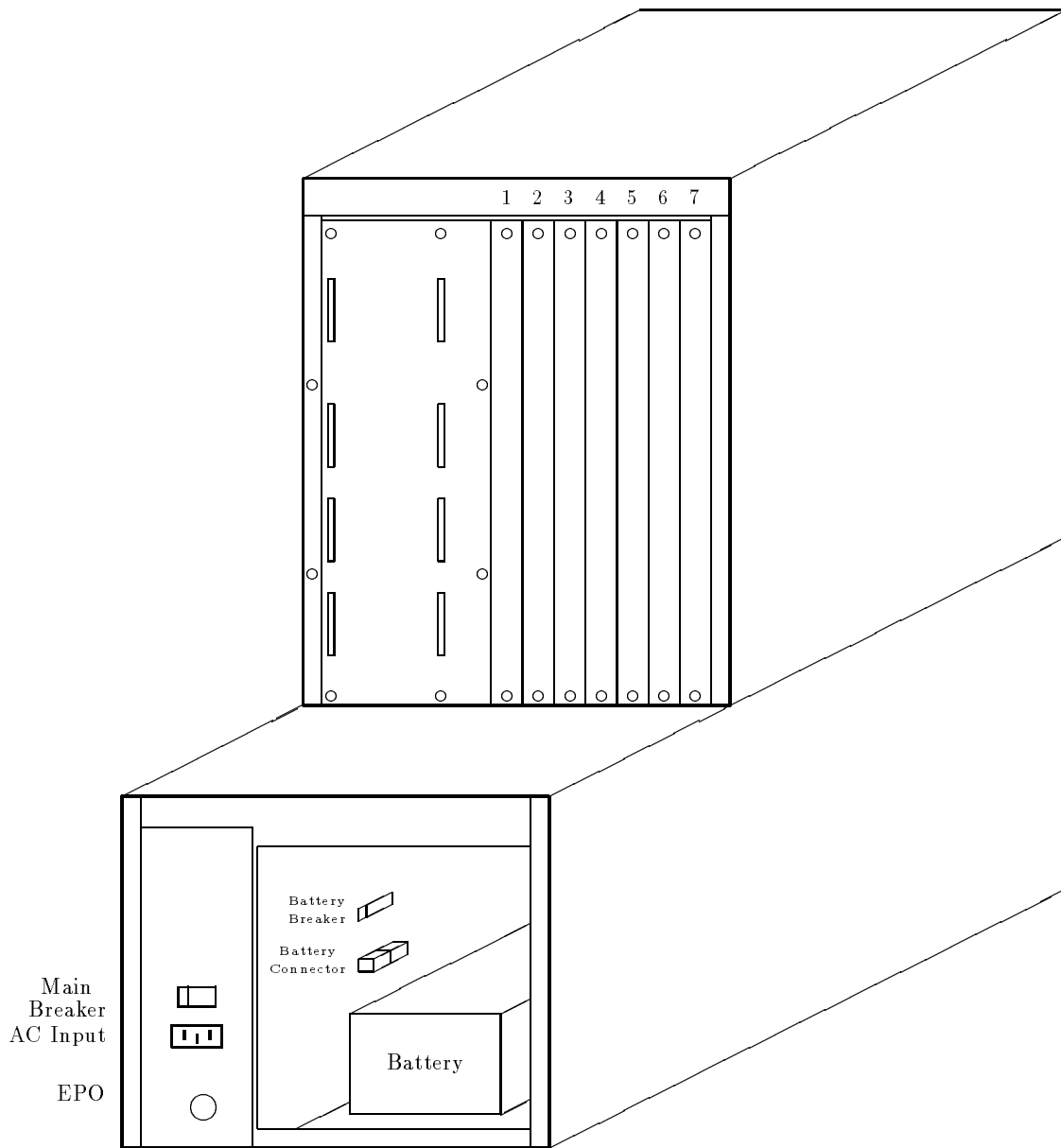
Safety Warning

Turn off the AC power at the system's main breaker.

Disconnect the AC power cord.

Remove rings, bracelets, watches, and any other metallic objects from your fingers, hands, and wrists. Also remove loose necklaces, neck chains, dangling earrings, and any other jewelry that might come into contact with parts of the machine. There is quite enough energy stored in the battery to cause a serious burn if it should discharge through any such object.

Figure 2.1: TOAD-1 System Rear View



(Battery Compartment is Open; Upper Rear Cover is Removed)

Facing the back of the machine (where the power cord attaches), the battery compartment is behind the perforated cover panel that occupies the right two-thirds of the lower section of the machine (to the right of the panel with the main AC breaker and power inlet). Remove the screws that secure the back cover panel; remove the cover panel. Refer to Figure 2.1 which depicts the system with the battery compartment open.

The internal battery breaker is located on the left wall of the compartment, on the partition that separates the battery compartment from the AC power inlet and power supply compartment. Turn off the internal battery breaker by pulling its handle towards you; the numeral “0” and the word “OFF” should be visible on the far side of the breaker handle when the breaker is off.

The battery is heavy; handle it carefully. Orient the battery so that when the battery is installed, the battery cables will be close to the rear of the system (i.e., cables toward you). There are four threaded posts protruding upwards from the bottom of the battery compartment. Remove the flange nut from each of these posts. Put the battery in the compartment, aligned so that the slots on the base of the battery are over the threaded posts. Use the flange nuts to secure the battery to the bottom of the battery compartment.

Connect the battery cable to the connector located on the left wall of the battery compartment, just below the battery breaker. Match the colors of the battery cable to the colors on the connector. Turn the battery breaker on: push the handle away from you; the numeral “1” and the word “ON” should be visible on the breaker handle when the breaker is on.

The main system cooling fan is located at the top front (away from you) of the battery compartment. Verify that your installation does not interfere with the movement of the fan assembly. (The fan blades should turn easily by hand. Due to the construction of the fan motor, the fan will demonstrate a preference to stop at particular positions.)

Reinstall the back cover panel, securing all the screws. Reconnect the AC power cord.

2.3 Power Connections

2.3.1 Mains Power

The power connection is an IEC-950 male connector, mounted in the lower left in the back of the system (facing the rear of the cabinet). Connect the power cord to this plug.

Plug the system into either a 110—120 VAC 20A circuit or to a 208—240 VAC 10A circuit. The power system automatically detects the line voltage and adjusts itself. As noted in Section 1.1, XKL recommends that this system be connected to a separate circuit, especially if the system is to operate on 110—120 VAC.

The main circuit breaker is both the system power on/off switch and a protective device. This breaker switches both AC and DC power. Overload on either AC or DC will trip this breaker. It is located on the rear of the unit near the AC power connection, protected from accidental actuation by a sheet-metal tab.

A second circuit breaker, accessible inside the battery compartment, interrupts only DC power from

(and recharging power to) the battery. If this breaker is left off, the battery will not be available to keep the system running when line power fails, nor will the battery recharge. This breaker is tripped by an EPO event (see below); it must be reset manually after an EPO event.

2.3.2 Emergency Power Off

Provided the system is properly connected into a computer room Emergency Power Off (EPO) circuit, as described in Section 1.1.2, actuation of the room EPO switch will cause the battery circuit breaker to trip. (Presumably, the operation of the EPO switch will also remove AC power from the system at the same time.) This breaker must be reset manually, after removing the cover from the battery compartment, as described in Section 2.2.2. The breaker trip coil is set for 65 VAC and it disconnects when the breaker trips. Thus, the EPO circuit will operate correctly at any voltage from 90 VAC to 240 VAC without overloading the breaker coil.

2.4 External Device Connections

Connections to SCSI buses, Ethernets, and the console terminal are protected by a sheet metal cover over the upper half of the rear of the cabinet. To gain access to these connections, remove the three screws securing the cover to the cabinet; then slide the cover straight back, lifting it slightly. Replace the cover after making the connections and securing cabling to external devices.

2.4.1 Console Terminal

The console terminal port is a DB25 plug (male) wired as a DTE (Data Terminal Equipment) connection, mounted on the back panel of the XKL-1 module; the pin assignments are shown in Table 2.1. To connect to a local terminal (also a DTE), one needs a null modem or correctly wired cable.¹ This cable may be strain-relieved by securing the cable to one of the wire-tie anchors installed in the lower rear panel.

The port characteristics are fixed at 9600 baud, 8 data bits (no parity), 1 stop bit.

2.4.2 Auxiliary Console

The auxiliary terminal port is presented as a DB25 plug, wired as a DTE (Data Terminal Equipment) connection. The (standard) signal assignments are shown in Table 2.1. To connect this port to a modem (a DCE, Data Communications Equipment), one needs a modem cable.² As noted above for the console port, attachment of the auxiliary port to another DTE requires a null modem or a special cable. Port serial characteristics are the same as for the console port.

¹ The correct cable connects pin 2 of the console port to pin 3 of the terminal, and pin 3 of the console port to pin 2 of the terminal.

² Modems vary, so some experimentation may be required. Start with a cable that is wired "straight through", i.e., pin 1 to pin 1, pin 2 to pin 2, etc.

Pin	Function	Console	Aux
1	Frame Ground	*	*
2	TxD (from XKL-1)	*	*
3	RxD (to XKL-1)	*	*
4	RTS (from XKL-1)		*
7	Signal Ground	*	*
8	CD (to XKL-1)		*
20	DTR (from XKL-1)		*
22	RI (to XKL-1)		*

Table 2.1: RS-232 Connector Pin Assignment

The auxiliary terminal port functions only when it is enabled by a console command; see Section 4.3.

2.4.3 SCSI Devices

The TOAD-1 System will be delivered with SCSI cables and “target identification” numbers already set up in a working configuration. The following material is for your reference in the event that you plan to change the configuration of SCSI buses and devices. If your configuration includes the XTU-1 reel-to-reel tape drive, you may want to familiarize yourself with the description of XTU-1 cabling in Section 2.4.3.4

All system disk and tape devices must conform to the SCSI-2 (Small Computer Systems Interface) standard. Since this “standard” allows for a number of interface choices which do not interoperate, XKL has selected an actual standard from those allowed: a 16-bit differential bus from the controller ports. A bus runs from the controller port to a number of 16-bit differential devices, after which it may be narrowed to 8-bit differential. Once the bus width has been reduced, only 8-bit devices can be attached farther from the controller.

Each XRH-1 controller presents four fast, wide, differential SCSI-2 buses. In principle, any device meeting the SCSI-2 specification could be connected; however, operating system support is limited to those devices that we have fully tested. At present, the operating system supports disk drives from several vendors, one vendor’s reel-to-reel tape drive, one vendor’s DAT drive, and one vendor’s DLT cartridge tape drive. There are some peculiarities unique to 36-bit computers and others unique to the XRH-1: *customers should consult with XKL technical staff before attempting to connect devices that are not recognized by the system and its software.*

2.4.3.1 Connectors

All connectors on the XRH-1 are high-density, 68-pin, SCSI-2/3 connectors, often referred to as P-connectors. This connector supplies all the signal and control lines to implement the differential 16-bit SCSI-2 bus.

Some external devices, notably reel-to-reel tape drives, are available with an 8-bit electrical interface only. The most common connector for these devices is a low-density, 50-pin, “micro ribbon”

connector. (Beware: this connector is also used in many SCSI-1 applications, but most SCSI-1 cables are single-ended and these will not work properly with differential SCSI devices.) To make the transition between different connectors and different bus widths, XKL has available an adapter box (XKL part number 30008-00001), which may be secured to the rear panel of the system cabinet. A system ordered with a reel-to-reel tape drive will come equipped with this adapter box and necessary cables.

2.4.3.2 General SCSI Cabling

Each SCSI bus starts at the XRH-1 (electrical termination is provided there). Cables that serve internal devices loop into the I/O section of the unit, visit devices there, and reappear on the back panel of the I/O section; at that point, a terminator must be installed, or the bus must be extended to external devices (where it will terminate). A cable serving only external devices attaches directly to the XRH-1 and runs to the first device. All buses must be electrically terminated at their ends. If the last device does not have internal termination, the bus must be terminated with a differential terminator. A 16-bit bus must be terminated with a 16-bit differential terminator (XKL part number 50026-00001). If the bus travels through the 16-bit to 8-bit SCSI converter box, the 8-bit end of the bus must be terminated by the last device or by an 8-bit differential terminator (XKL part number 50000-00001).

Strain relief anchors for external bus cables are mounted on the lower rear panel. We recommend securing all external cables to these anchors with cable ties.

2.4.3.3 DAT Drive Cabling

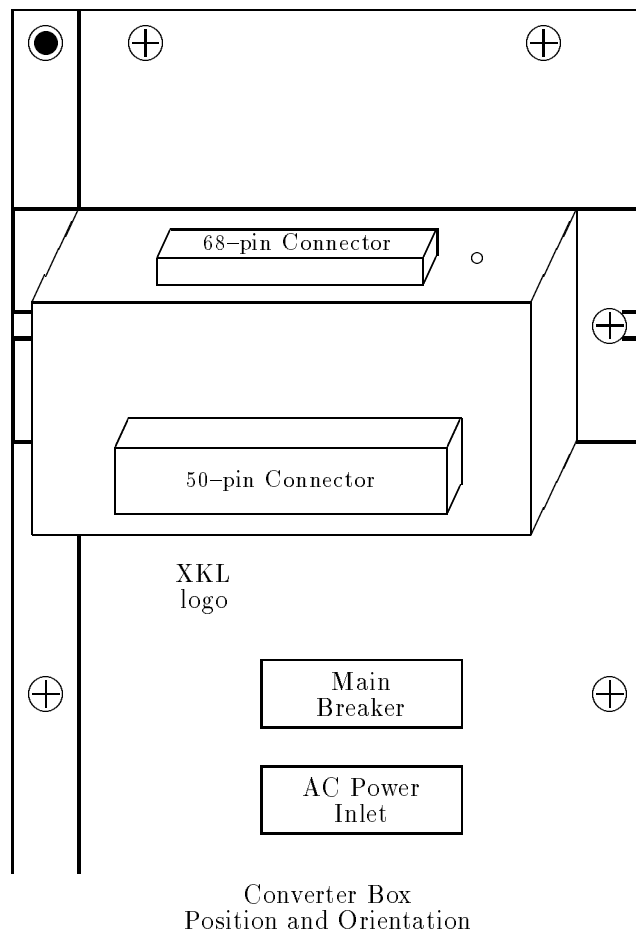
The (internal) DAT drive is available with a single-ended SCSI interface only, so a bus converter is located inside the I/O section of the cabinet to change the differential bus to a single-ended one. This single-ended SCSI bus comes out of the converter and goes to the DAT drive; it is terminated on both ends. There are four 50-pin connectors on this internal bus cable, intended for additional internal DAT drives that may be mounted in the chassis. It is not intended that this single-ended bus be used for any devices external to the system frame.

2.4.3.4 XTU-1 Cabling

The XTU-1 reel-to-reel (9-track) tape drive is available with a narrow (8-bit) differential SCSI electrical interface only. The XTU-1 includes the cables, the 16-bit to 8-bit converter box, and the bus terminator necessary to connect the XTU to the system.

Install the 16-bit to 8-bit differential SCSI converter box (p/n 30008-00001). The converter box mounts on the exterior rear of the machine, across the upper portion of the power inlet panel, just above the "XKL" logo, as depicted in Figure 2.2.

Remove the uppermost screw from the right edge of the power inlet panel; loosen the screw opposite that one at the left edge of machine frame. Orient the converter box so the 68-pin connector points up and the 50-pin connector points towards you. (Note: the 50-pin connector is actually larger

Figure 2.2: XTU 16-bit to 8-bit SCSI Converter Box

than the 68-pin connector.) Slide the “ear” of the converter box under the loosened screw, replace the screw that was removed, and tighten both screws.

Attach the tape drive to the SCSI bus as follows: use the short 16-bit differential SCSI cable from an XRH-1 port³ to the converter box. Use the long 8-bit differential SCSI cable from the converter box to the tape. (Use the strain relief anchors, mounted on the lower rear panel, to secure the external bus cable with cable ties.)

The tape will have two connectors for SCSI cables: plug the long cable into one connector and the 8-bit differential SCSI terminator into the other; it doesn’t matter which is which at the tape.

2.4.3.5 The SCSI “Standard”

CAUTION: Not everything that is called “SCSI” is electrically compatible with the particular bus types that XKL supports. Most SCSI devices are single-ended. In the TOAD-1 System, most internal and all external SCSI devices are differential, for increased data transfer reliability. Single-ended terminators and many plausible-looking cables are not compatible with the TOAD-1 System. XKL can supply a variety of cables and terminators that are of the correct type. Based on the descriptions above, competent local vendors may be able to supply cables and terminators that are compatible.

2.4.3.6 SCSI Target ID Selection

In general, XKL recommends that devices be configured to spread activity as evenly as practicable over the available buses.

On each SCSI bus, the target ID for each device (including the XRH-1) must be unique. After selecting which devices are to be attached to which SCSI bus, the person configuring the system must assign target IDs for each device and configure those devices accordingly. The only consideration that recommends one ID over another is that 8-bit devices are limited to having IDs in the range 0 through 7, and 16-bit devices can “talk” only to an initiator whose ID is in the range 0 through 7. For this reason, we recommend that the XRH-1 ID be set to zero; further, on a bus that contains both 8-bit and 16-bit devices, the 16-bit devices ought to be assigned IDs in the range 8 through 15.

The default target ID for the XRH-1 is 0 on each SCSI bus. This can be changed by the following console command (see Section 3.2):

```
XKL-1%Define Scsi-Id slot bus_number target_ID RET
```

The target ID for each disk and tape unit must be set according to its manufacturer’s specification. Since there are as many schemes as device types, the best plan is to follow the manufacturer’s directions.

Systems delivered from XKL will have device target IDs set so that all are usable. Individual

³If no XRH-1 port is available, remove a 16-bit SCSI terminator from a SCSI bus and install the short 16-bit cable instead of the terminator.

peripherals ordered from XKL, to be added to an existing system, will come with appropriate directions.

2.4.4 Network

Four independent connections to the IEEE 802.3 standard Ethernet are supported on each XNI-1. Each connection can be made via twisted pair (10-Base-T) or via an AUI device.

The AUI connection is used when a port is connected to a Media Access Unit (MAU) either directly or with an AUI cable. Examples of such MAUs are multiport repeaters and transceivers for coaxial cables.

For a 10-Base-2 (thin coaxial cable) connection, the MAU is customarily attached to the 15-pin AUI connector at the back of the XNI-1 board. Coaxial cable is run between the MAU and two other connection points.⁴

An RJ-45 female connector on the backpanel of the XNI-1 supplies a 10-Base-T connection. Twisted-pair cable must be provided, running between this connector and the 10-Base-T hub providing network access. This hub is the MAU in this case; it is usually located at some distance from the system.

Network address assignment and other configuration steps are discussed in Section 4.1.2.

2.5 Indicator Lamps and Jumpers

The system has three lamps on the front panel to indicate the operating status of the machine:

Green indicates that DC power is on.

Red The combination red/green indicates that the system is running on its battery, because of a loss of AC power.

The red lamp lit by itself indicates that the internal temperature has been exceeded (or that the cooling air path was obstructed), and that the machine has shut itself down.

Yellow indicates the level of activity on the system. It has been carefully calibrated at the factory so that the lamp is off when the system is idle, and it is at full brightness when there is so much excitement that no job is getting much service.

In addition to the lamps on the front of the machine, there are indicator lamps visible through the module cover panel of each of the different system boards. The approximate behavior of each of these lamps is described in the following subsections.

⁴Or terminate one end of the coax at the MAU and run coax to one other connecting point.

2.5.1 XKL-1 Processor

Above each RS-232 port connector is a green lamp that is turned on when that port is activated. The indicator for the console port will normally be on. The indicator for the auxiliary console port will be on only when that port is activated; see Section 4.3.

In the lower part of the XKL-1 module board cover, a single red lamp, labeled “Fault”, indicates the occurrence of a microcode or private memory (MemA) parity error. Either of these errors is fatal; the XKL-1 will either restart or halt⁵, depending on whether a jumper block is installed at J3 on the board. Below this lamp are four green lamps that are used to signal the progress of self-tests, performed at power-on. These lamps are off during normal system operation.

A jumper block is normally installed at J3, to cause restart in case of error. This jumper is located on the component side of the PC board, between the red LED and a 50-pin header, very close to the module cover panel. If this jumper block is removed, the XKL-1 will halt on the occurrence of either a microcode or MemA parity error.

2.5.2 XRH-1 SCSI Bus Controller

The XRH-1 has five indicator lamps. The red “Fault” lamp is located between the Port 0 and Port 1 SCSI connectors. There are four green lamps, one located above each SCSI bus connector. Each lamp reflects the status of the corresponding SCSI bus.

After initialization, during normal operation, the “Fault” light should be off; each green lamp is turned on when the corresponding bus is on-line and idle. While a transaction is in progress on a bus, the corresponding lamp is turned off. Thus, a blinking green lamp indicates SCSI bus activity. Even when TOPS-20 is idle, these lights should blink as the operating system scans the devices to see if their status has changed. After initialization, if the “Fault” lamp is off, a green lamp that remains off indicates either a bus that has been taken out of service or one that is “hung”. (SCSI devices on a malfunctioning bus can be moved to working buses until the hardware can be fixed or replaced. If it is necessary to reconfigure the SCSI buses, it is best to do so with the system shut down and power off.)

During the XRH-1’s power-up (and hard reset) sequencing, blinking green lamps indicate that self-testing is in progress; the fault light may be on for 30 seconds or so during this initialization and self-testing.

The fault lamp is lighted continually if the self-test fails, or if the XRH-1 has released “need power” during a power failure. If the fault lamp blinks, the board is malfunctioning. If the fault light stays on, a fatal XRH-1 error has been detected, or a power failure is pending.

⁵In this context, “halt” means that execution of the microcode will stop (in contrast to the PDP-10 HALT instruction that stops the macro-code program). When the microcode is stopped, there are no console functions whatsoever.

2.5.3 XNI-1 Ethernet Controller

The red “Fault” lamp indicates a microcode parity error on the XNI-1. If it blinks continuously or stays on, the module is faulty.

Each port is decorated with 5 lamps to indicate the status of that port. These lamps all turn on briefly when power is applied and when the operating system resets the controller.

On each port, the “Link Pol” lamp should be on. If a cable is connected to the RJ-45 connector and to a 10-Base-T hub, then when traffic appears on the cable, this lamp may go out, indicating that the polarity of the connection is reversed. If this happens, recheck the 10-Base-T wiring. The XNI-1 automatically corrects for this polarity reversal, so data reception will still work normally.

The operating system lights the “Port OK” lamp for each port to which it expects a network to be connected. The lamp will not light from the time power is turned on until the first time TOPS-20 starts operation.

The “Link OK” lamp should be on when a live 10-Base-T cable is plugged into the RJ-45 connector, but off for a thin-wire/MAU connection, or if a cable is disconnected at its far end.

The “Tx” lamp lights when data is being sent from the XNI-1.

The “Rx” lamp lights when data is arriving at the XNI-1. An active network will cause the “Rx” lamp to flicker, regardless of whether or not the TOAD-1 System is the target of the traffic.

2.5.4 XMG-1 Memory

There is a single green lamp on this module. It is turned on when initialization software has cleared memory and has determined that the module functions correctly. This lamp indicates that the module is available to the operating system (“on line”).

Chapter 3

Starting the System

This chapter describes what you should see on the console terminal when first starting a TOAD-1 System. *Console commands*, i.e., commands that can be typed on the console and auxiliary console only, to boot the operating system are described.

Before turning the power on, be sure that the installation steps described in Chapter 2 have been completed. Connect a console terminal to the TOAD-1 System's console port, and turn on the console terminal. Make sure the terminal parameters are set as described in Section 2.4.

3.1 Initial Boot

Assuming a console terminal is connected to the TOAD-1 System when you first apply power, you should see something resembling the following:

```

System Processor (XKL-1, 1995) Ver - 1133
TDBoot version 1.2(25)-1
System ID 32., system options 0
Microcode version 2, options 200000
All system startup parameters are enabled.
Testing cache
Testing pager
Polling and verifying bus configuration
Memory configuration and test:   The following line may take 5 minutes
Slot 3. XMG-1 (32MW), Testing: SDdAa, On line at LPN 0
Starting not-ready devices
Device configuration:
Slot      Device
  1.      1./ 1.   : Seq. Access   "HP           ", "C1533A           ", "9503"
                Serial number: N/A

```

```

Tape MTA0:
3./ 8.   : Direct Access "SEAGATE ", "ST15150W      ", "0017"
Serial number: 00658476
Structure PS:(BS), 0. of 1. unit
[Begin AUTO-BOOT delay, type ^C to abort or any other key to proceed now]

```

Note: The memory testing takes about 5 minutes (for each 32 M word module). As long as the program displays the “spinner”, which is a succession of characters that appear to spin, progress is being made. There will be a short delay after the letter “S” is printed following “Testing:”. There will be a one- to two-minute delay following each of the letters “D”, “d”, “A”, and “a”.

Error: If you see no timeout at all, suspect the console terminal, its parameters, and its communications wiring.

If the timeout indicates a problem with the configuration, type <CTRL/C> to stop the boot process and attempt the command

```
XKL-1%Define Configuration * 
```

followed by

```
XKL-1%.i     Note the period preceding the “i”
```

If these fail to correct the problem, seek help from XKL.

This timeout displays various system resources as the CPU discovers them: where (and how much) memory is present, and the layout of each SCSI bus that has responding disk or tape drives on it. (Devices switched off or unplugged from power will not be shown, since they do not respond.)

About one minute after the message “Begin AUTO-BOOT delay”, the system will go on to boot itself. The delay allows the operator an opportunity to interrupt the automatic process and direct the system to a different activity.

If the process is interrupted (by typing <CTRL/C>), the system will respond with the prompt **XKL-1%**, which shows that the micro command parser is ready for a command; the % signifies that the PDP-10 instruction set processor is halted.

The console command **Boot** loads and starts the operating system or a stand-alone program. In the absence of a structure or filename as a parameter to **Boot**, the *default boot string* will select the file name; in the absence of a default boot string, the file named <SYSTEM>**MONITR.EXE** will be sought on the TOPS-20 *auto-boot structure*. Another structure and/or filename may be specified as a parameter to the **Boot** command; the replacement name string is effective for that command only. The default boot string can be specified as the parameter to the console command **Define Boot** as shown here:

```
XKL-1%Define Boot file_specification 
```

Thereafter, the *file_specification* string will be used in subsequent **Boot** commands.

The auto-boot structure is preassigned when systems are shipped from the factory; it may be changed using the CHECKD program. Details are in Appendix C.

If there is no auto-boot structure, you must specify a structure name in the `Boot` command.¹ If there is no bootable structure at all, you can boot from tape, by supplying an explicit boot target (e.g., `mta0:`) The file to be booted must be the next one on the tape (usually the first).

If the auto-boot process is allowed to continue, the typeout will look similar to that depicted below. The underlined portions represent responses that you must type on the terminal.

Note

The customer initially receives an unconfigured system which will type various error messages during initialization. These error messages are indicated below. (Your mileage may vary.) Hopefully, the error messages will indicate the name of the file that must be changed to correct the problem. Please follow the configuration and customization tasks described in Chapter 4 to correct these problems.

```
[Reboot requested]
[Using default string: ""]
[Structure PS:]
[Directory <ROOT-DIRECTORY>]
[Directory <SYSTEM>]
[File PS:<SYSTEM>MONITR.EXE.1]
[Loading .....]
[File entry vector is at 1,,161, length is 5]
[PDVA at 1,,141077]
[Filled through LPN 2126]
[Starting at 1,,161]
[PS mounted]
%Error in interface number in SYSTEM:INTERNET.ADDRESS file
nea^#<n>, nnn.nnn.nnn.nnn, ethernet
[Loading Internet host names]
%Bad host number format in SYSTEM:HOSTS.TXT
net : x^xx.xxx.0.0 : YourNetworkName :::
[OK]
% LOADGW: Unknown keyword in file:<SYSTEM>INTERNET.GATEWAYS.1
ALWAYS XXX.XXX.XXX.XXX
^
SETSPD: Processing SYSTEM:7-CONFIG.CMD
TOAD-1 System, TOPS-20 Monitor 7(102113)-1
Monitor built Thu 29-February-1996 8:20AM
Network Time Setter
? JSYS ERROR: No number for that host name
  (This will be corrected when the internet parameters are set up.)
Enter current date and time:
```

¹If there is no auto-boot structure, TOPS-20 will also need to be told the name of the structure to use as it starts up.

Enter the date and time in the form “mm-dd-yy hhmm” or as shown in the example. Type to tell TOPS-20 to go on. TOPS-20 will repeat the date back to you and ask for confirmation. If you have entered the correct date, type “Y” and ; otherwise, type “N” and and TOPS-20 will ask for the date and time again.

```
Enter current date and time: 1 March 96 16:25 
You have entered Friday, 1-March-1996 4:25PM,
is this correct (Y,N) y 
Why reload?
```

The response to “Why reload” is entered into the system error file, `BS:<SYSTEM-ERROR>ERROR.SYS` as part of the reload record. The first word of the response must be one of a small number of keywords; for a list of valid responses, you may type “?” and . For the time being, we shall type “New install system”.

Notes

If you do not respond to the “Why reload” question within 60 seconds, the system assumes “Other” and continues the startup procedure.

If you type an unrecognized response to “Why reload”, TOPS-20 will complain and prompt you again.

```
Why reload? new install system 
Run CHECKD?
```

Next, TOPS-20 asks if you want to run the CHECKD program. If you respond “Yes” to this question, CHECKD will be started and it will check the disk to see if it can find any “lost pages”. The functions of CHECKD are further explained in Section 5.3.6 and in Appendix C. Although it is sometimes worthwhile (and sometimes mandatory) to run CHECKD, for this time we shall respond “No” and type .

Note

If you do not respond to this question within 60 seconds, TOPS-20 will answer the question “No” and continue the startup procedure.

```
Run CHECKD? no 
```

```

1-Mar-96 16:25:09 DDMP: Started
1-Mar-96 16:25:10 SETSPD: Checking for crash DUMP
SYSJOB 7(78) started at 1-Mar-96 1625
1-Mar-96 16:25:28 SYSJOB: Job 3: @ LOGIN OPERATOR
1-Mar-96 16:25:28 SYSJOB: Job 1: @ LOGIN OPERATOR
1-Mar-96 16:25:29 SYSJOB: Job 4: @ LOGIN OPERATOR
1-Mar-96 16:25:29 SYSJOB: Job 5: @ LOGIN OPERATOR
1-Mar-96 16:25:30 SYSJOB: Job 6: @ LOGIN OPERATOR
1-Mar-96 16:25:30 SYSJOB: Job 2: @ LOGIN OPERATOR
1-Mar-96 16:25:32 SYSJOB: Job 5: Host@ ENABLE
1-Mar-96 16:25:32 SYSJOB: Job 5: Host! RUN SYSTEM:RAISE-QUOTA
1-Mar-96 16:25:32 SYSJOB: Job 6: Host@ ENABLE
1-Mar-96 16:25:32 SYSJOB: Job 6: Host! RUN SYSTEM:MAILST.EXE
1-Mar-96 16:25:33 SYSJOB: Job 2: Host@ ENABLE
1-Mar-96 16:25:33 SYSJOB: Job 5: Host! RUN SYSTEM:LPTSPL/BACK
1-Mar-96 16:25:33 SYSJOB: Job 2: Host! RUN SYSTEM:SMTJFN /BACK
1-Mar-96 16:25:33 SYSJOB: Job 4: Host@ ENABLE
1-Mar-96 16:25:34 SYSJOB: Job 1: Host@ ENABLE
1-Mar-96 16:25:34 SYSJOB: Job 1: Host! RUN SYSTEM:SETNAM
1-Mar-96 16:25:34 SYSJOB: Job 4: Host! RUN SYS:MMAILR
1-Mar-96 16:25:34 SYSJOB: Job 1: Host! GET SYS:OPR
1-Mar-96 16:25:34 SYSJOB: Job 3: Host@ ENABLE
1-Mar-96 16:25:34 SYSJOB: Job 3: Host! CD DOMAIN:
1-Mar-96 16:25:36 SYSJOB: Job 3: Host! RUN RESOLV
1-Mar-96 16:25:36 SYSJOB: Job 5: Host! RUN SYSTEM:LPTSPL/BACK
1-Mar-96 16:25:37 SYSJOB: Job 2: Host! RUN SYSTEM:NETSRV
1-Mar-96 16:25:37 SYSJOB: Job 5: [LPTSPL0]
1-Mar-96 16:25:37 SYSJOB: Job 1: Host! DEPOSIT SHUTUP -1
1-Mar-96 16:25:38 SYSJOB: Job 5: Host! RUN SYSTEM:TCPSPL/BACK
1-Mar-96 16:25:38 SYSJOB: Job 5: Host! RUN SYSTEM:TCPSPL/BACK
1-Mar-96 16:25:38 SYSJOB: Job 5: [TCPSPL0]
1-Mar-96 16:25:38 SYSJOB: Job 5: Host! RUN SYSTEM:TCPSPL/BACK
1-Mar-96 16:25:40 SYSJOB: Job 5: [TCPSPL1]
1-Mar-96 16:25:40 SYSJOB: Job 5: Host! RUN SYSTEM:TCPSPL/BACK
1-Mar-96 16:25:40 SYSJOB: Job 5: [TCPSPL2]
1-Mar-96 16:25:40 SYSJOB: Job 1: [Shared page made private]
1-Mar-96 16:25:40 SYSJOB: Job 1: Host! START
1-Mar-96 16:25:41 SYSJOB: Job 1: OPR> SILENCE
1-Mar-96 16:25:41 SYSJOB: Job 1: OPR> TAKE SYSTEM:SYSTEM.CMD
1-Mar-96 16:25:42 SYSJOB: Job 1: OPR> TAKE SYSTEM:PRINTER.CMD
1-Mar-96 16:25:42 SYSJOB: Job 1: OPR> PUSH
1-Mar-96 16:25:43 SYSJOB: Job 1: End of COMAND.CMD.1
1-Mar-96 16:25:43 SYSJOB: Job 1: @ ENABLE
1-Mar-96 16:25:43 SYSJOB: Job 1: ! ^ESET LOGIN ANY
1-Mar-96 16:25:43 SYSJOB: Job 1: ! POP
1-Mar-96 16:25:44 SYSJOB: Job 1: OPR> WAIT 60
1-Mar-96 16:26:43 SYSJOB: Job 1: OPR> WAIT 60
1-Mar-96 16:27:42 SYSJOB: Job 1: OPR> RESTORE

```

Once the system comes up to this point, you can log in as Operator. Type <CTRL/C> to start the EXEC; the EXEC will print a brief banner line and prompt with the “@” character. Type the **Login** command:

```
@login operator xkl-toad RET
```

As you type this command, the initial password, *xkl-toad*, will not echo to the output; your terminal should look like the following:

```
Your TOAD from XKL, TOPS-20 Monitor 7(102113)-1
@login operator
End of COMAND.CMD.1
Host@
```

Security Note

We recommend that you change the password for Operator *immediately*. A full discussion of the security implications is in Section 4.4, page 33; that section also shows how to change a password. **Please remember your new password.**

A variety of site-specific alterations **must** be made to integrate the system with your network environment and to tailor it to the user community. These are described in Chapter 4.

3.2 Other Boot Operations

Up to now in this section, we have assumed that only one terminal (physically connected to the console RS-232 port) was active; this led to some ambiguities in referring to the commands typed, since all were typed at the same terminal.

Throughout this document, we attempt to distinguish *console commands* from others. A console command will be recognized only when it is typed in response to the prompt **XKL-1>** or **XKL-1%** on the console (or the auxiliary console).

The console commands are documented in the *TOAD-1 System Architecture Reference Manual*, in the appendix on TDBOOT. These commands are used to boot the system, to define various operational parameters, mostly related to the hardware installed, and to perform diagnostics to isolate a fault in a malfunctioning system. Some console commands can be issued while TOPS-20 is running. To get the attention of the console command interpreter while the system is running, type <CTRL/\> on the console; this should produce the prompt **XKL-1>**. Then issue a console command. To give the console terminal back to TOPS-20, type the console command:

```
XKL-1>Continue RET
```

For some console commands, the TOPS-20 system must be stopped. Such commands, if attempted at the `XXL-1>` prompt, will result in an error message; in that case, the running program (TOPS-20) can be stopped by the command:

```
XXL-1>Halt RET
XXL-1
```

Once the program is stopped, all console commands are possible.

Caution

Console commands that are illegal while the system is running are likely to destroy the “state” of the computer and make it impossible to continue the operation that was interrupted. Where possible, make sure that timesharing operations have been shut down in an orderly way before giving any such command. If timesharing operations are active, the system can be shut down by the procedure documented in Section 5.1, page 39.

3.3 Abnormal Conditions

This section can not be an exhaustive treatment of everything that could go wrong. This section discusses some of the anomalies in system startup that might be caused by maintenance activities.

3.3.1 Board Swapping

In some maintenance and diagnostic activities, the system circuit boards may be moved to different backplane slots. The system is designed to operate properly regardless of the permutation of its circuit boards. However, the bootstrap program, `TDBOOT`, will notice that the boards have been moved, and it will not automatically boot the system.

If you get a message from `TDBOOT` announcing that it has detected a change in the configuration of the boards, you must give the console command

```
XXL-1%define configuration * RET
```

to refresh `TDBOOT`'s understanding of which board is where. Thereafter, the system should boot without unusual manual intervention.

3.3.2 Backplane Swapping

The system is electronically identified by a ROM installed on the system backplane. If the backplane is changed without moving the ID ROM from the old backplane to the new one, the system will not boot properly because it can't find the auto-boot disk.

In this case, the best thing to do is to change the ID ROM on the backplane.

Otherwise, boot the system by manually specifying the structure from which to boot. As TOPS-20 starts, it will be unable to find the right structure automatically, so it will ask for a structure name. When TOPS-20 is running, use the CHECKD program to redefine the auto-boot system ID for your boot structure; see Appendix C.

3.3.3 Disk Structure Problems

See Section 5.3, page 41.

Chapter 4

System Administration

This chapter covers some of the most important aspects of customizing TOPS-20 to serve your user community. Topics covered include the configuration of TOPS-20 for your network environment, the creation of login accounts for users, and the creation of file-storage directories. This chapter also discusses security considerations in the network environment, and the configuration of printers on the network.

4.1 Configuring TOPS-20

4.1.1 System Definitions

The logical name **BS:** represents the name of the *boot structure*. The boot structure serves two different purposes:

- The boot structure is automatically recognized by **TDBOOT** as the structure from which to load the TOPS-20 monitor.
- TOPS-20 recognizes the boot structure, also known as the *system structure* as the first structure to mount, as the structure to swap on (for which reason this structure is also known as the *swapping structure*), and as the structure from which to load programs and command files for system initialization (e.g., **SETSPD** and **SYSTEM:7-CONFIG.COMD**).

The boot structure is identified with two marks: it must be flagged as an “auto-boot” structure, and it must contain a system ID that matches the current system’s identification number. Moreover, to be recognized by **TDBOOT**, it must be the only structure marked this way. In order to be recognized by TOPS-20 as the system structure, the boot structure must also contain swapping space.

The file directory **BS:<SYSTEM>** contains many files by which the system is configured and customized.

A number of files need to be altered at each site to customize the operation of TOPS-20. These alterations may be done any time after the initial boot is completed. Most of the changes take effect

the next time the system is rebooted.

These files are generally protected from being changed by unprivileged users. Thus, to effect the changes described below, you must be use a privileged user name, e.g., Operator, and have the privileges active, via the EXEC command:

```
@enable
!
```

Note that after you use the **Enable** command, the EXEC changes its prompt character from “@” to “!”. The different prompt helps to remind you that whatever privileges you have are now active.

The following are the files that deserve particular attention in customizing:¹

- **BS:<SYSTEM>7-CONFIG.COMD**

This file contains several kinds of commands that are carried out when the system starts running. Each command is system-wide in its application. The sections of this file are

- System-wide logical name definitions: these are almost identical to job-wide logical names but are defined for everyone. You will need to make specific changes to define your site’s printing environment.
- Batch system and Scheduler settings: whether or not to establish *classes* of jobs and the fraction of the machine’s time to allot to each. If no classes are defined, whether or not to let batch jobs compete with interactive ones. Whether or not to bias the system to favor one or the other of either compute-bound or interactive jobs.
- Disk/tape subsystem declarations: how tape archiving works, whether or not to have TOPS-20 process the labels of labeled tapes, what to do with unlabeled tapes, and which disk structures are shared, if any.
- Enabling the use of a “Login Structure”: the logical name **PS:** represents the name of the *public structure*, the structure that contains file directories for individual users of the system. This structure is also known as the *login structure* since it contains the users’ main directories and their passwords. Often, **PS:** and **BS:** are the same structure. However, a customer may define a login structure other than **BS:** by using **CHECKD** to mark a structure as the login structure, and then enabling the use of the login structure by a command in this file.
- Timezone and daylight time parameters.
- Miscellaneous settings: password length, what accounting is being done, and a few others.

Appendix A contains a detailed listing of these commands and their parameters. That appendix describes how to declare the system name, and change the system defaults for system devices, accounting, performance improvements, scheduler controls, file archiving and migration, as well as other system functions.

¹ Many of these files are provided as templates with modification dates 100 years prior to their correct modification date. This is done so that these files will not replace more recent versions when loading from tape. Copies of these files are also placed in **BS:<SYSTEM.TEMPLATES>** for reference.

- **BS:<SYSTEM>MONNAM.TXT**

This file contains one line of text that begins the announcement text output to a terminal when it first connects to the system.

Whenever a user types <CTRL/C> before logging in, when a user makes an initial connection to TOPS-20, or when a user gives the EXEC command:

```
@Information Version 
```

the system prints a message similar to:

```
Your Toad from XKL, TOPS-20 Monitor 7(101604)-1
```

where “7(101604)-1” is the version and edit number of the operating system. You can change the system name, “Your Toad from XKL”, by entering the name you want for your system into the file **BS:<SYSTEM>MONNAM.TXT**.

The text of the system name can contain up to 105 characters. For practical purposes, it should not exceed 32 characters to fit on a single 72-character line of terminal output.

Be enabled; be connected to **BS:<SYSTEM>**; use the command:

```
!Copy TTY: MONNAM.TXT 
```

Then enter the system name and type <CTRL/Z> to end the file.

Use the **Copy** command to give the system a new system name. After you enter the command, the system waits for you to type the new name of your system. After you enter the new system name, type <CTRL/Z>. In the example, the new system name is “Installation-test System”.

```
!COPY TTY: MONNAM.TXT 
TTY: = MONNAM.TXT.1
Installation-test System~Z
!
```

(Note: the characters “~Z” are the echo of <CTRL/Z>.)

You can change this file at any time, but TOPS-20 reads the system name only when the monitor is started (whenever you boot the system).

- **BS:<SYSTEM>TAPNAM.TXT**

This file contains the abbreviated system name that will be written on the volume header label of any labeled tape initialized by users of the system. (Presuming that you have configured the system to process tape labels.) The system name in a volume header label has just ten character positions, so you may have to abbreviate the system name to make it fit in ten or fewer characters.

- **BS:<SYSTEM>SYSTEM.CMD**

This file contains a sequence of OPR commands that take care of a variety of housekeeping chores when TOPS-20 begins operation. OPR is the system operator’s utility program; its commands are described in Appendix B.

- **BS:<SYSTEM>PRINTER.CMD**

One of the housekeeping chores that seems to need taking care of repeatedly is the definition and startup of print spooling. Therefore, we recommend moving these commands into a

separate file so that they can be invoked at need, without the overhead of a full startup via the `SYSTEM.COMD` file.

- **BS:<SYSTEM>LOGIN.COMD**

This file contains commands that are executed on behalf of each job at the time it is logged in. These commands take effect before any commands in the user's `LOGIN.COMD` or `COMAND.COMD` files, described in Section 4.2.

- **BS:<SYSTEM>COMAND.COMD**

This file contains commands that are executed on behalf of each job at the time it is logged in and whenever an EXEC is created for an existing job. These commands take effect before any commands in the user's `COMAND.COMD` files, described in Section 4.2.

- **BS:<SYSTEM>BEWARE-MESSAGE.TXT**

This file contains a message that is presented to every user when an EXEC is started on a terminal for a new job. You may omit this file altogether, or place a message in it.

- **BS:<SUBSYS>LPFORM.INI**

This file contains forms control definitions for the LPTSPL print spooler. LPTSPL is the spooler that can send data to a printer that is attached to a serial port of a network terminal server. More information is presented in Section 4.5.1.

- **BS:<SUBSYS>LPFORM.TXT**

This file contains forms control definitions for the TCPSPL print spooler. TCPSPL is the spooler that can send data to a network host or network printer that supports the `lpd` protocol; see Section 4.5.2.

4.1.2 Network Configuration

The files listed below affect the software that operates the network. In addition to the actual information that directs the system in its use of the network, each of these files includes comments that describe its function in greater detail than is presented below. Please read those comments as you change the files.

- **BS:<SYSTEM>HOSTS.TXT**

This file contains the names and IP network addresses of computer systems “known” to this machine. The file is read at the time TOPS-20 starts operating, so that IP addresses of systems mentioned in this file are known immediately. Among these might be names of systems on your local net, names of gateways, and names of other service providers (e.g., name-lookup services). Changes to this file can be made effective immediately by using the command:

```
!^Eset internet host-table RET The first character is <CTRL/E>
```

- **BS:<SYSTEM>INTERNET.ADDRESS**

This file contains commands that configure XNI-1 ports. Each XNI-1 port can be assigned an IP network addresses and be told how it should expect to see data. There should be one command line in the file for each active Ethernet port. Lines beginning with semicolons are treated as comments. The following lines are examples:

```
nea#0, 198.55.36.131, ethernet, logical-network-mask:255.255.255.0, packet-size:1500, preferred
nea#1, 192.94.202.40, ethernet, logical-network-mask:255.255.255.240, packet-size:1500, preferred
nea#2, 192.94.202.230, ethernet, logical-network-mask:255.255.255.240, packet-size:1500, preferred
```

Changes may be made to this file while the system is running. They will automatically take effect the next time the system is restarted, or they can be made effective immediately with the command:

```
!^Eset internet address 
```

- **BS:<SYSTEM>INTERNET.GATEWAYS**

This file contains the address of at least one gateway to which the system may forward traffic when it knows nothing about how to route it. Generally, all traffic destined for networks not directly connected to the XNI-1 must be forwarded through a network gateway. For example:

```
always 192.94.202.225
```

To notify TOPS-20 of changes to this file, use the command:

```
!^Eset internet gateway 
```

- **DOMAIN:RESOLV.CONFIG**

This file controls the way Internet host names are resolved into addresses. As shipped from XKL, this file contains very few actual control lines, and a considerable number of comments, indicating what local editing should be done. Edits to this file must reflect the way the Domain Name System (DNS) is set up for the network to which the TOAD-1 System is attached. The minimum initial changes that should be made are to specify a remote search path, so that DNS requests will be formed correctly, and to add any user names to the list of those authorized to send resolver control messages. (Initially, the logical name **DOMAIN:** is defined to be **BS:<DOMAIN>**.)

After editing **DOMAIN:RESOLV.CONFIG**, you can make the resolver take the changes by the following commands:

```
!domain:rsvctl    Start the Resolver Control program
RSVCTL>boot 
Are you sure? yes 
RSVCTL>exit 
!
```

After all the necessary changes are made, it is good practice to shut the system down and then reboot it to make sure that the changes actually produce the intended effects when the system restarts. Section 5.1, page 39 describes how to stop the system in an orderly manner; Section 5.2, page 40 describes how to reboot the system.

4.2 Users, Directories, and Accounts

A *user* is a person. A person gains access to TOPS-20 through the EXEC program's **Logi**n command. In **Logi**n, the user identifies himself or herself by supplying a *user-name* and a password. Once logged-in, a user has a *job* which is *connected* to a *directory*. A directory is a repository for files. A user-name is a name, assigned to the user by the system administrator, which identifies the *login directory* for the user. A user's login directory contains that person's mail file, other files unique to that person, and information describing the person's access rights (group membership) and privileges. Initially, following the successful completion of the **Logi**n command, the user's job is *connected* to his or her login directory. To be connected to a directory means that the user has "owner" access to the files in the directory; by default, i.e., unless otherwise specified, files written by the user will go into the connected directory. A job is a collection of processes that TOPS-20 runs on behalf of a user.

An *account* is a number or string, associated with a job or a file. Account strings (numbers) provide finer resolution in cost-accounting when one person may be working on multiple tasks, each of which may need separate time- and resource-accounting. (TOPS-20 can be told to check the validity of an account string.) The subject of "accounts" is mentioned here primarily to note that, as imprecise human beings, we may on occasion lapse into terminological inexactitude and refer to a user-name or login directory as an "account".

4.2.1 Defining New Users

Once the system is operational, you will probably want to define users so that system access may be shared. A user needs, at minimum, a directory with the *login* attribute (the default) and a password defined. A directory, whether for a user or for file storage only, is created or modified by the EXEC command, **Build**:²

```
@enable RET
!build ps:<roget> RET
!!work 1000 RET
!!perm 1000 RET
!!password thesaurus RET
!!RET
!disable RET
@
```

Often, it is a good idea to provide generic **LOGIN.COMD** and **COMAND.COMD** files for each user as a starting point. One common use for **LOGIN.COMD** files is to run a terminal query program that tries to decide what kind of terminal the user is using for a session.

²This use of the **Build** command is privileged; you must be a privileged user (wheel or operator) and have your privileges enabled.

The commands in `LOGIN.COMD` are processed only when a user logs in; the commands in this file should have job-wide application. Commands in `COMAND.COMD` are processed at login (after `LOGIN.COMD`) and any time a new EXEC command interpreter program is started.

In cases where a user should no longer be allowed to use a directory to log in, use the `Build` command to add the `frozen` attribute to the directory. The directory retains all its properties, including the ability to receive mail, but the user is no longer allowed to log in.

User Privileges

TOPS-20 recognizes that certain system maintenance and administrative functions are inappropriate for general users; these functions are restricted to users whose directories show that they have been granted the “privilege” (or “capability”) of overriding these restrictions.³

The three principal privileges in TOPS-20 are called “Wheel”, “Operator”, and “Maintenance”. The system administrator grants a privilege to a user by issuing an appropriate subcommand e.g., `Wheel` while using the `Build` command to create or change the selected user’s directory.

“Wheel” and “Operator” are highly privileged states. They differ slightly in that “Operator” is a bit more restrictive than “Wheel”, but a user who has the “Operator” privilege can build a directory that has the “Wheel” privilege.

The main use of “Operator” is for system backup and tasks to serve the operational requirements of other users. An “Operator” can make any access to any file. (Read access to every file is required for back up; other access is required for file restoration, etc.) An “Operator” can communicate to the output spoolers and batch system via the OPR program.

The main use of “Wheel” is for system administration tasks of an unusual nature, as might be required of a trusted and skilled programmer.

The “Maintenance” privilege allows such activities as accessing the disk without going through the file system. “Maintenance” is restricted to the tasks appropriate to testing and debugging system components. A user who has only the “Maintenance” privilege can not (by any usual means) change his or her own privileges or access files beyond what is usually permitted.

4.2.2 FINGER Data Base

Once a new user-name is created, it is a good idea to declare the identity of the user to the FINGER program’s data base; this is done with the DBEDIT program, which maintains a data base matching personal identities to user-names. This data base is used by the FINGER program, which fields queries about who is logged onto a system.

4.2.3 Mail Aliases

Depending on the e-mail system being used, one or more entries must be made in mail-forwarding files to make the linkage between the name(s) under which mail is received and the directory (or

³As a further safeguard, a user’s privileges are not effective unless they have been turned on by the EXEC’s `Enable` command.

directories) to which it should be delivered. The details of how to do this are covered in the documentation on the particular mailer being used.

4.2.4 Defining Directories

As noted in the previous section, a directory is a repository for files. In many installations, shared data is stored in files that do not belong to any individual user. Such files may be stored in *files-only* directories. To create this kind of directory, use the same **Build** process as before, but assign the *files-only* attribute:

```

@enable RET
!build ps:<trashbin> RET
!!work 10000 RET
!!perm 10000 RET
!!files-only RET
!!RET
!disable RET
@

```

This will produce a directory that is accessible to any number of users, provided they have proper access privileges. (Access privileges are covered in detail in Section 5.3.3.)

4.3 Console Terminal Operation

As noted in Section 2.4, provisions have been made for both a console terminal port and an auxiliary terminal port. The intended distinction between these is that the console port is always active, while the auxiliary port may be turned on or off. When the auxiliary port is turned on, it is effectively in parallel with the console: any output to the console port is also delivered to the auxiliary port, and any input from the auxiliary port is treated as though it had come from the console.

Both the XKL-1 microcode and the TOPS-20 operating system software rely on the console port as the definitive place to print notice messages; some error conditions make it impossible to record error information anywhere else. For this reason, a printer or printing terminal with an RS-232 interface may be attached to the console terminal to preserve a log of events leading up to such a serious system fault.

Activation of the auxiliary console port is controlled by two console commands:

```

XKL-1%Set auxiliary-port { off | on } RET
XKL-1%Define auxiliary-port { off | on } RET

```


The auxiliary port may be turned on only through the console port; it may be turned off from either port.

Both the **Define** and **Set** commands apply immediately; they differ in that the state established by the **Define** command persists (and is re-established) after a power shutdown. Thus, in installations where the auxiliary port is to be active at all times, the console command:

```
XKL-1%Define auxiliary-port on RET
```

should be given. This command alters the non-volatile configuration memory so that, after power is turned off and back on, the auxiliary port will be turned on. Contrariwise, security considerations may dictate defining the port to be off, except when it is set to be on.

This port may be connected to a modem to permit remote diagnosis of system problems. If this is done, the auxiliary port should be turned on only when it is needed. While the auxiliary port is active, the RS-232 signals RTS and DTR are asserted: RTS lets the modem know that data may be transmitted at any time; when DTR drops (because the port was deactivated), the modem will disconnect the phone call.

4.4 Security Considerations

This section deals in general terms with techniques for reducing the ability of unauthorized persons to gain access to your system.

4.4.1 Encrypted Passwords

Each disk structure specifies whether passwords for the directories are to be stored in plaintext or are to be encrypted. To reduce the risk of a break-in, passwords on all structures should be encrypted. While this does not guarantee password security, it eliminates an obvious threat. Disk structures that are preloaded by XKL are defined with password encryption on. To specify that passwords be encrypted on a disk structure, run the CHECKD program:

```
@enable RET
!Checkd RET
TOAD-1 CheckD>define PS: password-encryption RET
TOAD-1 CheckD>exit RET
!disable RET
@
```

4.4.2 Security Procedures

Other ways to reduce the threat of unauthorized access to the system and to data on it are:

- Change the password for Operator. This user-name, defined on all TOPS-20 systems, is a well-known point of attack. The default password for this directory is likewise well-known; thus, anyone might attempt a TELNET connection to a TOAD-1 System, and log in. Since the user-name Operator generally has wide access privileges to files on disks, the risk of data theft or damage is high unless this password is changed **before** the system is attached to a network, and kept secret thereafter.

To change the password for Operator, log in and perform the following commands:

```
@set directory password <operator> 
Old password: xkl-toad       Your old password will not echo.
New Password: new_password     The new password will not echo.
Retype new password: new_password 
@
```

Because the passwords that you type will not show on your terminal, this command requires that you type the new password twice, to confirm that you typed it the same way both times. Please remember your new password.

- Control physical access to the machine, its console, and the auxiliary console. From the console (and from the auxiliary console) it is easy to break into the system. (A person at the console can start kernel DDT without first logging in.) Therefore, when it is not in use, disable the auxiliary console with the console command:

```
XKL-1%Define auxiliary-console Off 
```

especially if this port is connected to an RS-232 line that is accessible from outside the machine room.

- Force users to change their passwords frequently. Avoid short, easily guessed passwords. See Section A.18, page 89 for means by which these ideas can be implemented.
- Consider installing network access controls and connecting the system to the network on the protected side of the access barrier. This reduces the probability that an unauthorized person will be able to break into the system from a remote computer anywhere in the world, even if the passwords for a directory are known. This is discussed further in the following section.

4.4.3 More Extreme Measures

For those who worry professionally about information leakage, it should be noted that there are a number of related ways that a determined attacker can learn things about a system in operation. Once a successful TELNET connection has been made to a TOAD-1 System, several commands are permitted in addition to **Login**; among these are **Finger**, **Systat**, **Help**, and **Information**. Any of

these commands can leak information about logged-in user-names or system data to an unauthorized person.⁴

If this level of leakage is unacceptable, the next defense is to permit only authorized incoming TELNET connections, and deny all others. Volumes have been written on ways to set up networks with more-or-less secure access; the most convenient we know about is a *security island* architecture: a subnet isolated by routers, to which a small number of (unsecured) machines are attached, as in Figure 4.1.

These machines have two purposes: to authenticate users and to serve as a visible front for the hidden parts of the network. Commonly, mail delivery services and Domain Name Service (DNS) are handled through these machines. In addition, a number of self-monitoring processes may run on these machines to aid in detecting and tracking break-in attempts.

The routers between the network segments are then charged with enforcing a connection policy. Incoming TCP connection requests are allowed only in two circumstances: first, from the public side of the island to these authentication machines; second, from the authentication machines to the secure parts of the network. Established connections are allowed to pass through the routers, so that TCP connections originating within the protected domain are not subject to authentication on the security island.

A (lower-performance) variant of this architecture is called a *firewall*. The most common implementation of that scheme places a single machine between the public and private segments of the network; a third network port on the firewall machine serves the island net. Software in the firewall machine then implements essentially the same policy outlined above. Such systems also offer some added network addressing flexibility, which may be interesting to some installations.

4.5 Network Printers

Since the TOAD-1 System was designed as a networked computer, all printing is via network printer spooling. There are two basic ways that TOPS-20 does network print spooling. Both are managed by the standard output queue management, but use different spoolers.

The first method uses the LPTSPL spooler, logical names of the form **LAZPTn:**, and the forms control file **SYS:LPFORM.INI** to send printer data directly to a networked printer. In this case, all the printer spooling is handled locally, and any special formatting instructions required by the printer (such as PostScript) must already be present in the file that you queue to the local TOPS-20 printer queue.

The second method uses the TCPSPL spooler, logical names of the form **PLPTn:**, and the remote queue-name control file **SYS:LPFORM.TXT** to forward a print job to a printer queue on another system⁵ which supports the **lpd** protocol that was originally defined for UNIX. In this case, the remote system handles printer control; it may preprocess the data in prescribed ways before printing. This scheme should be considered carefully before implementation because it transfers some of the burden to another machine.

⁴You may rebuild your system's EXEC to disallow those commands before log in.

⁵This discussion also applies to those printers that can be attached directly to the network and which support the **lpd** protocol.

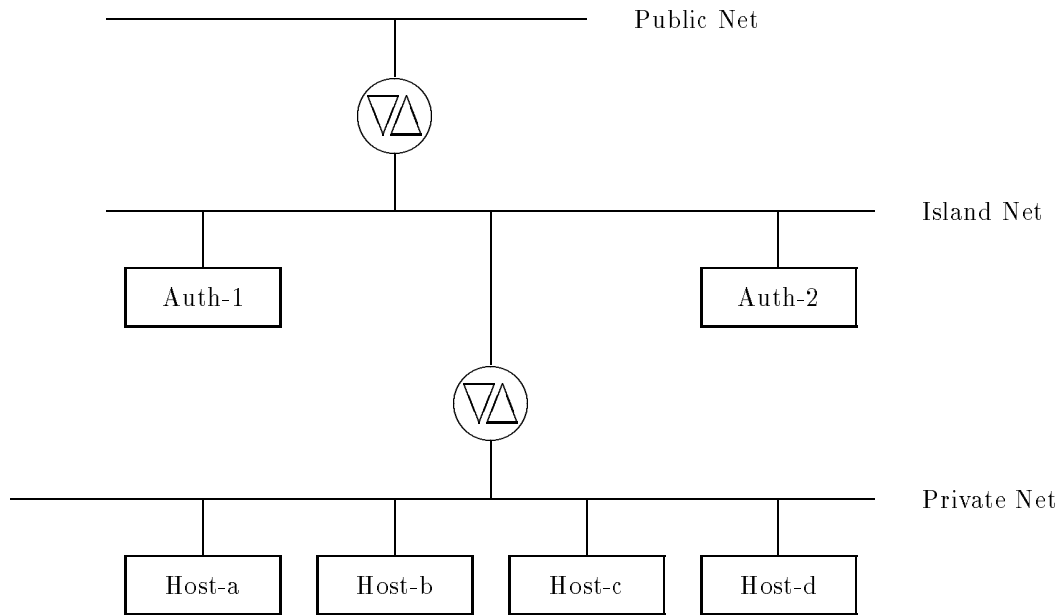


Figure 4.1: Network Security Island

4.5.1 Printing on a Printer Attached to a Terminal Server

To print on a printer attached to a terminal server, you need to have a copy of the LPTSPL spooler running for each local printer queue that accesses it. This usually is achieved by running enough copies of LPTSPL under one job controlled by SYSJOB.

Define a system-wide logical name `LAZPTn`: with the value `TCP:.rhost-rport`, where n is the local printer queue number (also known as the printer unit number), *rhost* is the remote host to connect to, and *rport* is the remote port for access to the desired printer. (Actually, the logical name can be anything you like, but it must be six or fewer characters to be acceptable to the queue management system.) The definition of this system-wide logical name is usually achieved by adding a line similar to the following in the file `SYSTEM:7-CONFIG.COMD`:

```
Define LAZPT0: TCP:.KENT-4002
```

This would direct files queued to local printer unit 0 to be printed on the printer attached to port 4002 on host “kent”.

To start printer spooling for each queue, you must issue OPR commands similar to the following:

```
OPR>SET PRINTER 0 FORMS POST RET
OPR>SET PRINTER 0 PAGE-LIMITS 0:5000 RET
OPR>START PRINTER 0 /DEVICE:LAZPT0: RET
```

The first command sets the *forms-name* associated with unit 0. This forms-name must also appear in the file `SYS:LPFORM.INI` (see below).

The second command allows any length job up to 5000 pages to be printed by one request on unit 0. Such a large number is required for a queue that handles PostScript, because the spooler tries to count formfeeds in the text as a way of estimating the number of pages; this count is unrelated to the number of output pages for a page description language.

The third command starts the printer queue for Unit 0 and associates it with a network printer `LAZPT0:`. It is necessary to use a system-wide logical name for this association, because the command will not accept the file specification used to indicate the remote host and port.

There must be an entry in the file `SYS:LPFORM.INI` for each forms-name used for a network printer. For example, the POST form (mentioned in the OPR command assigning the forms to a printer) might have an entry like this:

```
POST/HEADER:0/BANNER:0/TRAILER:0/LINES:10000/WIDTH:132/EOJ:4/SYNC:4
```

The `/HEADER:0/BANNER:0/TRAILER:0` switches prevent LPTSPL from inserting text into a PostScript stream. The `/LINES:10000` prevents LPTSPL from inserting extraneous formfeeds in the data stream. The `/WIDTH:132` prevents LPTSPL from breaking lines excessively. The `/EOJ:4` sends the value 4 (the ASCII character `<CTRL/D>`, which is interpreted as signalling the end of file by many PostScript printers) at the end of a job. The `/SYNC:4` sends the `<CTRL/D>` character between jobs. Sending

<CTRL/D> both times ensures that the printer knows that a new job is starting, even if the preceding one terminated abnormally or came from some other system that did not observe these niceties.

4.5.2 Printing to a Remote Printer Queue or a Networked Printer

To print to a remote printer queue, or to print to a printer directly attached to the network,⁶ you need to have a copy of the TCPSPL spooler running for each local printer queue that accesses it. This is usually achieved by running enough copies of TCPSPL under one subjob controlled by SYSJOB (usually the same job that runs copies of LPTSPL).

Define a system-wide logical name of the form `PLPTn:` with the value `TCP:.rhost`, where *n* is the local printer queue number (also known as the printer unit number) and *rhost* is the remote host handling printer access. The logical name must be of the form given, as TCPSPL looks for the logical name by appending the unit number to the string `PLPT`. The TCPSPL spooler automatically connects to the remote host at port 515 (this is the port assigned for remote requests to `lpd` on UNIX systems). The definition of this system-wide logical name is usually accomplished by adding a line, similar to the following, to the file `SYSTEM:7-CONFIG.CMD`:

```
DEFINE PLPT1: TCP:.fife
```

This command directs the TCPSPL for unit 1 to connect to host “fife” for remote print queueing.

To start printer spooling for each queue, issue OPR commands similar to the following:

```
OPR>SET PRINTER 1 FORMS POST RET
OPR>SET PRINTER 1 PAGE-LIMITS 0:5000 RET
OPR>START PRINTER 1 RET
```

These commands have exactly the same explanation as for the LPTSPL case above, except that the device `PLPT1:` is implied. The queue-name control file `SYS:LPFORM.TXT`, used by TCPSPL, is different. Each of its entries associates a form-name with a remote print queue:

```
POST:LPT001/NAME:ps
```

`POST` identifies the form type for print jobs in this queue. `LPT001` identifies this entry as being for local printer unit 1 (the leading zeroes are required to make the name six characters). For largely historical reasons, `LPT001` implies that data will go to `PLPT1:.` The target queue name for the print file on the remote host is identified by `/NAME:ps`. (These queue names are usually in `/etc/printcap` on UNIX systems).

⁶This description presumes that a printer that is directly attached to the network accepts the “`lpd`” protocol.

Chapter 5

System Operation

This chapter describes some of the tasks that are required occasionally: shutting down timesharing, rebooting, rebooting for stand-alone operation; file system tasks: creating a file structure, creating directories, backing up a file structure, checking file system consistency, rebuilding a file system from a back up tape, and formatting a disk. Further, tape back up policies are described, as are a set of periodic tasks.

5.1 Shutting Down the System

The TOPS-20 operating system generally has many activities under way at any moment. However, some activities may require the entire system's resources. A period in which access to the system is limited is called "stand-alone" time.

To terminate general system activities cleanly and to warn users of an impending shutdown, use the EXEC privileged command:

```
!^Ecease {now | date_and_time | +time_from_now} RET
```

As the designated time approaches, users are warned of the shutdown by messages on their screens. When the shutdown time arrives, all jobs, except for job 0 and the job logged in on the console terminal, are logged out. After the message "Ready for shutdown" appears on the console terminal, the system is in "stand-alone" mode, where only Job 0 (system services) and the job on the console terminal are active. To stop the system entirely (for example, before rebooting to resume normal timesharing), use the character <CTRL/\> to summon the console prompt, "XKL-1>". Then give the console command:

```
XKL-1>Shutdown RET
```

After a few moments the message "Halted" should appear, followed by the console prompt XKL-1%.

5.2 Booting the System

The usual boot process is accomplished by typing the console command:

```
XKL-1%Boot RET
```

The **Boot** command loads an executable file into memory and starts the program. By default, i.e., in the absence of parameters, the **Boot** command loads the file `<SYSTEM>MONITR.EXE` from the *auto-boot* structure.

The default file name, i.e., the file name that **Boot** uses when none is specified in the command, can be changed. The console command:

```
XKL-1%define boot file_name RET
```

changes any part of the default file name (or switches, see below). This change is “permanent”: it has effect on every subsequent **Boot** command, until it is changed by another **Define Boot** command.

The **Boot** command also accepts a file-name as a parameter; this parameter is effective just once, in the command where it appears. This might be useful when experimenting with a new version of the monitor, or when running a stand-alone diagnostic program.

Care for MONITR.EXE

The file `BS:<SYSTEM>MONITR.EXE` exists in the ordinary file system; it is a valid `.EXE` file. However, it has been specially contrived to run properly when loaded by TDBOOT’s **Boot** command.

This file can be copied and it will remain bootable. It can be patched via FILDDT (File DDT) and still be bootable. However, one patching technique (which worked in the DEC versions of TOPS-20) will fail when performed on the TOAD-1 System: do **not** copy `MONITR.EXE` by the sequence of EXEC commands **Get** and **Save**. The EXEC’s **Save** command will write a file that will not run properly when it is booted.

5.2.1 Rebooting the System for Stand-Alone Operation

Although TOPS-20 is a multi-user system, on some occasions, the system is wanted for *stand-alone* operation, in which all or most of the system facilities are present, but from which general users are excluded.

To boot TOPS-20 for stand-alone operation (presuming it is already shut down), you must start the system in its debugger programer (called KDDT, for Kernel DDT). This is accomplished by the `/start:+3` option in the console **BOOT** command. When KDDT is running, open the memory location called `DBUGSW`, change the value stored therein to 2, and close the location. Then start the system at location 161 (reported as the “entry vector location” by the **Boot** command). Your console session should resemble the following:


```

XKL-1%boot /start:+3 RET
[Using default string: ""]
[Structure PS:]
[Directory <ROOT-DIRECTORY>]
[Directory <SYSTEM>]
[File PS:<SYSTEM>MONITR.EXE.1]
[Loading .....]
[File entry vector is at 1,,161, length is 5]
[PDVA at 1,,141077]
[Filled through LPN 2126]
[Starting at 1,,164]
KDDT
DEBUGSW/ 0 2 RET Type "DEBUGSW", slash, "2" and press Return.
161ESC g The ESC will echo as "$".
[PS mounted]
.
.
.

```

This procedure sets the debugging flag to 2 and starts TOPS-20. The debugging flag, called "DEBUGSW", has several purposes. For stand-alone operation, when "DEBUGSW" has the value 2 or greater, the SYSJOB program reads the file `SYSTEM:7-SYSJOB.DEBUG` instead of the usual `SYSTEM:7-SYSJOB.RUN` file. If the `.DEBUG` file *omits* the line:

```
^Eset login any
```

then general login will *not* be allowed: jobs can be created only on the console and pseudo terminals. The `.DEBUG` file may contain as much or as little of the normal system utilities as you find appropriate to your stand-alone environment.

5.3 File System

This section describes some of the tasks associated with the file system.

5.3.1 Creating Disk Structures

A *disk structure* is a named collection of similar disks that, together, contain a file system. A file system is an organization of disks into a collection of named directories that contain named files.

A structure can be one disk or several. TOPS-20 supports structures up to a (present) limit of about 17 Gigabytes. If there are multiple disks in a structure, they all must be compatible as to capacity and geometry; generally, this condition is satisfied if the disks are all the same make and model.

Arguing for multiple small structures is the notion that each is independent, each can be backed up independently, and the failure of one does not mean the loss of data on the others. In favor of large structures is the idea that one structure is easier to allocate and to manage than several.

A new disk structure is created using the CHECKD program. If your disk(s) is(are) not yet formatted, see Section 5.3.8. This program knows how to label one (or more) formatted disk(s) so that directories can be constructed by TOPS-20. A sample session to create a single-unit non-login structure follows:

```

@Enable 
!Checkd 
TOAD-1 CheckD>create FROG: 
Enter alias: FROG: 
How many units in this structure? 1 
DECIMAL Channel, Controller, and Unit numbers for logical unit 0: 4,-1,6 
Number of pages to allocate for swapping? 0 
Owner name? Operator 
TOAD-1 CheckD>mount FROG: 
[Structure FROG: mounted]
TOAD-1 CheckD>define FROG: password-encryption 
TOAD-1 CheckD>dismount FROG: 
[Structure FROG: dismounted]
TOAD-1 CheckD>exit 
!disable 
@

```

Channel, Controller, and Unit numbers

The only question in this dialogue that requires advance preparation is the one about the channel, controller, and unit numbers. The TOPS-20 *channel number* is related to the SCSI bus to which a device is attached. Channels 0 to 3 are buses 0 to 3, respectively, on the first XRH-1; channels 4 to 7 are buses 0 to 3, respectively, on the second XRH-1, etc. By the “first” XRH-1 we mean the XRH-1 whose backplane slot has the lowest number; the “second” XRH-1 is in a slot with a number higher than the first, but lower than any other XRH-1. The controller is -1 for all supported SCSI devices. The unit number is the SCSI Target ID of the device on which the structure is being created.

If you type “?” in response to this question, you will get a list of the devices available; this list includes all three numbers. (Better yet, use the **Show** command in CHECKD before starting **Create**.)

This CHECKD process creates a skeleton disk structure, with a few directories and housekeeping files on it. A more complete description of the CHECKD program commands is in Appendix C.

5.3.2 Creating Directories

As mentioned in Section 4.2.1, the EXEC command **Build** is used to create or change a directory. For directories that are to be used only for data storage (not used as login accounts), the attribute *files-only* should be set during the **Build** operation.

5.3.3 File and Directory Protection

Every file and directory has a protection number that tells exactly who can do what to that file or directory. The interpretation of this number is different for a directory than for a file. The number consists of six digits divided into three pairs: the first pair says what access to grant the owner, the second pair says what access to grant to users of the directory's groups, and the third pair says what access to grant anyone else.

To be recognized as the *owner*, you need to *connect* to or *access* a directory; there is no registered owner. The EXEC commands **Connect** and **Access** are used to establish ownership access with respect to files in a directory.¹

To be considered a *group member*, your login directory must contain a user group number that matches one of the directory group numbers recorded in the directory being accessed.

If a user has not established owner access or group access to a directory, that user is considered *other* for the purpose of determining access to files.

File (and Directory) protection code numbers specify the types of access granted to "Owner", "Group", and "Others". The protection code number has six octal digits. The left pair of digits specify the owner's access permissions; the middle pair of digits specify group members' access permissions; the right pair of digits specifies other's access permissions.

The table below shows the kind of access each code allows; the numbers are octal. To specify a combination of allowed access privileges, OR the values for each field into a two-digit protection number for each class of access. Create a six-digit protection number from three groups of two-digit numbers, one pair of digits for each of owner, group, and other access.

<i>Value</i>	<i>File access</i>	<i>Directory access</i>
00	No access to the file	No access to the directory
02	Show filename in directory list	Reserved
04	Append to the file	Create files in the directory
10	Execute the file	Connect without a password
20	Write and delete the file	Reserved
40	Read the file	File protections specify access
77	Full access to the file	Full access to the directory

For example, file protection of 775200 means only the owner can change the file; anyone in the directory group can read it or execute it; and access is denied to others.

¹In addition to establishing ownership access, the **Connect** command sets the default directory for file access to be the connected directory.

To display the protections of your files, use the **Vdirectory** (verbose directory) command, or the **Directory** command with the **Protection** subcommand.

To change the protection of a particular file, use the command:

```
@Set file protection file_name protection_value RET
```

To set the protection a file will get when it is created in a selected directory, use the command:

```
@Set directory file-protection-default directory_name protection_value RET
```

Each subdirectory is a file in a superior directory and has a file protection word. The only meaningful value is 02, which controls whether or not the subdirectory is to be included in a listing of files in the superior directory. Otherwise, operations involving a directory are controlled by the *directory protection*. This directory protection number is interpreted according to the right column in the table above. To change a directory's protection, use the command:

```
@Set directory protection directory_name protection_value RET
```

Particular mention should be made of the 10 code for directory protection. It should be included for most directories in both the Owner and Group fields; it is embarrassing to be asked for a password when connecting to one of your own directories, and irritating to be asked for a password when connecting to a directory in a group of which you are already a member.

5.3.4 Managing Directory Access

A further discussion of group access is needed. In TOPS-20, a group is defined by a group number in both the directory to be accessed (the *directory-group* list) and in the login directory of the user who is permitted the access (the *user-of-groups* list). When a job attempts an access to a target directory, a match is required between an entry in the login directory's user-of-groups list and one of the target directory's directory-group list.

It is possible to arrange directory access using these built-in facilities; for systems with many directories, this approach is quite tedious. The GROUPE program² assists in managing group definitions by remembering a name corresponding to each group number and allowing the system administrator to add or remove user directories and file directories from named groups. This is still tedious, but less so.

If a change in group membership is made to a user's login-directory, it will not normally take effect until the next login. To make a new group available immediately to a logged-in job, the user must enter the command:

```
@access <login_directory> RET
```

²GROUPE (pronounced 'grüp-ē) is a *contributed program*, i.e., a program supplied by users. Such programs are supplied "as-is"; they are not supported by XKL.

Figure 5.1: .MIC File for System Back Up

```

@rewind mta0:
@set tape format high-density
@copy bs:<system>monitr.exe mta0:   Do not use Save: see Section 5.2,
page 40
@copy system:exec.exe mta0:
@copy sys:dluser.exe mta0:
@dluser
*structure ps:
*dump ps-dluser.dat
*exit
@copy ps-dluser.dat mta0:
@copy sys:dumper.exe mta0:
@set tape format core-dump
@dumper
*tape mta0:
*ssname PS: Back Up
*save ps:<*>*. *.* ps:<*>*. *.*
*rewind
*exit
@kmic

```

5.3.5 File System Back Up

Since the TOAD-1 System is a multiuser system, it is strongly recommended that the data be backed up onto tape regularly. There are a variety of motivations for this:

- Users expect to have their data available and do not always appreciate that things break. Back up tapes make it possible to recover data if a disk drive malfunctions.
- Users occasionally delete data that they really need; then they want someone to get it back for them. Back up tapes can supply the data.
- Managers expect disaster recovery plans to be in place. A sensible plan of back up and off-site storage will permit recovery from fairly large disasters in a short period of time.

DUMPER and DLUSER are a pair of programs that save and restore data and directories. DLUSER saves or restores directory information; DUMPER saves or restores files. These programs are used to create archival tapes and to create *crash recovery* tapes. We find it useful to have a MIC file with commands that produce the tape. Such a file is shown in Figure 5.1.

This command file builds a tape with files ordered as they would be needed to restart and rebuild a system without depending on anything that was stored on disk: first, the monitor; second, the EXEC command interpreter; third, the DLUSER program, which can create directories; fourth, the data file

for DLUSER, which contains the directory structure information; fifth, the DUMPER program, which can restore data into those directories; sixth and last, DUMPER's saveset which contains the data.

The procedure outlined above is most appropriate for the boot structure. Other structures can be backed up using the same procedure, or, for structures other than the boot structure, the copies of the ".EXE" files can be omitted, although the data file for DLUSER must be provided somewhere.

5.3.6 Periodic Tasks

When TOPS-20 stops abnormally, it may leave "lost pages" on the disk. These are disk pages that have been assigned to incompletely-created files. An incompletely-created file has not been connected to its data pages. The CHECKD program should be run periodically to reclaim the lost pages, making them available for file storage.

The function that locates lost pages is summoned by the CHECKD command **Check Bittable**. This process writes a file containing the page numbers of the lost pages.³ The lost pages can be returned to the file system by the CHECKD function invoked by the **Release** command.

CHECKD requires private access to the structure on which it is performing the Check Bittable function. This means that CHECKD can run on the boot structure or on the login structure only when the system is stand-alone. For the boot structure, you have the opportunity to run CHECKD whenever the system is rebooted manually: answering the question "Run CheckD?" in the affirmative will cause CHECKD to check the boot structure.

Note

If CHECKD encounters a problem while checking the boot structure, the system will not come up for timesharing. Instead, TOPS-20 will announce that there is a problem, and it will allow login only on the console terminal. If this should happen, consider doing a full back up of your boot structure. Some problems that CHECKD stumbles on are benign, but others are quite severe. It may be possible that the system can be rebooted to regular timesharing if CHECKD is not run, but at least think about doing a full back up before you reboot.

Further information about CHECKD is in Appendix C.

5.3.7 Rebuilding the Boot Structure from Tape

When the boot structure is damaged beyond repair, the procedure described here can be followed to rebuild it. Rebuilding is an activity that you should try to avoid: it takes a long time and it restores only the data as of the most recent back up tape. Newer data may be on the disk, and some effort may be worthwhile to recover it before beginning the rebuild sequence.

First, consider how to avoid rebuilding the boot structure:

³The **Check Bittable** function is more fully described in Appendix C.

- Does the structure have good home blocks? The console command:

```
XKL-1%Show home-blocks 
```

will display the structure name if the home blocks are good. If the home blocks are bad, you probably need to rebuild the entire structure.

- If the structure has good home blocks, can you boot, i.e., load an operating system, from the structure? If you can, then the <ROOT-DIRECTORY> and <SYSTEM> directories are intact. If you can't boot from the disk, boot TOPS-20 from a back up tape. Other directories may be intact, and you may be able to save data in their files.
- If TOPS-20 can run, you should try something less drastic than rebuilding: see the documentation for the CHECKD program, in Appendix C.
- If TOPS-20 mounts the structure but has other structure-related complaints, you might try reconstructing the <ROOT-DIRECTORY>, the index table, the bit table, etc. These procedures are dangerous: if not done correctly, they can destroy data; therefore, undertake these activities with extreme caution. These options are described as "Less Drastic Measures" in the next section.
- If you have a second structure that has swap space on it, the second structure could be mounted as the boot structure. Then programs such as CHECKD, FILDDT, etc. (if present on your second structure) could be used to examine and perhaps repair the failing structure. You may need to boot from tape, or boot from the copy of MONITR.EXE that you have thoughtfully provided on your second structure.
- If all of the above fail, rebuild the structure.

5.3.7.1 Less Drastic Measures

This section describes measures short of rebuilding the entire structure from a back up tape. Please read the next section, "Rebuilding from Tape", before attempting any of the following.

Boot the TOPS-20 (from tape or from an alternate structure), using the `/start:+1` switch in the `Boot` command. TOPS-20 will start, partially initialize itself, and enter a dialogue with you:

```
XKL-1%rewind mta0: 
XKL-1%boot mta0:/start:+1 
[Using tape MTA0: at 2./0./3./0.]
[Loading .....]
[File entry vector is at 1,,161, length is 5]
[PDVA at 1,,141077]
[Filled through LPN 2125]
[Starting at 1,,162]
[For additional information type "?" to any of the following questions.]
Do you want to replace the file system on the system structure? no 
```

By answering “no” to this question, you avoid the path that wipes out the entire structure (which is the path to take when rebuilding from tape). Instead, the system will ask about other tasks it could perform relating to the system structure (i.e., the boot structure).

After mounting the system structure, TOPS-20 will continue the dialogue by asking

Reconstruct ROOT-DIRECTORY?

and

Rebuild Bit Table?

Either question accepts a “yes” or “no” response.

Reconstructing the root directory is accomplished by deleting the old root directory file and copying the back up root file to replace the old one. This should be done if you believe that the old root directory has become corrupted, i.e., if you believe that some directories exist that can't be found any more.

The bit table should be rebuilt if the system tells you that bit table counts are bad, or if you see messages that suggest inconsistencies in the bit table (e.g., deleting a disk page already marked as free). The bit table is rebuilt in two steps: first TOPS-20 obliterates the old bit table; then it runs CHECKD to create a new one.

If neither option is selected, the TOPS-20 will not take any special action regarding the system structure: it will continue as a normal start-up.

5.3.7.2 Reinitializing the TOPS-20 File System from Tape

To reinitialize the TOPS-20 file system, you must define the name of the system structure (the boot structure) and the number and location of each disk drive that is part of the system structure.

By following this procedure, you will erase the pointers to any information already stored on the target disk units. You will reinitialize the TOPS-20 file system by creating:

1. New home blocks (which contain pointers to the beginning of the file system).
2. The directory <ROOT-DIRECTORY> (which contains pointers to all the directories in the system).⁴
3. The initial directories
 - <ACCOUNTS> ,
 - <NEW-SUBSYS> ,
 - <NEW-SYSTEM> ,
 - <OPERATOR> ,
 - <SPOOL> ,
 - <SUBSYS> ,

⁴Properly speaking, <ROOT-DIRECTORY> contains pointers to the “top-level” directories; those directories may have pointers to sub-directories.

- <SYSTEM>, and
 - <SYSTEM-ERROR>.
4. The system swapping space (the area allocated for the movement, by the monitor, of pages between memory and disk).

Note

See the TOPS-20 System Manager's Guide for a detailed explanation of all the above.

Later, you will reload data into the newly-initialized system from tape.

If you decide to rebuild a structure, you need two things: a crash-recovery tape, and the details of where the target disk is attached. The target disk information is obtained by using the console command:

```
XKL-1%show configuration mass-storage RET
```

You can also get this information by typing “?” at the point in the dialogue where it is asked for, but if there are multiple disk structures, the output from the **Show Configuration** command may be more useful.

A crash-recovery tape must contain the following files, in order:

- **MONITR.EXE**, the executable image of the TOPS-20 operating system.
- **EXEC.EXE**, the executable image of the TOPS-20 command processor.
- **DLUSER.EXE**, the executable image of the directory dump and restore program.
- A file of text data from which the **DLUSER** program can reconstruct directories.
- **DUMPER.EXE**, the executable image of the file dump and restore program.
- A **DUMPER** saveset that contains the data that you will restore to the rebuilt disk.

Insert the tape in the tape drive⁵. In this example, we assume that your tape is in the drive identified as “**MTA0**”; if your tape is in some other drive, you will have to use the appropriate name wherever “**MTA0**” appears in this example. Use the console command:

```
XKL-1%boot mta0:/start:+1 RET
```

to load the operating system file into memory and to start it at the first alternate in the entry vector.

The monitor will partially initialize itself and then begin a dialogue with you:

⁵See the Operational Note at the end of Section A.3, page 79: the names by which **TDBOOT** identifies the tape units may differ from those you are accustomed to.

```

XKL-1%rewind mta0: 
XKL-1%boot mta0:/start:+1 
[Using tape MTA0: at 2./0./3./0.]
[Loading .....]
[File entry vector is at 1,,161, length is 5]
[PDVA at 1,,141077]
[Filled through LPN 2125]
[Starting at 1,,162]
[For additional information type "?" to any of the following questions.]
Do you want to replace the file system on the system structure?

```

To create the system structure, type "YES" and .

Note

If you make a mistake at any time during this procedure, type <CTRL/\> to get to the XKL-1> prompt. Then type the console command:

```
XKL-1>Halt 
```

Then you may restart this procedure.

```

Do you want to replace the file system on the system structure? yes 
Do you want to define the system structure?

```

To make TOPS-20 write new home blocks describing the system structure, respond "Yes" and press .

```

Do you want to define the system structure? yes 
How many disk units are in this structure:

```

Count the number of disk units you plan to have in your system structure. Type the answer and . This example uses 1.

```

How many disk units are in this structure: 1 
On which "CHANNEL,CONTROLLER,UNIT" is logical unit number 0 mounted?

```

If you collected data previously via the console command:

```
XKL-1%Show Configuration Mass RET
```

you will have to translate “slot” and “bus” to a TOPS-20 channel number; this is explained in Section 5.3.1, page 42.

Even if you know the answer, please type “?” to let TOPS-20 show you what disks it knows about:

```
On which "CHANNEL,CONTROLLER,UNIT" is logical unit number 0 mounted? ?
[Enter a triple of numbers separated by commas that specify the channel,
 controller, and unit upon which the appropriate disk is mounted. The
 following is a list of valid channel, controller, and unit numbers:
   0,-1,1 ;Type = SEAGATE ST15230W          0298
]
On which "CHANNEL,CONTROLLER,UNIT" is logical unit number 0 mounted?
```

Notes

The example shows channel and unit numbers for a specific configuration. The numbers for your system will probably be different, as they are generated by your own system configuration.

The controller number is always -1 for all system structures.

If a drive that you want to use is not listed, the drive may not be cabled correctly, or it may not be powered on. Abort this procedure by typing `<CTRL/\>` followed by the console command:

```
XKL->Halt RET
```

Correct the problems and start over.

Type the channel number, the controller number, and the unit number of the desired drive, followed by RET. You must answer this question once for each disk unit in the structure.

The controller number of all drives is -1 .

```
On which "CHANNEL,CONTROLLER,UNIT" is logical unit number 0 mounted? 0, -1, 1 RET
Do you want the default swapping space?
```

Error: If your answer is invalid, the system prints one of several messages and repeats the question. You can simply repeat the current step.

If your answer is valid but is not the one you wanted, abort this procedure.

The default swapping space is 100000 pages. This reflects the growth of programs due to extended addressing and the use of high-level languages, the expanded memory capacity of the system, and the desire for more simultaneous users. It can be reduced cautiously. It can be expanded to about 250000. We recommend that you take the default. (Note: the actual number of pages allocated to swapping will be slightly larger than the requested amount, as the swapping space is rounded upwards to an integral number of cylinders on the structure.)

Respond “Yes” and type .

```
Do you want the default swapping space? yes 
Do you want to enable password encryption for the system structure?
```

Error: If you type “No” and then type an invalid number, TOPS-20 prints:

```
? Invalid number of swapping pages for this type of disk
```

Then it returns to ask if you will accept the default swap space.

If you type a number that is valid but which is not the one you wanted, type <CTRL/\>, halt the processor, and start over.

If you want user passwords to be encrypted on the disk, type “Yes” and . We strongly recommend that you encrypt passwords.

```
Do you want to enable password encryption for the system structure? yes 
What is the name of this structure?
```

Note

Before answering with a structure name, see the *TOPS-20 System Manager's Guide* for information on naming structures.

TOPS-20 recognizes any six-character alpha-numeric name as a valid structure name. Some programs may have difficulty if the first character is numeric. Type the name of the structure and . The system prints a message to indicate that the structure is successfully defined and mounted. The following examples use “PS” as the structure name:

```
What is the name of this structure? PS 
[Structure "PS" successfully defined]
[PS mounted]
```

At this point the new structure has been defined, initialized, and mounted. TOPS-20 is running, but there are no files on the disk.

If the disk unit is freshly formatted, TOPS-20 will complain that there are no BAT blocks. If this happens, respond “Yes” and .

```
? PS unit 0 has no BAT blocks.
Do you want to write a set of prototype BAT blocks? yes 
```

Error: If certain error conditions occur while TOPS-20 is mounting the structure (a drive is write protected, for example), the system prints an error message followed by:

```
?Have the problems mentioned above been corrected yet:
```

When the problem has been corrected, type “Y” and .

```
[Loading Internet host names]
%%No SETSPD.
TOAD-1 System, TOPS-20 Monitor 7(101212)-6, built Fri 29-September-1995 9:01AM
Enter current date and time:
```

The message “%%NO SETSPD” means that the system cannot run the SETSPD program. This is expected because you have not yet loaded the SETSPD program into the file system. This message can be printed at any time before the system requests the date and time.

Enter the current date and time in the form mm-dd-yy hhmm and type . The system will repeat the date back to you and it will request confirmation; type “Y” and .

```
%%No SETSPD.
TOAD-1 System, TOPS-20 Monitor 7(101212)-6, built Fri 29-September-1995 9:01AM
Enter current date and time: 10-6-95 1320 
You have entered Friday, 6-October-1995 1:20PM,
is this correct (Y,M) y 
Why reload?
```

Before you get a chance to answer the question “Why Reload?”, the system will print some pent-up errors. Again, these relate to files that are absent because the disk has been scrubbed clean.

When the system is done ranting, answer “Why reload?” with “New” and other verbiage, then type .

```

Why reload?
6-Oct-95 13:20:00 BUGINF: NOADDR - Failed to find SYSTEM:INTERNET.ADDRESS file
Job: 0, User: OPERATOR
6-Oct-95 13:20:00 BUGINF: NOHSTN - Failed to find host names SYSTEM:HOSTS.TXT
Job: 0, User: OPERATOR
New Build new structure 
Problem with <SYSTEM>ACCOUNTS-TABLE.BIN
Cannot find error message file - Account validation is disabled
6-Oct-95 13:20:11 DDMP: Started
No SYSJOB

```

Note that “New Build new structure” answers the question “Why reload?” that appears several lines before the response.

The system prints the message “Account validation is disabled”: the ACTGEN program did not run because it is not on disk. TOPS-20 runs the DDMP process. Then it prints “No SYSJOB” because SYSJOB is not on disk. Ignore these messages.

After the message “No SYSJOB” appears, type <CTRL/C>. Normally, this would start the EXEC, the TOPS-20 command processor, but the EXEC isn’t on the disk. Instead, you will get the MINI-EXEC.

Each command in the MINI-EXEC (MX) is a single letter; MX will finish typing the command name as soon as you type the first letter. If you make typing errors while in MX, type <CTRL/U> to erase the current line.

The crash-recovery tape contains EXEC.EXE as the second file. TDBOOT read the monitor from the first file; it left the tape positioned to read the second file.

Type “G”; MINI-EXEC prints “ET FILE ”. Next, type “MTA0:” and . This will load the EXEC command processor from tape into memory. Now, type “S”; MX types “TART”; type to start the EXEC. When the EXEC starts, it will print a header containing its version number; then it will prompt with the “@” character.

Note

If you are using a tape drive other than tape drive 0, modify your command accordingly in the following steps and/or error recovery procedures, for example, MTA1:, MTA2:.

```

^ C
NO EXEC
MX>gET FILE mta0: 
MX>_START 
TOPS-20 Command processor 7(4143)-5

```

Ⓞ

You are now logged in as Operator, a privileged user-name. Enable the privileges, so that you will be able to write files on other directories.

You have already read the first two files from the tape. (Those were the monitor and the EXEC.) The third file is the executable image of the DLUSER program. We shall run DLUSER from tape and rebuild the structure's directories according to the data found in the fourth tape file. When DLUSER starts, give it the command **Structure** and supply the actual name of the structure that you are building. In this example, we use the name **PS:**, but you should use the actual name that you gave in response to the monitor's question "What is the name of this structure?":

```
@enable [RET]
!set tape format high-density [RET]
!run mta0: [RET]
DLUSER>structure ps: [RET]
DLUSER>load mta0: [RET]
DONE.
DLUSER>exit [RET]
!
```

The process initiated by the **Load** command may take 5 to 10 minutes, depending on the extent and the complexity of the directories that you are restoring. When DLUSER finishes loading tape data into the new structure's directories, use the command **Exit** to leave DLUSER and return to the EXEC.

Errors: If you make an error typing a command to DLUSER, you can erase the present command line by typing <CTRL/U>.

If you have already typed [RET], you can stop DLUSER by typing ^C once or twice. Then you will have to rewind and reposition the tape:

```
!rewind mta0: [RET]
!copy mta0:,mta0: nul: [RET] Read and discard monitor and EXEC files
!run mta0: [RET] This should restart DLUSER
```

Security Note

If you restore your boot structure from the Ship tape supplied by XKL, the Operator password will be restored to *xkl-toad*; this should be changed as soon as possible.

The next stage of reconstructing the disk is to use DUMPER to restore all the files from the crash-recovery tape. DUMPER is rather particular about the names of the files it will restore: it needs to be told the structure name from which the data were saved and, if different, the name of the structure onto which they are to be restored. If you don't know the name of the structure from which the data were saved, the name "DSK*:" will match any name.

See the DUMPER description in *TOPS-20 User Utilities Guide* for an explanation of any messages that DUMPER may print on the terminal.

Run DUMPER from the fifth file on the tape. Tell DUMPER which tape to use; then enter the **Restore** command as shown below:

```
!run mta0: 
DUMPER>tape mta0: 
DUMPER>restore dsk*:<*>*. *.* ps:<*>*. *.* 
Saveset "PS: Back Up" 6-Oct-95 1103
...
End of Saveset.
Total files restored:    10903
Total pages restored:   144474
DUMPER>
```

(The number of files and pages that are reported to you will vary, depending on the tape that you use.)

When DUMPER finishes the saveset, it will print statistics. Then it will prompt again for a command. Tell DUMPER to "Exit". Then tell the EXEC to unload the tape:

```
DUMPER>exit 
!unload mta0: 
!
```

If at any point you need to reposition the tape, the command:

```
!Copy MTA0: NUL: 
```

is much faster than the command:

```
!Skip MTA0: 1 file 
```

Finally, run MAKDMP to create <SYSTEM>DUMP.EXE. Because this file is very large, it is not customarily marked as dumpable; hence it cannot be restored from tape. (This program may take 5 or 10 minutes to complete.) In the example below, the file is created for a system that has 32 Mwords of main memory. Size your DUMP.EXE to be as large as your actual memory configuration:


```
!makdmp [RET]
MAKDMP>create bs:<system>dump.exe 32 Mwords [RET]
MAKDMP>exit [RET]
!
```

Note

If at some future time, you add more main memory to the system, please recreate the `DUMP.EXE` file, sized to accommodate the additional memory.

Various system services were missing from the disk when the system was started; consequently, they are not running. It is generally a good idea to shut the system down (see Section 5.1, page 39) and reboot before going much further.

If you have restored from a back up tape that was made from your data, your system will be configured as it was when that back up was created. If you have restored from the Ship tape that XKL supplied with the system, you will need to perform the customization steps described in Chapter 4.

5.3.8 Formatting a Disk

Disks supplied to you by XKL will be correctly formatted for use in the TOAD-1 System.

SCSI disks usually come from their factories formatted for 512-byte sectors. TOPS-20 needs either 576- or 2304-byte sectors.

Please aim carefully

We are sure that you already know that the format process utterly destroys whatever information is on a disk. We apologize for mentioning it again.

The format command can be performed only on a halted system. If TOPS-20 is running, shut it down as described in Section 5.1.

Use the console command:

```
XKL-1%Show home-blocks [RET]
```

to identify the disk units attached to the system. Note the slot number of the XRH-1 as reported in the display. Note also the bus number and target ID number of the particular disk to be formatted.

Use the console command:

```
XKL-1%format slot bus target RET
```

to begin the format process.⁶ The format command will proceed through a dialogue to be sure that you have the right disk. For the high-capacity disks that XKL supplies, the formatting operation can take more than an hour, during which the system won't do anything else.

Following the first format phase (called "hard format"), the second format operation ("soft format") will happen automatically. If the format command determines that the first phase is not mandatory, it will offer to bypass it and go directly to the soft format operation.

Hard format establishes the disk sector size so that TOPS-20 can use the disk. Soft format asks the disk a series of questions to determine (and record) the precise geometry of the disk so that TOPS-20 can make best use of it.

5.4 Tape Backup

Depending on the needs of your organization, a more or less elaborate schedule of incremental and full back up to tapes may be required. The DUMPER program is usually used for this purpose.

Careful thought should be given to the tradeoffs between accumulating many tapes (to be sure of having all saved versions of data), and selectively reusing tapes (to reduce physical storage requirements). Further, depending on the kind of disaster recovery your organization needs, it may be necessary to archive tapes in geographically separate locations. Such archives can either be made in addition to locally stored back ups, or they can be selected from the back up library after back up data reaches a certain age.

Definitive guidelines are scarce on this topic, because it is essentially an insurance issue. As such, the choices made will reflect the degree of protection that an organization is willing to buy for itself. Design of the back up policies is made considerably less mysterious if this point is kept in mind.

When making a backup of the boot structure, it is a good idea to make a tape containing a copy of the Monitor, the EXEC, DLUSER, data for DLUSER, and DUMPER as the first five files on the tape. You may put this data on a separate tape, or in front of the files backed-up by DUMPER.

When making a backup of any structure, it is a good idea to have up-to-date DLUSER data for the structure included on the tape. This can be done by the scheme described in the previous paragraph, or you may run DLUSER prior to running DUMPER and leave the data in a well-known place, e.g., <OPERATOR>USERS.TXT on the given structure. (In such a case, restoration of the complete structure requires the additional steps of using DUMPER to restore that file and then running DLUSER to restore the directories before running DUMPER to restore all the files.)

⁶The format command accepts a *logical unit number* (LUN) following the target number. At present, TOPS-20 supports only disks that have one logical unit (which is numbered 0); you may omit the LUN for these units.

5.5 Tape Handling

To read industry-standard tapes, the TOAD-1 System can be ordered with the XTU-1 nine-track tape subsystem. Many sites will have such tapes that archive data of considerable age. XKL recommends that this data be transferred to new DAT (or DLT) cartridges, for several reasons:

1. To make sure that the data can still be read from the archive tapes.
2. To capture the archival data on a more robust and more compact medium.
3. To reduce the volume of archive storage.

It is difficult to provide a general scheme for copying nine-track tapes to DAT tapes, because of the variation in the programs that wrote the original tapes.

If the original tapes were written by DUMPER, then use DUMPER to restore the data on tape to disk. Then use DUMPER to save the disk data on a DAT tape. (A separate structure devoted to the tape copying operation may facilitate this transfer: create the structure anew for each save-set that you want to copy. One modern disk drive has sufficient capacity to hold any but the largest of the older TOPS-20 structures. By recreating the structure for each new save-set that you copy, you automatically destroy all files and directories that were created by the previous save-set.) It may be possible to save multiple save-sets on the same DAT tape, to achieve savings in storage space and media costs. (Writing multiple save-sets on the new medium is a particularly good idea if you are using a DLT.)

If the original tapes were written by BACKUP, similar ideas apply. Use DUMPER in “interchange” mode to read such tapes. We recommend that DUMPER in “native” TOPS-20 mode be used to write the data.

If you have tapes in some other format, you may need a special program to read the tapes. If the tape files can fit in a TOPS-20 file, read the tape into files and use DUMPER to save them on new media. (The maximum capacity of a single TOPS-20 file is 262144 disk pages, or about 600 megabytes. That is somewhat larger than the capacity of two 3600 foot reels of nine-track tape.) If tape files do not fit into TOPS-20 files, you will need a special program that copies from tape(s) to tape(s).

5.6 The OPR Program

OPR is a general operator interface program which is used to communicate with GALAXY system components; OPR is also used to control other system functions. The commands available are described in Appendix B. This program is used mostly to deal with batch streams and printer queues, although it also has facilities for handling operations that must deal with labeled industry-compatible tape volumes.

5.7 Other Periodic Tasks

This section describes some tasks that should be performed periodically.

5.7.1 Accounting Data

TOPS-20 collects accounting data from which the system manager can determine which users are consuming CPU and related resources.

Data is recorded in `BS:<ACCOUNTS>SYSTEM-DATA.BIN`, the accounting file, for such events as log in, log out, printing, checkpoints, and system reloads. Data is logged to the accounting file via the `USAGE JSYS`.

The accounting file is opened automatically by TOPS-20 when it is initialized. The file remains open throughout the time TOPS-20 is running. The file can't be backed up to tape, because it can't be accessed by another process while TOPS-20 has it open. The file can't be read or deleted while TOPS-20 has it open.

In order to back up the file, or process accounting records, or to delete it, the file must be closed (whereupon, TOPS-20 will open a successor file with a higher generation number). The procedure detailed below will close the accounting file. We advise you to put this procedure in a `.MIC`, `.CMD`, or other command file. Probably it is sufficient to perform this procedure weekly or monthly. You may find it helpful to perform this as part of back up procedure for the boot structure.

```

@reset
!enable
!XDDT
1/ .USCLS
USAGE ESC X      Note: the ESC will echo as "$"
~Z
!disable

```

The contributed program `FREP` provides an example of processing the binary records found in the accounting file. The source for `FREP` is found in the `<SRC.SU-UTILITIES>` directory.

5.7.2 The `BS:<SPOOL>` directory

A variety of files can be found in the `BS:<SPOOL>` directory.

Among these are files sent to device `LPT:`, the spooled line printer (or to device `LPTn:` for spooled printer unit n). Since there is no real line printer, files to be printed are held here until they can be sent through the network to whatever printers are accessible. (Files directed to `LPT:` or `LPTn:` get the default forms, thus only one type of data can be sent to a particular printer unit.

The file `ORION-SYSTEM.LOG` is the current log file for the program `ORION`. This file contains a log of system reloads, etc. The files `ORION-SYSTEM-LOG.nnn` are old versions of `ORION`'s log. Generally, they have no intrinsic value: they are a record of requests to the operator and operator responses. Sometimes problems in queue management can be discovered and corrected by reference to this log.

A new log is started when the `CLOSE LOG` command is given in the `OPR` program.

5.7.3 The `SERR:ERROR.SYS` file

This file contains a log of system errors (monitor “bug halts”, “bug checks” and “bug infs”), device errors, device statistics, and reload information.

The program `EP` reads this file and translates it to a human-readable format.

Chapter 6

Running the TOPS–20 UETP Package

This chapter describes how to run the User Environment Test Package (UETP). This program performs a cursory check of the system.

UETP starts a number of tests that can run either once or repeatedly. The source files are kept in the directory <UETP.LIB>, while the temporary data files and results of the tests are stored in the directory <UETP.RUN>. For more information on UETP, see the *User Environment Test Package Reference Manual*.

Note

You should perform all the steps in this chapter on the operator's console terminal (CTY). Otherwise, you must return to the CTY after completing the steps in this chapter to resume control of the system.

6.1 Obtaining a Tape Drive

This section describes how to obtain a tape drive for diagnostic purposes or for loading system software. When you are done with the tape drive, release it for timesharing use by following the procedure described in Section 6.5.

Log in on the console terminal (CTY) as a privileged user; enable Wheel or Operator privileges.

6.1.1 Systems without Tape Drive Allocation

If your system does not use tape drive allocation, use the following command to find a tape drive that is available to your job:

```
!information available devices 
DSK, TOED, MISC, DX, SHIP, MTA1, LPT, LPT0, LPT1, LPT2, LPT3, LPT4,
LPT5, LPT6, LPT7, LPT8, LPT9, LPT10, LPT11, LPT12, LPT13, LPT14, LPT15,
LPT16, LPT17, LPT18, LPT19, CDR, PTY6-23, NUL, TCP, PIP
Devices assigned to/opened by this job: TTY0
!
```

This typeout indicates that **MTA1:** is available to your job. Use the **ASSIGN** command to secure it for your job:

```
!assign mta1: 
!
```

Continue at Section 6.1.3

6.1.2 Systems with Tape Drive Allocation

If your system uses tape drive allocation, follow this procedure to obtain a tape drive for your use.

Type **OPR** and to run the OPR program:

```
!OPR 
OPR>
```

To find a tape drive that you can use for mounting the tape, type the **OPR** command:

```
OPR>SHOW STATUS TAPE-DRIVE /ALL 
```

The system responds with a list of tape drives and their status.

```
OPR>show status tape-drive /all 
OPR>
15:58:35          --Tape Drive Status--
 Drive State      Valid  Req#  Job#  User
-----
MTA0:  Unloaded
      AVR: Yes
MTA1:  In Use     DEK    31    45    NHELLIWELL
      AVR: Yes, Write: Yes, Unlabeled
MTA2:  Loaded
```



```

        AVR: Yes, Write: Yes, Unlabeled
MTA3:  Unloaded
        AVR: Yes
OPR>

```

Before mounting the tape, you should first make a free drive unavailable to timesharing users. Use the command:

```

OPR>SET TAPE-DRIVE MTA $n$ : UNAVAILABLE RET

```

to do this, where n represents the unit number of an unloaded unit. The example uses drive number 0, but you can specify any drive that is not in use:

```

OPR>Set tape-drive mta0: unavailable RET
Enter text and terminate with ^Z

```

When you set a tape drive unavailable, the OPR program wants you to record the reason for your action. Type an appropriate reason; finish by typing <CTRL/Z>:

```

Enter text and terminate with ^Z
Using tape for diagnostics ^Z
OPR>
15:59:07          -- Tape Drive Set Unavailable --
                MTA0: set unavailable for assignment by MOUNTR
SJ  0:
SJ  0: 15:59:07   -- Tape Drive Set Unavailable --
SJ  0:          MTA0: set unavailable for assignment by MOUNTR
OPR>

```

Note

Some of the output depicted above comes from OPR; some comes from job 0, SYSJOB, which is using the console terminal for its output.

Use the **Exit** command to leave the OPR program:

```

OPR>exit RET
!

```

Setting the tape drive “unavailable” means that TOPS-20 will not use that tape drive to satisfy any user’s Mount request. However, you still must assign the drive to your job to assure that no other user can access the tape drive.

Use the **Assign** command to assign the selected magnetic tape drive to your job. The example uses drive number 0, but you should use the drive that you set unavailable. The EXEC just prints its prompt when the drive is assigned:

```
!assign mta0: RET
!
```

Error: If the device is assigned to another job, the system prints the message:

```
?MTA0: Already assigned to job n
```

Go find a different drive that is unloaded.

6.1.3 Using a Logical Name for the Tape Drive

This section is optional. You may consider following this procedure if the tape unit that you are using is *not* **MTA0**.

Most of this manual is written as though the tape unit that you are using is **MTA0**. Although you will be reminded throughout to use the actual name of the tape unit you have assigned, it may be easier to give the following command:

```
!Define MTA0: MTA n: RET
```

where *n* is the actual unit number of the tape you have assigned, e.g.,

```
!Define MTA0: MTA2: RET
```

This command creates a *logical name*, **MTA0**, that the system associates with the actual unit **MTA2**. This name applies only to your current job.

By the way, the command

```
!Define MTA0: RET
```

erases the definition of the logical name **MTA0**.

6.2 Running the Standard Tests

Log in at the console terminal as a privileged user and enable your capabilities.

Use the **Take** command to read commands from **PS:<UETP.LIB>SET-UP.CMD**. This command file contains all the commands necessary to create the subdirectories needed by UETP.

```
!TAKE PS:<UETP.LIB>SET-UP.CMD RET
[New]
[New]
.
.
```

```
End of <UETP.LIB>SET-UP.CMD.1
!
```

Connect to the directory that contains the standard tests:

```
!Connect PS:<UETP.RUN> RET
!
```

To start the UETP program, use the Run command. The program prints the prompt “UETP>”:

```
!RUN UETP.EXE RET
[dd-mmm-yy hh:mm:ss      User Environment Test Package ]
UETP>
```

Error: If UETP stops with the message “?You must have IPCF privileges ...”, use the EXEC’s BUILD command to add the IPCF capability to your directory. After doing so, you will have to log out and log in again to make the privilege effective. Start over at the beginning of this section.

Use the Take command to load the standard verification tests from the file **VERIFY.CMD**. This command causes the standard tests to be loaded and the first test (VERIFY) to begin. The program prints several messages followed by status information. In the example, the field *hh:mm:ss* indicates the time the message was generated.

```
UETP>Take VERIFY.CMD RET
UETP>ENABLE VERIFY/CYCLE:1      Verify that the correct software
[hh:mm:ss  ENABLE COMPLETED] is installed.
UETP>BEGIN
[hh:mm:ss  BEGIN COMPLETED]
UETP>DEFAULT/CYCLE:00:15
[hh:mm:ss  DEFAULT COMPLETED]
UETP>ENABLE RANCBL
[hh:mm:ss  ENABLE COMPLETED]
UETP>ENABLE RANFOR
[hh:mm:ss  ENABLE COMPLETED]
UETP>ENABLE MTA0
[hh:mm:ss  ENABLE COMPLETED]
UETP>STATUS
[dd-mmm-yy hh:mm:ss]
  Test  Depth Status  Cycle Times Error Error Start
  name          run   count limit time
=====
VERIFY VER   Running    1     0     0     0 dd-mmm-yy hh:mm:ss
RANCBL VER   Enabled   0:15    0     0     0
RANFOR VER   Enabled   0:15    0     0     0
```

```
MTAO  VER  Enabled  0:15    0    0    0
START VERIFY  hh:mm:ss
END   VERIFY  hh:mm:ss
```

Note

If any messages appear between the “Start” message and the “End” message, there are errors in the files you received. Do not proceed any further.

The last two messages signify the beginning and ending of the test. If the first test (VERIFY) finishes without error, go on.

Note

In the following steps, you should use a tape drive that you assigned to your job by having followed the procedure in Section 6.1, page 63.

The standard tests include a random I/O Cobol test, a random I/O Fortran test, and a test called “MTAO”, which tests the tape drive `MTAO:`. The tape test requires that a “scratch” magnetic tape be mounted on drive 0. If you are using a tape drive other than `MTAO:`, you must disable the test named `MTAO` and enable the test of the other unit. To enable the test for a specific tape drive numbered “*n*”, type:

```
UETP>Enable MTAn RET
```

For example, to disable the test of drive 0 and to enable the test of drive 2 you would type:

```
UETP>Disable MTA0 RET
[hh:mm:ss  DISABLE COMPLETED]
UETP>Enable MTA2 RET
[hh:mm:ss  ENABLE COMPLETED]
UETP>
```

Select a “scratch” magnetic tape (the medium) that does **not** contain any information that you want to keep: the diagnostic will write on the tape. Now, mount the tape on the drive you want to test. If the tape drive is of the reel-to-reel variety, be sure that a write ring has been inserted on the tape; otherwise, be sure that the tape cartridge is write-enabled. Make sure that the tape drive is on line before proceeding to the next step.

To start the remaining tests, type the command `Begin`. The program will then print a series of messages:

```
UETP>BEGIN RET
[hh:mm:ss  BEGIN COMPLETED]
UETP>
```

```

START  RANCBL  hh:mm:ss
START  RANFOR  hh:mm:ss
START  MTA2    hh:mm:ss
END    RANCBL  hh:mm:ss
END    RANFOR  hh:mm:ss
END    MTA2    hh:mm:ss

```

If messages other than those shown in the example appear, the tests generating the messages have failed.

When all tests are completed, the system prints the following status information:

```

[All tests complete on processor # 2102]
[dd-mmm-yy hh:mm:ss]
  Test  Depth Status  Cycle Times  Error Error Start
  name          run    count limit time
=====
VERIFY VER  Ended    1      1      0      0 dd-mmm-yy hh:mm:ss
RANCBL VER  Ended   0:15    1      0      0 dd-mmm-yy hh:mm:ss
RANFOR VER  Ended   0:15    1      0      0 dd-mmm-yy hh:mm:ss
MTA2   VER  Ended   0:15    1      0      0 dd-mmm-yy hh:mm:ss

```

At this point, you can either type **Exit** to terminate UETP, or begin the tests for optional software. The following step shows how to terminate UETP. If you want to test optional software, you should skip this step and proceed to the next section.

Type “Exit” to stop UETP:

```

UETP>Exit RET
!

```

Clean up the temporary directories created by UETP by following the steps in Section 6.4.

6.3 Running Optional Tests

This section describes how to enable tests for optional software products. Be sure to enable tests only for the optional software products you have installed.

Table 6.1 contains a list of optional software products for TOPS-20 and the UETP command needed to start the test for the product.

To start a test for an optional software product, do the following:

1. Type an **Enable** command to select the test of a particular software product:

```

UETP>Enable product_name RET

```

See Table 6.1 for the correct command for your specific product, for example,

```
UETP>Enable FORTRA RET
[hh:mm:ss ENABLE COMPLETED]
UETP>
```

2. Type an **Enable** command for each product you wish to test.
3. Type the **Begin** command to start the tests:

```
UETP>Begin RET
[hh:mm:ss BEGIN COMPLETED]
UETP>
```

The **Begin** command starts the standard tests and all the optional software tests you enabled. The UETP starts all the tests immediately.

Periodically give a **Status** command to find out which tests are running and how many times they have been run:

```
UETP>Status RET
[dd-mmm-yy hh:mm:ss]
  Test  Depth Status  Cycle Times  Error  Error  Start
  name                               run   count limit  time
=====
VERIFY VER   Ended      1      1      0      0 dd-mmm-yy hh:mm:ss
RANCBL VER   Running    0:15    0      0      0 dd-mmm-yy hh:mm:ss
RANFOR VER   Queued     0:15    0      0      0 dd-mmm-yy hh:mm:ss
MTA2  VER   Queued     0:15    0      0      0 dd-mmm-yy hh:mm:ss
```

In the example, the **RANCBL.SUP** test is running, the **RANFOR.SUP** and **MTA2.SUP** tests are in a wait state, and the **VERIFY.SUP** test has ended.

Note

If there is a number other than 0 (zero) in the column “Error Count”, print the **ERRORS.LOG** file, when the tests are finished, to find out what caused the error.

The UETP program does not notify you when all the tests are completed. To find out when all the tests are completed, give a **Status** command. When all the tests are completed, your output from the **Status** command resembles the following:

```
UETP>Status RET
[dd-mmm-yy hh:mm:ss]
  Test  Depth Status  Cycle Times  Error  Error  Start
  name                               run   count limit  time
=====
VERIFY VER   Ended      1      1      0      0 dd-mmm-yy hh:mm:ss
```

```
RANCBL VER Ended 0:15 1 0 0 dd-mmm-yy hh:mm:ss
RANFOR VER Ended 0:15 1 0 0 dd-mmm-yy hh:mm:ss
MTA2 VER Ended 0:15 1 0 0 dd-mmm-yy hh:mm:ss
```

Note that all tests are reported as “Ended”.

To exit from the UETP program, type the Exit command. The EXEC prints its enabled prompt “!”:

```
UETP>Exit RET
!
```

Error: If you try to exit while there are tests still running, UETP will print the message
 %You still have tests running. "ABORT ALL" before exiting.
 Wait until all the tests are completed and then reissue the **Exit** command.

6.4 Deleting Temporary Directories

Perform the operations in this section to delete the temporary directories created by UETP.

To delete the temporary directories, use the command:

```
!Take <UETP.LIB>CLEAN-UP.CMD RET
```

The system prints the word [Old] for each directory it deletes:

```
!TAKE <UETP.LIB>CLEAN-UP.CMD
[Old]
[Old]
.
.
.
End of <UETP.LIB>CLEAN-UP.CMD.3
!
```

6.5 Release the Tape Drive

This section describes how to release a tape drive so that it can be used by other timesharing users. Before doing this, be sure you have unloaded the tape that you were using.

Type the EXEC command:

```
!Deassign MTA0: RET
```

to release the magnetic tape drive for use by other users. The EXEC prompts again.

If your system uses tape drive allocation, return the tape drive to the pool of allocatable units. Run the OPR program and enter the command to make the drive available for allocation:

```
!Opr   
OPR>Set tape-drive mta0: available   
OPR>Exit   
!
```


Table 6.1: UETP Commands for Optional Software Product Tests

Product	UETP Command
Algol-20	Enable Algol
APL-20	Enable APL
Basic-Plus-2	Enable Basic
Cobol-68	Enable Cobol
Cobol-74	Enable Cbl74
DBMS-20	Enable DBMS
Fortran	Enable Fortra
Sort-20	Enable Sort

Appendix A

7-CONFIG.COMD Commands

This appendix describes the system configuration command file, **SYSTEM:7-CONFIG.COMD**, giving a detailed list of the configuration commands and their parameters. This appendix describes how to declare the system name, change the system defaults for system devices, accounting, performance improvements, scheduler controls, file archiving and migration, and control other system functions.

Each time the system starts, it runs the SETSPD (Set System Parameter Defaults) program automatically. SETSPD reads the definitions found in the configuration command file, **SYSTEM:7-CONFIG.COMD**¹. Commands appear in this file to specify details of how TOPS-20 should operate; some of these must be customized to your particular environment.

Among the characteristics defined in the configuration command file are

- Logical names
- Magnetic tape drives
- Line printers
- Local time zone
- Daylight savings time
- Login structures
- Directory parameter setting
- Account validation
- Accounting shift changes
- Password management
- Security controls
- Performance improvements
- Scheduler controls
- Tape drive allocation

¹ When SETSPD is run for the first time, the logical name **SYSTEM:** will be defined as **BS:<NEW-SYSTEM>,BS:<SYSTEM>**.

- System level messages
- Network information
- File archiving and migration
- Off-line structures
- Memory dumps
- Cluster data gathering
- MSCP-served disk drives

The file can include comments. Comments are preceded by the exclamation point “!” character and they continue through the end of the line.

Your system comes with a sample `BS:<SYSTEM>7-SETSPD.COMD` file which you should edit to adapt to your site’s requirements.

A.1 Defining System–Wide Logical Names

See the *TOPS-20 System Manager’s Guide* for a description of logical names.

Insert a command in the following format to define any special system–wide logical names that you require. These logical names are defined when the system starts; thereafter, they are available to all users. The command format is shown here:

```
DEFINE name: definition_list RET
```

The initial `7-CONFIG.COMD` file contains definitions that have been crafted to work reasonably well, whether or not a separate login structure is used. Most of the definitions reference the structure name `BS:` which is itself the system–wide logical name for the *boot structure*. The boot structure is the structure from which `TDBOOT` will load the monitor. The boot structure is also the structure on which `TOPS-20` does swapping; for this reason this structure is sometimes called the *swapping structure*. Further, the boot structure is also the structure on which `TOP-20` expects to find the executable files containing system programs (and on which it expects to log accounting data, errors, spool requests, etc.); for this reason, the structure is also known as the *system structure*.

The definitions in your initial `7-CONFIG.COMD` file assume that all system software will be located in directories on `BS:.` The *public structure*, known by the logical name `PS:`, is the structure on which users’ login–directories (and mail) are found. `PS:` and `BS:` are the same structure unless you define a separate login structure.

A.2 Selecting an Editor

You have the option of selecting the default editing program (editor) with which your users create and edit files using the `Create` and `Edit` commands. As delivered to you, the default editor is the EMACS screen–editor. Other editors are available. Digital Equipment provided two editors for

TOPS-20: EDIT, a line-oriented editor, and TV, a screen-oriented editor. If you want users to use, e.g., the TV program, replace the existing definition of **EDITOR:** by the command line:

```
DEFINE EDITOR: SYS:TV.EXE
```

in the configuration command file.

By defining the logical name **EDITOR:** for their own jobs, users can override your choice of default editor, as long as the editor they specify is available at your installation.

If you want your system's default editor to be the EDIT program, remove the definition of **EDITOR:** from the configuration command file.

A.3 Defining Magnetic Tape Logical Unit Numbers

TOPS-20 identifies each magnetic tape drive by a *unit number*. The unit number forms part of the name by which users refer to a particular unit, e.g., **MTA0:** is the name of the first tape, **MTA1:** is the name of the second, etc. Unit numbers are assigned by the system based on the channel number, controller number, and the target number of the tape drive.²

The magnetic tape drive with the lowest channel, controller, and unit numbers (with “channel” taken as the most significant figure and “unit” as the least significant) is identified by TOPS-20 as **MTA0:**.

Sometimes this method is inconvenient to users because if the system were recabled, the physical drives could become known by different device names. Such a change would invalidate batch files and documented procedures. So, TOPS-20 provides a means by which the system administrator can assign particular device names to particular tape units. For some units, this method, once established, is automatic; for other units, the method requires alteration of the configuration file when recabling occurs. The commands in the configuration file give you the ability to keep drive unit device names constant, in spite of recabling.

Use the (privileged) program SYSDPY to display information about your SCSI devices including the tape drives. Be enabled; use the command:

```
!Sysdpy ducserial/1cmaker/2 RET
```

to start the SYSDPY program, in the “Display Units” (du) mode, with the serial number display moved to column 1 and the maker and model number display moved to column 2.

An example of what you might see on your screen is presented in Figure A.1.

This display shows that the system has two tape units. **MTA0:** is at Channel 2, Controller -1, and Unit 3; it has no intrinsic serial number, so TOPS-20 has invented one for it, based on its channel, controller, and unit numbers. **MTA1:** is also on Channel 2 and Controller -1, but it is unit number 5, and its serial number is the 10-character string “ZG52401386”.

Write down the information you need: for each unit, record its 10-character serial number, and its

²The TOPS-20 scheme of channel, controller and unit numbers is explained in more detail in Section 5.3.1, page 42.

Figure A.1: Display Units, DU, from the SYSDPY Program

Serial Num	Maker	Model	CKU	DB-Addr	Chn	Kon	Unt	Device	Type	Cyl-F
			C	1666571	0				XRH	
00050221	SEAGATE	ST12550W	U	1670001	0	-1	5	TOED	Disk	96
3401A25191	HP	C2490A-300	U	1670461	0	-1	15	MISC	Disk	156
			C	1670671	1				XRH	
3404A61898	HP	C2490A-300	U	1671521	1	-1	7	DXX	Disk	133
3404A72924	HP	C2490A-300	U	1672251	1	-1	11	SHIP	Disk	131
			C	1673001	2				XRH	
XKL-C02U30	HP	C1533A	U	1673331	2	-1	3	MTA0	Tape	
ZG52401386	Quantum	DLT4000	U	1674251	2	-1	5	MTA1	Tape	26214
			C	1673561	3				XRH	
			C	1675251	4				XNI	

This display is obtained by the command:

```
!Sysdpy ducserial/1cmaker/2 RET
```

24-character maker and model. To exit from SYSDPY, type the letter "E" and RET.

Use the **SCSI-TAPE** command in the configuration file to identify these units. The **SCSI-TAPE** command requires a unit number, a 24-character manufacturer and model string, and a 10-character serial number string. The strings are enclosed in quotation marks, as shown in this example:

```
SCSI-TAPE 0 "HP      C1533A      " "XKL-C02U30"
SCSI-TAPE 1 "Quantum DLT4000    " "ZG52401386"
```

If a third tape were added to the system and put on channel 1, TOPS-20 would normally give it the device name **MTA0:** and change the names of the other drives accordingly. However, with these commands present in the configuration file, the original two units keep the names they were known by; the new tape drive becomes **MTA2:**.

The use of the **SCSI-TAPE** command to assign unit numbers to specific tapes is handy also in systems that have XTU-1 external reel-to-reel tape units. These units have individual power switches, and they need not be turned on when the system is started. If turned on subsequent to TOPS-20 becoming operational, TOPS-20 will recognize the tape and add it to the system. Meanwhile, by use of the **SCSI-TAPE** command, the other tape drives have already been assigned their accustomed names.

Notes

A unit serial number is a 10-character string. Be sure to enter all ten characters, including leading or trailing blanks, if needed. In the example above, the serial number of the disk, unit 5 on channel 0, is “00050221 ”, which includes two trailing blank spaces.

The maker and model string in the command parameter must be 24 characters. You must put blank spaces in the parameter precisely where they appear in the SYSDPY screen.

Tape units that have no intrinsic serial numbers (reported as “XKL-C....”) must be handled specially: if you recable your SCSI equipment, you will have to update the command file manually.

Operational Note

If you make use of the `SCSI-TAPE` command, be aware that the `TDBOOT` program will **not** know about your assignments of new names to the various tapes. Thus, any time you must refer to a tape in `TDBOOT`, please double check to be sure that you and `TDBOOT` are using the same names.

A.4 Defining the Local Time Zone

The time zone in which your installation is located can be represented as the number of hours west or east of Greenwich. For example, Eastern Standard Time is zone 5; Central Standard Time is zone 6; Mountain Standard Time is zone 7; and Pacific Standard Time is zone 8. To set your local time zone, you can enter the following command in the configuration command file:

```
TIMEZONE n
```

where *n* is a decimal number between -12 and 12 inclusive. West of GMT is positive; east is negative. Note that this is the **reverse** of the usual scheme of numbering timezones: U.S. Eastern Standard Time, “GMT-5”, is represented in this command as +5. Zones -12 and 12 represent the same clock time but on opposite sides of the international date line. If you do not specify a time zone in the configuration command file, the time zone will default to zone 0, signifying British time.

A.5 Specifying Daylight Savings Time

You can specify one of three Daylight Savings Time³ (DST) options in the configuration command file. The three options are:

³Known also as “Summer Time” in parts of the English-speaking world.

- Automatic** Change at the time now prescribed by law in the USA. Turn on DST at the first Sunday in April and return to standard time at the last Sunday in October.
This is the most convenient option in most states of the USA (excepting Arizona, Hawaii, and parts of Indiana), subject to the whims of Congress.
- Off** Do not do DST conversions.
Use this option if DST is never observed, or during Standard Time in countries where the “Automatic” algorithm does not apply.
This option is appropriate for GMT, Greenwich Mean Time, (also known as UT, Universal Time) which does not change with the season. This is also appropriate for tropical areas where DST is not observed.
- On** Always do DST conversions.
Use this option during Daylight Savings Time (Summer Time) in countries where the “Automatic” algorithm does not apply.

To specify the DST option for your system, type the command **DAYLIGHT** followed by the appropriate argument. The following is only an example. You must determine which option is appropriate for your site.

DAYLIGHT ON

Note

If the “Automatic” setting does not apply, the configuration file must be edited explicitly and **SETSPD** must be run manually (see Section A.24, page 94) to adjust the system time for the shift (in either direction) between Standard Time and Daylight Time.

A.6 Directory Parameter Setting

You have the option of allowing users to change their directory parameters. If you decide to allow users to do this, no command is necessary in the configuration command file, because the system default is to:

ENABLE DIRECTORY-PARAMETER-SETTING

However, if you decide not to allow users to change their directory parameters, you should put the command:

DISABLE DIRECTORY-PARAMETER-SETTING

in the configuration file. When this command is given, it prevents the user from changing any of his or her directory parameters unless he has “Wheel” or “Operator” privileges.

Note

If the default is taken, users are allowed to change some of their directory parameters with the EXEC's

@SET DIRECTORY ...

commands.

A.7 Account Validation

You have the option of requiring all users to enter a valid account name when they are logging in to the system. If you decide to require valid accounts, you must remove the command:

DISABLE ACCOUNT-VALIDATION

from the configuration file; absent that command, the system will default to enable account validation.

The system is delivered with account validation turned off because, prior to turning on account validation, you must create the necessary account data files and run the account generator program (ACTGEN) to create your account data base file, **BS:<SYSTEM>ACCOUNTS-TABLE.BIN**.

With account validation disabled, the system checks each account string 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.

More information concerning accounts is contained in the *TOPS-20 System Manager's Guide*.

A.8 Performance Improvements

A.8.1 Working Set Swapping

If you add the command:

ENABLE WORKING-SET-PRELOADING

to the configuration file, the entire working set of a process is brought into memory at once when it is the process's turn to run. Otherwise, this action occurs on a page-by-page basis.

Note

You should enable working set swapping only if your system runs large compute-bound jobs. However, if your system runs many interactive jobs or if you are unsure which description better fits your system, take the default:

DISABLE WORKING-SET-PRELOADING

Preloading the working-set is more helpful in systems that have small main memory configurations. This is not likely the case in a TOAD-1 System.

A.9 Scheduler Controls

The following sections describe the bias control, traditional scheduling, and class scheduling.

A.9.1 Bias Control

You have the option of deciding how your system's CPU time is to be divided between interactive and batch jobs. You do this by entering the command:

BIAS *n*

in the configuration command file, where *n* is any integer between 1 and 20.

The smaller the value of *n*, the larger the percentage of CPU time that is given to interactive jobs. If you do not give this command, a default value of 11 will be taken, which provides a balance between good interactive response and reasonable throughput.

A.9.2 Traditional Scheduling

Traditional scheduling does not distinguish between classes of users; it generally treats in similar ways all users who are doing similar things. Traditional scheduling does distinguish CPU-intensive jobs from interactive jobs (and may favor one or the other, depending on the Bias setting), and it can be told to reduce the priority of batch jobs relative to jobs controlled by a person at a terminal.

If you want to use traditional scheduling, you do not have to enter any command into the configuration command file, because the system default is to

DISABLE CLASS-SCHEDULING

Scheduling Low Priority for Batch Jobs

Although you must use the class scheduler to give batch jobs a high priority, you can give batch jobs a lower priority using the traditional scheduler. Add the command:

BATCH-BACKGROUND

to the configuration command file.

This command specifies that all batch jobs run on the lowest priority queue, also known as background queue. This means that after processing all interactive jobs, the scheduler selects and runs batch jobs waiting in the queue.

Do **not** use this command if you enable the class scheduler.

A.9.3 Class Scheduling

You can use class scheduling to divide your system's CPU time among different classes of jobs. Class scheduling assigns percentages of CPU time to jobs according to their account or according to a policy program written especially for your site.

If you use either kind of class scheduling, you can allocate any unused CPU time (windfall) among active jobs, or withhold it from all jobs.

A.9.3.1 Class Scheduling by a Policy Program

You must have a policy program in order to use either of the following commands.

The command for enabling class scheduling using a policy program and allocating windfall is

ENABLE CLASS-SCHEDULING POLICY-PROGRAM ALLOCATED

The command for enabling class scheduling using a policy program and withholding windfall is

ENABLE CLASS-SCHEDULING POLICY-PROGRAM WITHHELD

If you enable class scheduling using a policy program, you cannot also enable class scheduling using accounts: you should skip to Section A.10 after giving either of these two commands.

A.9.3.2 Class Scheduling by Accounts

If you do not have a policy program to use with class scheduling at your site, you can still enable class scheduling by assigning jobs to classes according to the account under which each job is running. (See the *TOPS-20 System Manager's Guide* for information about associating accounts with particular classes.) To do this you must first enter **Create** commands into the configuration command file, specifying the percentage of CPU time that is to be given to jobs in each class.

You must enter a **Create** command for each class that your system will use, with an appropriate percentage for each class. The sum of the percentages you specify must not add up to more than 100%, although they may add up to less than 100%. If they add up to less than 100%, the percentage of CPU time that is not assigned becomes part of the system's windfall. Each **CREATE** command is a line in the configuration file.

The commands shown in the following example divide 75% of the system's CPU time among three classes of jobs. For example, the command

```
CREATE 1 .20
```

specifies that jobs running under accounts in class 1 are to be given a total of 20% of your system's CPU time. You must supply the correct class numbers and percentages for your own system.

```
CREATE 1 .20
CREATE 2 .40
CREATE 3 .15
```

Batch jobs can be put into a special class of their own by including the command:

```
BATCH-CLASS n
```

where n is the number of the class you choose for batch jobs. If you do not enter a **BATCH-CLASS** command, batch jobs are given CPU time according to the class associated with the account of the job's owner at the time of submission. The example below creates a class for batch jobs and then assigns 25% of CPU time to this class with the **CREATE** command. You must enter commands that are appropriate for your system.

```
BATCH-CLASS 4
CREATE 4 .25
```

Now you must enter the command that enables class scheduling using accounts. You must also decide whether to allocate or withhold any windfall CPU time. The default for this command is **ALLOCATED**. It is recommended that you take this default. Enter the command:

```
ENABLE CLASS-SCHEDULING ACCOUNTS ALLOCATED
```

Caution

Do not enable class scheduling without first creating classes as described above. Class scheduling will not function properly without classes.

A.10 File Archiving and Migration Parameters

The following sections describe tape recycle periods for use with the file archiving and migration features.

A.10.1 Archive Tape Recycle Period

TOPS-20 provides a feature that allows off-line storage of selected files on magnetic tape for long periods of time. You can specify how long these files are to be saved by inserting a command into the configuration command file stating, in days, the recycle period of the tapes. For example, to save archival files on tape for five years, you would enter the command:

```
ARCHIVE-TAPE-RECYCLE-PERIOD 1825
```

(because 5 years at 365 days/year is 1825 days).

If you do not insert an `ARCHIVE-TAPE-RECYCLE-PERIOD` command in the configuration command file, a default value of 3650 (ten years) will be taken.

A.10.2 Migration Tape Recycle Period

TOPS-20 provides a feature that allows automatic off-line storage of old or little-used files on magnetic tape for short periods of time. You can specify how long these files are to be saved by inserting a command into the configuration command file stating, in days, the recycle period of the tapes. For example, to save migrated files on tape for three months, you would enter the command:

```
TAPE-RECYCLE-PERIOD 90
```

(because 3 months with 30 days/month is 90 days).

If you do not insert a `TAPE-RECYCLE-PERIOD` command in the configuration command file, a default value of 180 (six months) will be taken.

A.11 Tape Drive Allocation

TOPS-20 provides for the automatic assignment of tape drives to users who wish to read or write magnetic tapes. In order to use this feature, which also allows your system to read and write the labels of standard labeled tapes, you must insert the command:

```
ENABLE TAPE-DRIVE-ALLOCATION
```

into the configuration command file. If you do not give this command, your users must assign tape drives with the EXEC's `Assign` command; their labeled tapes will be treated as unlabeled tapes.

Treatment of Unrecognized Tapes

If you enable tape drive allocation, you must decide what action the system should take if a labeled tape is mounted improperly (for example, if the user specifies the wrong label type or density). If you want the system to unload the tape so that it cannot be accidentally erased, you must add:

```
TAPE-RECOGNITION-ERRORS UNLOAD
```

to the configuration command file. If you do not give this command, the system will treat improperly mounted labeled tapes as unlabeled tapes; it will continue processing them. That is, the system default is

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

A.12 Accounting Shift Changes

The accounting facility allows you to provide varied billing rates for system use at different times. You do this by entering commands into the configuration command file that specify the time of day for each rate change. You should give one command for each time the rate changes. Use days of the week or the keywords **WEEKDAYS**, **WEEKENDS**, and **ALL**, and times in either 24-hour or AM/PM format. (See the *TOPS-20 System Manager's Guide* for detailed information about accounting shift changes.)

CHANGE commands are entered into the configuration command file to provide special rates for evenings and weekends. The following is only an example; you must provide the correct days and times for your own system.

```
CHANGE 8:00 WEEKDAYS
CHANGE 17:00 WEEKDAYS
CHANGE 0:00 SATURDAY
```

Note

You do not have to enter **CHANGE** commands into the configuration command file. If you do not enter any **CHANGE** commands, the accounting shift change feature of system accounting is not used.

A.13 Controlling System Message Levels

You have the option of notifying users of certain events, such as resource problems (level 0 messages) and operational events (level 1 messages).

The level 0 messages are

- [Caution - Swapping space low]
- [Caution - SPT space low]
- [Caution - Disk space low on system structure]

The level 1 messages are

- [System *name* going down in....]
- [Shutdown canceled]
- [Deleted files will be expunged from system structure in 30 seconds]
- [System structure expunge completed]

To allow the system to send either type of message to users, add the command **Enable** for each type. The **Disable** command will prevent the system from sending either type of message.

This example enables the system to send both types of system messages to users:

```
Enable System-Message-Level-0
Enable System-Message-Level-1
```

A.14 Network Information

Enter the following command line into your configuration command file in order to identify your system:

```
Node node_name
```

The *node_name* should match the local name given in **BS:<SYSTEM>HOSTNAME.TXT**.

A.15 Designating Off-line Structures

Note

The disks supported on the TOAD-1 System are not expected to go off-line. Hence, this facility is not expected to be of any practical use.

The off-line structure facility designates to the monitor which structures are offline. This prevents processes from accessing or opening files that are on disk units that the TOPS-20 monitor knows are offline. You can also set a timeout interval between the time the disk goes offline and the time when the monitor sets the structure off-line.

The command to enable the off-line structure facility is

```
Enable Offline-Structures mm:ss
```

This is the default (with a timeout interval of one minute). You need only to enter a command in the configuration command file if you want to change the timeout interval (between 1 and 15 minutes) or to turn the off-line structure facility off. The command to turn off the off-line structures facility is

```
Disable Offline-Structures
```

This example shows how to change the off-line structures timeout interval to five minutes:

```
Enable Offline-Structures 05:00
```

A.16 The Login Structure Facility

The login structure facility allows you to specify a structure other than the boot structure (the system structure) as the structure that TOPS-20 will use to locate user-names and login directories. That is, the login structure is the home for users' directories and files, distinct from the boot structure which is then used for swapping and storage of systems programs.

Any (one) structure that is available to the system at startup time can be defined to be the login structure. When a login structure is used, the logical name PS: is redefined so that it names the login structure. (In systems where a login structure is defined, the boot structure will contain a small number of login directories, e.g., Operator, for use only when the login structure is not mounted. In normal operation, Operator will log in to the login structure.)

There are at least two scenarios in which a login structure is useful. First, for efficiency of data back up, user data (which may be presumed to change rapidly) is segregated (on the login structure) from systems programs and less-frequently changing data (on the boot structure). Second, in a cluster environment (not yet supported by TOAD-1 System), where the same set of users may use any system in the cluster. In this case, user data (the login structure) is on a disk shared by all the systems in the cluster, and swapping space (which is private to each processor) is on the boot structure.

The command to enable the login structure facility is

```
ENABLE LOGIN-STRUCTURE
```

This command instructs the monitor to attempt to find a login structure when the system is starting. If this command is not issued, the monitor will not attempt to find a separate login structure: the boot structure will be used as the login structure.

The command to turn off the login structure facility is

```
DISABLE LOGIN-STRUCTURE
```

which bypasses any attempt to define a login structure that is separate from the boot structure. This is the default.

A.17 Preventing Fast Login

By using the `/FAST` switch with the `LOGIN` command, users can bypass processing of the `LOGIN.COMD` and `COMAND.COMD` files, which may be set up to limit users' computing environments. For example, sets of users may be allowed only to read mail or run some particular application program. You can prevent fast logins by entering the command:

```
DISABLE FAST-LOGIN-OPTION
```

in the configuration command file.

Fast login is in effect by default.

See the *TOPS-20 User's Guide* for information on the `LOGIN.COMD` and `COMAND.COMD` files. The *TOPS-20 Commands Reference Manual* describes the `LOGIN` command.

A.18 Password Management

Password management features allow you to check that users at your site do not choose passwords that are too short or that are otherwise easy to guess, such as a name or initials.

A.18.1 Designating Minimum Password Length

You can control the minimum length of each user's unencrypted password string. The minimum password length can be set to between 1 and 39 characters. To set the minimum password length always at system startup, enter the command:

```
Enable Minimum-Password-Length n
```

in the configuration command file.

A.18.2 Changing Passwords Regularly

You can control how often users must change their passwords. You can set the numbers of days a password is valid since the time it was last changed. You can set the number from 1 through 366. The default number of days is 30. To add this feature, you must insert the command:

```
Enable Password-Expiration n
```

into the configuration command file.

A.18.3 Disallowing Certain Passwords

You can 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. See the *TOPS-20 System Manager's Guide* for details on how to setup the file.

To add this feature, you must insert the command

```
Enable Password-Dictionary
```

into the configuration command file.

A.19 Dumping on Non-fatal System Errors

The monitor can dump its memory area to a disk file when BUGCHKs and BUGINFs occur. The dump-on-BUGCHK feature helps debug the system of non-fatal, “continuable” errors by providing a dump file for examination.

To enable the dump-on-BUGCHK feature, you must enter the command:

```
Enable Dump-On-Bugchk Facility
```

in the configuration command file. At least one of the following commands is further required to enable dumping of all or specific BUGCHKs and BUGINFs that can be dumped:

```
Enable Dump-On-Bugchk All Bugchks
Enable Dump-On-Bugchk All Buginfs
Enable Dump-On-Bugchk Bug bugname
```

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.

If the feature is disabled, the system produces only “crash” dumps associated with fatal errors; it does not produce any “continuable” dumps.

Dumps are taken only as often as the Dump-On-Bugcheck 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
```

A.20 Disabling the Console Terminal for Operator Output

The `Enable Job0-CTY-Output` command is in effect by default. It allows the operator to receive messages at the console terminal about system resources, such as when disk space is low on the system structure or when swapping space is low.

If you do not want these messages to be displayed on the console terminal, put the command:

```
Disable Job0-CTY-Output
```

in the configuration command file.

A.21 Enabling the Access Control Program

The TOPS-20 access control program (ACJ) helps you govern the access to many of the system's resources and services, reduce or prevent unauthorized access to the system, and investigate occurrences of unauthorized access. It can control scheduling classes, the bias control, batch background queue, logging in, use of physical resources (tape drives, terminals, structures), and enabling capabilities.

The command to enable the ACJ program is

```
Enable System-Access-Control-Job
```

See the *TOPS-20 System Manager's Guide* for more information on the ACJ program.

A.22 Setting Console Terminal Parameters

The command `Terminal` can be used to set some parameters of the terminals directly connected to the TOAD-1 System. Only the console terminal (and the auxilliary, which is either disabled or treated exactly the same as the console) is directly connected, so only the console can be affected by this command.

You can specify whether or not the console (and auxilliary) receives system messages or user messages. A line receives messages by default. To specify that the console is not to receive system or user messages use the either or both of the commands that follow.

To keep the line from receiving "Send All" messages (messages sent via the TTMSG JSYS addressed to all lines), use:

```
TERMINAL 0 IGNORE-SYSTEM-MESSAGES
```

To prevent the line from receiving messages sent via the TTMSG JSYS that are addressed specifically to this line, use:

```
TERMINAL 0 IGNORE-USER-MESSAGES
```

You can specify that input on the console terminal is to be ignored when the line is inactive. When input is being ignored and characters are typed, no <CTRL/G> (bell) is sent, as is the normal case when characters are typed on an inactive line. The command is

```
TERMINAL 0 NOBELL
```

(Other keywords are recognized in the **TERMINAL** command, but they have no defined effect in the TOAD-1 System.)

A.23 Commands not Supported in the TOAD-1 System

The following commands exist in SETSPD but they are not supported in the present TOAD-1 System implementation of TOPS-20.

A.23.1 Assigning MSCP-served Disk Drives

NOTE

The facility described in this section relates to CFS, the Common File System, which is not presently implemented in the TOAD-1 System. The effects of these commands in the TOAD-1 System is undefined.

MSCP (Mass Storage Communication Protocol) is responsible for granting access to MASSBUS disks across the CI (Computer Interconnect). To have MSCP grant access to disks across the CI, use the **Allow** command, in the following format, in the configuration command file:

```
Allow drive_type serial_number
```

drive_type is RP04, RP05, RP06, RP07, or RP20.

where: *serial_number* is the serial number of the disk drive.

To determine the disk drive's serial number, use the OPR command:

```
OPR>Show Configuration Disk RET
```

(See the *TOPS-20 Operator's Command Language Reference Manual* for information about this command.)

To disallow access to a served disk that was allowed access, enter the following command in the configuration command file:

```
Restrict drive_type serial_number
```

Disks are restricted by default if you do not specify **Allow** commands.

The **Dontcare** command specifies that no coordination is needed for the specified drive.

Enter the **Allow** command for each disk drive to be served by MSCP. The commands shown are only examples: you must determine the correct definitions for your system.

```
ALLOW  RP06  1243
ALLOW  RP20  3327
```

A.23.2 Cluster Data Gathering

Note

Clusters are related to CFS, Common File System, and are not implemented in the TOAD-1 System.

Cluster data gathering is a feature that allows systems in a cluster to easily share information from any system in the cluster. This feature enables users to obtain a **Systat** of the entire cluster or of a specified job on a node in the cluster. It also allows a user to obtain the system status of nodes in the cluster and to send messages to nodes in a cluster.

A.23.2.1 Cluster Information

You can allow information to be shared among nodes in the cluster. If you want cluster information to be shared, you do not have to enter any command into the configuration command file, because the system default is to

```
ENABLE CLUSTER-INFORMATION
```

However, if you decide not to share information with a node in the cluster, you must enter the command:

```
DISABLE CLUSTER-INFORMATION
```

in the configuration command file. (This feature is not totally turned off, since GALAXY components are allowed to gather remote information at any time.)

A.23.2.2 Cluster Sendalls

You can allow the local node to receive information from remote nodes in the cluster. This information includes class scheduling information, job information, and other miscellaneous information. If you want the local node to receive cluster sendalls, do not enter any command into the configuration command file, because the system default is to

ENABLE CLUSTER-SENDALLS

However, if you decide not to receive cluster sendalls, you must enter the command:

DISABLE CLUSTER-SENDALLS

in the configuration command file. (This feature is not totally turned off, since GALAXY components are allowed to perform cluster sendalls when necessary.)

A.23.3 The Magtape Command

The command **Magtape**, in DEC's TOPS-20 implementations, performs a function analogous to the **SCSI-Tape** command. The command parameters are different, so the TOAD-1 System has a different command.

A.23.4 The Printer Command

The command **Printer**, in DEC TOPS-20 implementations, performs various control functions on directly-connected printers. The TOAD-1 System has only network printers.

A.24 Running SETSPD Interactively

The SETSPD program can be run interactively by a privileged user:

```
!7-SETSPD 
Process 7-SETSPD.COMD (Y or N)?
```

Answer "Y" and if you want SETSPD to process your new configuration file. Or, as we do in this example, answer "N" to interact with the program. In the example, by typing "?", we ask SETSPD what commands it recognizes. Then, the example goes on to give a command and exit:

```
!7-SETSPD 
Process 7-SETSPD.COMD (Y or N)? n 
7-SETSPD$? one of the following:
ALLOW ARCHIVE-TAPE-RECYCLE-PERIOD
BATCH-BACKGROUND BATCH-CLASS
BIAS CHANGE
CREATE DAYLIGHT
DEFINE DISABLE
```

```

DONTCARE
EXIT
NODE
RESTRICT
TAPE-RECOGNITION-ERRORS
TERMINAL
7-SETSPD$Daylight On 
7-SETSPD$exit 
TOED:< >!

```

Note

Some commands recognized by SETSPD are not implemented in the TOAD-1 System version of TOPS-20. Among these are commands pertaining to CFS (the Common File Server) and MSCP (the Mass Storage Control Protocol).

Appendix B

OPR Commands

This appendix describes the commands available in the OPR program. The OPR program is the general operator interface to TOPS-20. Queues for printing and batch processing are managed through this interface. Service requests, from users or from the programs they run, are satisfied by operator intervention and response via this program.

Abort this command allows you to terminate a job request which is in progress on an input or output device or in a batch stream.

```
OPR>ABORT keyword argument { switch } RET
```

where *keyword* and *argument* can be one of the following:

- **Batch-stream** *stream_number*
- **Printer** *unit_number*

followed by zero or one of these switches:

- **/Request-id:** *request_id*
- **/Purge**
- **/Reason:** *one or more lines of text*
- **/Error-processing** (for batch-stream only)
- **/Noerror-processing** (for batch-stream only)

Align allows you to cause a “forms-alignment” file to be printed repeatedly on the line printer so that you can align the forms. (This is not of much use for page-description printers.)

```
OPR>ALIGN PRINTER unit_number (filespec) { switch } RET
```

the switches should be zero or more of the following:

- **/Stop**
- **/Pause:** *number_of_seconds*
- **/Repeat-count:** *number_of_repeats*

Backspace allows you to reprint previously output pages in the file currently printing on the line printer.

```
OPR>>BACKSPACE PRINTER unit_number { switches } RET
```

- /Copies: *number_of_copies*
- /File
- /Pages: *number_of_pages*

Cancel allows you to cancel any job request that is processing or is waiting to be processed, or any mount request for a tape or disk structure.

```
OPR>>CANCEL keyword argument(s) RET
```

- Batch-request { *request_id* | *user_name* | * } { /reason: *comment* }
- Mount-request { *request_id* | *structure_name* }
- Printer-request { *request_id* | *user_name* | * }

Close allows you to close the current ORION log file buffer and create a new log file buffer. The log file contains entries for each print job, batch job, and various operator interactions. The closed log file is written to disk so that an up to date file can be copied, backed up, or printed.

```
OPR>>CLOSE LOG RET
```

Continue allows you to continue a job request that was temporarily stopped with the **Stop** command.

```
OPR>>CONTINUE keyword RET
```

- Batch-stream *stream_number*
- Printer *unit_number*

Disable allows you to turn off various system processes as defined by the available keywords.

```
OPR>>DISABLE keyword RET
```

- Automatic-volume-recognition { Tape-drives | *tape_drive_name*: }
- Class-scheduler
- File-retrieval-waits
- Logging
- Output-display {All | Batch | File | Mount | MCP | Printer | User}
- Queue-requests

Dismount allows you to dismount structures and tapes currently known to the system.

```
OPR>>DISMOUNT keyword argument RET
```

- Structure *structure_name*:
- Tape-drive *tape_drive_name*:

Enable allows you to turn on various system facilities as defined by the available keywords.

```
OPR>>ENABLE keyword RET
```

- Automatic-volume-recognition { Tape-drives | *tape_drive_name:* }
- Class-scheduling /Class-assignments: {Account | Policy-program}
- Class-scheduling /Windfall: { Allocated | Withheld }
- File-retrieval-waits
- Logging
- Output-display {All | Batch | File | Mount | NCP | Printer | User}
- Queue-requests

Enter allows you to enter an alternative command processor for an application type.

OPR>ENTER *application_type* RET

where *application_type* must be defined by the GALGEN program.

Exit allows you to leave OPR command level and return to (enabled) EXEC command level.

OPR>EXIT RET

Forwardspace allows you to skip pages that are currently printing on the line printer.

OPR>FORWARDSPACE PRINTER *unit_number* { *switches* } RET

- /Copies: *number_of_copies*
- /File
- /Pages: *number_of_pages*

Help allows you to display the function, format, arguments, and switches of any OPR command.

OPR>HELP *keyword* RET

Hold allows you to keep particular requests in the queues from being processed. You must give the RELEASE command to allow the held jobs to be processed.

OPR>HOLD *keyword* { *request_id* | *user_name* | * } RET

- Batch-jobs
- Printer-jobs

Identify allows you to identify a tape mount to the system when either the tape is unlabeled or you have Automatic-Volume-Recognition disabled.

OPR>IDENTIFY MTAn: *keyword* RET

- Request-ID *request_id*
- Scratch
- Volume-ID *volume_name*

Modify allows you to change the priority of a job request waiting in a queue for processing. The higher the priority number, the sooner the job will be processed.

OPR>MODIFY *keyword* { *request_id* | *user_name* | * } { *priority 1-63* } RET

- Batch-request
- Printer-request

Push allows you to leave OPR command level and enter (a new) TOPS-20 EXEC command level. You must give the EXEC's **POP** command to return to OPR command level.

```
OPR>PUSH RET
```

Release allows you to release a job request that was held with the **Hold** command.

```
OPR>RELEASE keyword { request_id | user_name | * } RET
```

- **Batch-jobs**
- **Printer-jobs**

Report allows you to make journal entries into the system error file.

```
OPR>REPORT message RET
```

Requeue allows you to stop an “in progress” job request, put it back in the queue in a “hold” state, and cancel the request in progress. You must give the **Release** command to activate the job request after it has been requeued.

```
OPR>REQUEUE keyword argument { /REASON: comment } RET
```

- **Batch-stream** *stream_number*
- **Printer** *unit_number* { **Beginning** | **File** | **Job** | **Current** }

Respond allows you to reply to any message that was sent to you via WTOR (Write To Operator with Response) from a user.

```
OPR>RESPOND message_number response RET
```

Send allows you to send single- or multiple-line messages to destinations as defined by the available keywords.

```
OPR>SEND keyword message RET
```

- **All**
- **Batch-stream** *number*
- **Job** *number*
- **Operator**
- **Terminal** *number*

Set allows you to set various system parameters for the system devices as defined by the available keywords. The parameters are the arguments to the keywords. Fuller descriptions are available via **help set**.

```
OPR>SET keyword argument { switches } RET
```

- **Batch-stream** *stream_number*
- **Batch-stream** *range*
- **Disk-drive** **Channel** *n* **Drive** *n*
- **Job** *job_number*
- **Online**
- **Printer** *device_number*
- **Printer** *range*

- Scheduler
- Structure *structure_name*:
- Tape-drive *tape_drive_name*:
- Terminal

Show allows you to display various types of information as defined by the available keywords.

OPR>SHOW *keyword argument* { *switch* } RET

- Control-file Batch-stream *n* { /Lines:*m* }
- Messages
- Operators
- Parameters { Batch | Printer }
- Queues { All | Batch | Mount | Printer | Retrieval }
- Route-table
- Scheduler
- Status { Batch | Disk | Printer | Tape-drive }
- Time

Shutdown allows you to terminate the scheduling of job requests for the devices you specify, as defined by the available keywords.

OPR>SHUTDOWN *keyword argument* RET

- Batch-stream *stream_number*
- Printer *unit_number*

Start allows you to start up automatic scheduling of job requests through the devices you specify, as defined by the available keywords.

OPR>START *keyword argument* RET

- Batch-stream *stream_number*
- Printer *unit_number*

Stop allows you to temporarily stop any device as defined by the available keywords. You must give the CONTINUE command to resume the processing of job requests through the device.

OPR>STOP *keyword argument* RET

- Batch-stream *stream_number*
- Printer *unit_number*

Suppress allows you to suppress the carriage control on the line printer. All lines are single-spaced. Blank lines and form feeds are ignored.

OPR>SUPPRESS PRINTER *unit_number* { /FILE | /JOB | /STOP } RET

Switch allows you to switch a mount request that is using a tape volume to a new volume on another tape drive. This command should be used only when a user program wishes to switch volumes without issuing another tape mount request.

OPR>SWITCH *request_id vol_id tape_drive* RET

Take allows you to execute a series of OPR commands from a specified command file.

```
OPR>TAKE filespec { /DISPLAY | /NODISPLAY } RET
```

Appendix C

CheckD Commands

The CHECKD program is used for a variety of bookkeeping chores related to the maintenance of TOPS-20 file systems. This annotated listing of the command set offers some indications of how the commands are actually used to maintain and change the configuration of the file storage system.

Check the bit table or the directories on a disk structure. The check bit table function can be performed only for a structure that is mounted for the exclusive use of CHECKD. Thus, the boot structure and the login structure cannot have their bit tables checked during timesharing.

To check the bit table of the boot structure, answer the “Run CheckD” question at system start-up with “yes”. Note that when CHECKD is run in this way, it must complete successfully in order for the system to come up in regular operation. If CHECKD fails for the boot structure, the system will prohibit any allocation (or deallocation) of disk space on the boot structure, and it will allow login only on the console terminal. CHECKD can be run with the system in stand-alone operation, in the hopes that the underlying cause of the failure can be discovered and repaired.

To check the login structure, bring the system up in stand-alone mode (as described in Section 5.2.1) and run CHECKD.

You can check any structure, including the boot structure, when the system is stand-alone. However, if you choose to check the boot structure while the system is stand-alone, please make sure that system background activities (batch, printer spooling, etc.) are quiescent before and during the check.

```
CheckD>Check bittable structure_name [RET]
```

This command directs CHECKD to scan the directories and files of the selected structure, building an internal version of the structure’s bit table. Correct the system bit table where files occupy pages that are marked as unused. Write the lost pages file containing the pages that are marked as used in the system bit table but which are not occupied by any file. (See the **Release** command.)

```
CheckD>Check directory structure_name [RET]
```

Scan the directories of the selected structure, rebuilding any which need to be corrected.

Configure permits the user to change the setting of particular CHECKD variables. These variables control different aspects of CHECKD's function. The value of these variables can be displayed with the **show configuration** command.

```
CheckD>Configure feature { on | off} RET
```

where “feature” is one of the following:

- **bat-block-delete** allows entries to be deleted as well as added in BAT-edit mode. Otherwise, entries may only be added.
- **ecc-error-marking** causes pages containing correctable ECC errors discovered during a verify operation to be added to the BAT block. Otherwise, a correctable ECC error permits the page to remain in use.
- **soft-error-marking** causes occurrence of a soft error during a verify operation to result in the page being marked in the BAT block. Otherwise, soft errors (those that do not recur on retrying the operation) leave no track.

Create a new file structure. This command enters a dialog with the user, prompting for the structure alias, the number of units in the structure, the channel, controller, and unit numbers for each of the units in the new structure, the amount of swap space to allocate, and the owner's name. After it has all the information, this command attempts to create the structure.

Only units that are not part of a currently mounted structure may be selected as part of the new structure. In a multi-unit structure, all units must be of compatible sizes.

If the structure is to be bootable, it must have swapping space allocated on it. Structures intended only for customer data storage (i.e., not system data storage) should be defined with zero swapping space. If the structure is defined to have swapping space, CHECKD will also ask if the structure is to be used for auto-boot. There should be exactly one auto-boot structure for each system.

The “owner's name” is recorded in the new structure's home blocks, but it otherwise serves no purpose; this field defaults to the user name of the person running CHECKD.

```
CheckD>Create new_structure_name RET
```

Define changes properties of a mounted structure. Such changes persist despite subsequent dismounts and remounts of the structure, but changes are not effective until the next time the structure is mounted. See the **Set** command to make an immediate change. The keyword **no** can precede other keywords to negate their sense.

```
CheckD>Define structure_name {no | } keyword RET
```

- **Automatic-startup *system_serial_number*** defines (or removes) the number of the system to be booted from this structure. When setting a system serial number, if the serial number is omitted from the command, CHECKD will use the serial number of the current system. The **no automatic** option does not require a serial number.
- **boot-parameters**, and one or both of the following:
 - **read-parameters**
If set, directs the TDBOOT program to use these parameters.

- **halt-on-dump-errors**
If both this flag and Read-Parameters are set, TDBOOT will not reboot the system unless the memory dump of a crashed system can be completed.
- **data-cache**
causes the system to use the XRH-1 write data cache for this structure.
- **do-not-care**
allows the system to mount this structure, even if an unknown system has a direct access path to it. Otherwise, the system will not mount a structure if it can be accessed by an unknown system.
- **large-directories**
enables (disables) the use of large directories on this structure. A large directory holds more files than a limited directory.
- **login-structure** *system serial number*
enables (disables) the use of the selected structure as a login structure by the specified system.
- **name** *newname*
changes the physical name of the selected structure to *newname*. The **no** option is not applicable in this command.
- **password-encryption**
enables (disables) the encryption of passwords on the selected structure. This command affects the way that new passwords are stored subsequently. Any passwords already set on directories are left as they were.
- **write-verification**
requires one or both of the following:
 - **data** causes read-after-write verification for the file portion of the selected structure.
 - **swapping** causes read-after-write verification for swapping operations on the selected structure.

Dismount the selected structure or unit

```
CheckD>Dismount {structure_name | unit_number_triple} RET
```

Edit switches to BAT-edit mode, the Bad disk Address Table editor, which has its own prompt and its own set of commands. For details, use the **help** command from within BAT-edit. Usually, **verify** is performed before entering BAT-edit.

```
CheckD>Edit structure_name RET
```

Exit stops the CHECKD program.

File-DDT invokes a binary-file editor, allowing the operator to make arbitrary changes to a file. This is a convenience, to allow quicker repairs to be made when the file system needs manual repair.

Help offers help with the syntax of the commands. Help with the semantics is beyond the scope of the command.

Mount a selected file structure or a disk unit, so that it is a part of the accessible file system.

```
CheckD>Mount {structure_name | unit_number_triple} RET
```

Push invokes a command interpreter (i.e., the EXEC). To get back into CHECKD, the EXEC command is `pop`.

Read reads structure (or unit) information from special regions of the disk. Data is read from the specified unit or from all units of the specified structure and translated to human-readable form.

`CheckD>Read {structure_name | unit_number_triple} keyword RET`

- **bat-block**
Read and print the BAT (Bad disk Address Table) information for the specified structure or unit. Any of **primary** (the default), **secondary**, or **both** can follow, to identify which of the BAT blocks to read.
- **home-block**
Read and print the Home block information for the specified structure or unit. Any of **primary** (the default), **secondary**, or **both** can follow, to identify which home block to read.
- **port-data-block**
Read and print the contents of the Port Data block, a region of a disk unit that records how CFS systems can access this unit.
- **zone-page**
Read and print the contents of the unit's Zone page. This information describes the geometry of the disk drive.

Rebuild the bit table of the selected structure. The bit table is reinitialized and then the disk is scanned to identify which pages are occupied. The resulting bit table is written to replace the former bit table of this structure.

`CheckD>Rebuild structure_name RET`

Reconstruct either the index table, using the MSTR JSYS, or the root directory, by copying the backup copy of this directory, and then reinitializing the bit table.

`CheckD>Reconstruct {index-table | root-directory} structure_name RET`

Release the pages specified in the lost-pages file. Lost pages are those marked as occupied in the structure's bit table but which are not actually used by any file. This command allows the system to regain the use of these pages. Following the command, the lost-pages file is deleted. The lost-pages file is created by the `check bittable` command. When CHECKD is run during system startup, the lost-pages file is put in the `BS:<OPERATOR>` directory.

`CheckD>Release lost_pages_file_name structure_name RET`

Scan reads a list of disk page numbers from the given file. Then it scans the files of the selected structure to identify which file(s) contain the listed pages.

Sample uses include determining which files claim any page previously identified as being multiply assigned, and finding which files contain pages marked (as bad) in the BAT blocks.

`CheckD>Scan file_name structure_name RET`

Set properties of a mounted structure. In contrast to the **Define** command, **Set** changes the properties immediately, but does not change them permanently.

CheckD>Set {*structure_name* | *unit_number_triple*} {**no** | } *keyword* RET

- **dismount-in-progress**
Set or clear the “dismount in progress” state. While a structure is marked “dismount in progress”, any request to increment the structure mount count will be denied.
- **domestic**
Set or clear domestic status of structure.
- **regulated**
Set (clear) regulation for this structure. A regulated structure cannot be accessed until the job accessing the structure has incremented the structure’s mount count.

Show information about the program or the disk structure(s).

- **show configuration**
Display the state of program variables (the ones that are changed with the **configure** command).
- **show permanent structure**
Display the state of permanent structure variables (the ones that are changed with the **define** command).
- **show status structure**
Display the status of a mounted structure. The display includes the status of the variables that are changed with the **set** command.
- **show status**
Display the status of all disk units in the system.
- **show status channel, controller, unit**
Display the status of the selected disk unit.

Verify readability of all pages of the selected structure, or of the swapping region only. Every page is read. If an error results from the read operation, the disk address of the affected page is reported, and recorded in the BAT blocks. After the scan, **CHECKD** will enter the BAT-edit command mode.

CheckD>Verify {**file-system** | **swapping**} *structure_name* RET

Appendix D

TOPS–20 Start Up Operations

This appendix documents the general order in which TOPS–20 initializes itself and system services. The second section describes system variables that control various facilities.

D.1 TOPS–20 Initialization

- Initialize system memory, establish memory maps, the CSTs, the other sections needed for the operating system.
- Locate and mount the system structure. TOPS–20 uses very nearly the same procedure for locating the system structure as TDBOOT uses to select the boot structure. (The difference is that TOPS–20 requires that the system structure have swapping space.) Thus, this structure is generally known as the “boot structure”.
- Call SLNINI to initialize system logical names according to the following table. Where the name *bs* appears in the following list, TOPS–20 replaces it with the actual name of the boot structure.

SYS	bs:<SUBSYS>
HLP	SYS:
SYSTEM	bs:<NEW-SYSTEM>, bs:<SYSTEM>
EDITOR	SYS:EDIT.EXE
ACCOUNT	bs:<ACCOUNTS>
DEFAULT-EXEC	SYSTEM:EXEC.EXE
PS	bs:
POBOX	PS:
BS	bs:
SPOOL	bs:<SPOOL>
SERR	bs:<SYSTEM-ERROR>

(Note: if, subsequently, a login structure is mounted, the logical name **PS** is redefined to be the name of the login structure.)

- Call GETNAM to process **SYSTEM:MONNAM.TXT**

- Call SERINI to initialize `SERR:ERROR.SYS`
- Call MNTINI to read the file `SYSTEM:INTERNET.ADDRESS` and initialize the multi-net device. MNTINI also reads `SYSTEM:HOSTS.TXT`, unless the debugging flag, `DEBUGSW`, contains the value 2 or more, in which case MNTINI reads `SYSTEM:HOSTS.DEBUG` instead.
- Call INTINI to read `SYSTEM:INTERNET.GATEWAYS` and initialize processing of Internet Protocol (IP).
- TOPS-20 then runs `SYSTEM:n-SETSPD.EXE`, to process configuration commands in the file `SYSTEM:n-CONFIG.CMD`. (The value *n* is based on the TOPS-20 monitor's version number.)
- Call GETNAM to process `SYSTEM:MONNAM.TXT` again, in case some logical names have changed since the first time GETNAM was run.
- TOPS-20 then runs `SYSTEM:TIME.EXE` which attempts to read the current time from the network. If the TIME program cannot find the current time from a network server, the system will prompt to request manual entry of the date and time.
- Call ENACT to enable the verification of account strings. ENACT reads or creates the file of valid account strings, `BS:<SYSTEM>ACCOUNTS-TABLE.BIN`.
- Unless the debugging flag is set (i.e., when `DEBUGSW` contains a value less than 2), ask the question "Why Reload?". Record the response in the error log.
- Unless the debugging flag is set (i.e., when `DEBUGSW` contains a value less than 2), ask the question "Run CheckD?". If the response is affirmative, run `SYSTEM:CHECKD.EXE` and check the bit table of the boot structure.
- If SETSPD enables a login structure, and if `DEBUGSW` has a value less 2, call FNDLGS to locate and mount the login structure. If a login structure is mounted successfully, redefine the logical name `PS:` to be the name of the login structure.
- Start the CHKR fork. (The initial fork continues as the DDMP fork.) The CHKR fork will then:
 - Open `BS:<SYSTEM>ERROR.REPORT` for CHKR fork output.
 - Call USGINI to make the system's binary accounting detail file ready for use: if the file `ACCOUNT:SYSTEM-DATA.BIN` exists, it will be positioned so that the system can append new records to it; otherwise, it will be created. USGINI adds the system reload record to the accounting details. USGINI also will read or create `ACCOUNT:CHECKPOINT.BIN.1`.
 - Run `SYSTEM:n-SETSPD.EXE`, again. If `DEBUGSW` contains a value that is less than 2, SETSPD will process `BS:<SYSTEM>DUMP.EXE`. The protection of `DUMP.EXE` will be set to 770000 and it will be copied to `DMP:DUMP.CPY;P770000`.¹ If the logical name `DMP:` is `NUL:`, no copy will be made. If `DMP:` is undefined, the copy is made to `BS:<SYSTEM>DUMP.CPY;P770000`.
 - Run `SYSTEM:n-SYSJOB.EXE`. SYSJOB will process the file `SYSTEM:n-SYSJOB.RUN` unless `DEBUGSW` contains a value greater than or equal to 2, in which case SYSJOB processes `SYSTEM:n-SYSJOB.DEBUG`. In the usual case, SYSJOB will start various system services and enable other terminals to log in.

¹ SETSPD may invent a different name for the dump copy file, based on the name of the latest BUGHLT and the edit number of the monitor.

- Proceed with normal CHKR duties.
- Fork 0 becomes the DDMP fork.

D.2 TOPS-20 System Variables

The following system variables affect the overall behavior of TOPS-20. These can be changed by patching² `BS:<SYSTEM>MONITR.EXE` and rebooting, or they can be changed at boot time by entering `KDDT` before the system starts (see Section 5.2.1).

<code>DEBUGSW</code>	<code>=0</code>	Normal
	<code>≠ 0</code>	Keep <code>KDDT</code> locked down (i.e., present in memory). Breakpoint on <code>BUGHLT</code> . Don't release fork lock after timeout. Do not take swap space low action.
	<code>> 1</code>	Enable ARP debugging. Allow homeblock rewrite when not refreshing filesystem via the <code>start:+1</code> dialogue. Don't mount the login structure. Don't ask about running <code>CHECKD</code> . Read <code>HOSTS.DEBUG</code> instead of <code>HOSTS.TXT</code> . Read <code>n-SYSJOB.DEBUG</code> instead of <code>n-SYSJOB.RUN</code> . Write enable the monitor (permit inserting breakpoints). <code>CHECKD</code> can work on <code>BS:</code> or <code>PS:</code> bit table after startup.
	<code>> 2</code>	Use monitor creation date for current system time.
<code>DCHKSW</code>	<code>≠ 0</code>	Break on <code>BUGCHK</code> if <code>KDDT</code> is present.
<code>DINFSW</code>	<code>≠ 0</code>	Break on <code>BUGINF</code> if <code>KDDT</code> is present.
<code>DINISW</code>	<code>≠ 0</code>	Break just before and just after calling <code>PGRINI</code> .
<code>EDDTF</code>	<code>≠ 0</code>	Keep <code>KDDT</code> locked down (present in memory).
<code>SMLKF</code>	<code>≠ 0</code>	Keep the swappable monitor locked in memory.
<code>WEFLAG</code>	<code>≠ 0</code>	Write enable the monitor.
<code>DEBUGIP</code>	<code>=0</code>	Turn on networks if <code>DEBUGSW < 2</code> .
	<code>=1</code>	Turn on networks if <code>DEBUGSW > 1</code> .
	<code>=2</code>	Turn on networks always (default).

²Use `FILDDT` to patch the monitor. Do **not** use `Get` and `Save`: see Section 5.2, page 40.

Appendix E

Generating a Tailored Batch and Spooling System

This appendix describes how to generate a TOPS-20 batch and spooling (GALAXY) system specifically tailored for your installation. To do this, you must have already installed a standard Release 6 GALAXY system. The standard GALAXY system is satisfactory for most sites. However, you can use the steps in this appendix if you need to generate a GALAXY system having different parameters from the standard one.

Note

While the steps in generating a tailored Release 6 GALAXY system can be used with timesharing users logged in, the critical phase of stopping the standard Release 6 system and starting up your tailored one (Section E.4 and Section E.5) should, for safety reasons, be performed on a stand-alone system. The queues need not be empty, however, as long as a Release 6 (as opposed to Release 5 or earlier) GALAXY system is already running.

The following paragraphs summarize the procedure for generating and starting up a new GALAXY system:

1. If necessary, run the DUMPER program to restore the GALAXY source files to disk. In addition to the source code, the files include a special batch control file called **GALAXY.CTL**.
2. Run the GALGEN program to create a file containing the new GALAXY system parameters. The name of the parameter file is **GALCNF.MAC**.
3. Submit the control file **GALAXY.CTL** to your currently running batch system. The batch job that is created by **GALAXY.CTL** reads the parameter file created by GALGEN; it generates a new set of GALAXY files of type **.EXE** in a build area. The batch job creates a log file which should be saved for reference.
4. Stop all components of the currently running GALAXY system (QUASAR, LPTSPL, etc.)

5. Copy all the old GALAXY files from their current area (usually **SYSTEM:**) to some other area.
6. Copy all the new GALAXY files from the build area to **SYSTEM:**.
7. Start up the components of the new GALAXY system.
8. Delete the **.EXE** files of the old GALAXY system when the new batch system is operating correctly. If you want to be cautious, you can save the old **.EXE** files on tape or in a backup directory.

You may delete the GALAXY source files. However, if your installation can spare the disk space, it is more convenient to leave the sources on disk. That way, if any code patches must be made to the batch system, you have the sources readily available. Otherwise, you have to restore the sources from the software tape if patching is required.

Note

The procedures described below assume that you have a Release 6 GALAXY system currently running. These procedures can be carried out during normal timesharing.

The steps in this appendix assume that you are using the console terminal (**CTY**).

You must be a privileged user. Log in and enable Wheel or Operator privileges to perform these procedures.

E.1 Restoring the GALAXY Source Files

The GALAXY source files are normally provided to you in the directory **<SRC.7.GALAXY>**; related information is supplied in the directoried **<SRC.7.GALAXY.BUILD>** and **<SRC.7.GALAXY.RUNTIME>**. If the files are there, you need not restore them from tape; skip to Section E.1.2. Otherwise, you will have to restore the files from the “Ship” tape that was delivered with your system.

To restore the GALAXY source files, you must build the necessary directories, mount the Ship tape, and run the **DUMPER** program. Perform the operations described in this section to restore the GALAXY files.

E.1.1 Restoring the Galaxy Sources

Note

The examples in this section use **MTA0:** as the tape drive. However, you can assign and use any tape drive that is available to you: **MTA0:** is not required. Be sure to substitute the name of the tape drive that you assigned wherever “**MTA0:**” appears in this appendix.

Obtain a tape drive by following the procedure described in Section 6.1.

Create directories to hold the GALAXY source files. Although you can give the directories any name, it will be easiest to use the names `<SRC.7.GALAXY>`, `<SRC.7.GALAXY.BUILD>`, and `<SRC.7.GALAXY.RUNTIME>` because these are the names of the directories on tape. If the directories `<SRC>` and `<SRC.7>` do not already exist, they must be built also.

Decide which structure you want to build the directories on, and connect to any directory (e.g., `<OPERATOR>`) on that structure. (By being connected to the target structure, you do not have to give the structure name explicitly in the `Build` commands that follow.)

If it does not already exist, build the directory `<SRC>`:

```
!Build <SRC> 
```

The system displays either “[New]” if the directory does not already exist, or “[Old]” if the directory exists. The subcommand prompt, “!!” follows.

If the system displays “[Old]”, type `<CTRL/C>` to stop the build process: the directory already exists. Otherwise, create it. It is a files-only directory, but it does not require any other parameters. Type the subcommand “Files” and an empty subcommand line to finish the `Build` command:

```
!Build <SRC> 
[New]
!!files 
!!
!
```

Build the directory `<SRC.7>`, if it does not already exist:

```
!Build <SRC.7> 
```

The system displays either “[New]” if the directory does not already exist, or “[Old]” if the directory exists. The subcommand prompt, “!!” follows.

If the system displays “[Old]”, type `<CTRL/C>` to stop the build process: the directory already exists. Otherwise, create it. Type the subcommand “Preserve” and an empty subcommand line to finish the `Build` command:

```
!Build <SRC.7> 
[New]
!!preserve 
!!
!
```

The “**Preserve**” subcommand allows the subdirectory to be created without regard for the superior directory’s disk quota or subdirectory quota.

If it does not already exist, build the directory <SRC.7.GALAXY>:

```
!Build <SRC.7.GALAXY> RET
```

The system displays either “[New]” if the directory does not already exist, or “[Old]” if the directory exists. The subcommand prompt, “!!” follows.

If the system displays “[Old]”, type <CTRL/C> to stop the build process: the directory already exists. Otherwise, create it. Type the indicated subcommands to give an allocation of disk space to the new directory. Type RET on an empty subcommand line to finish the **Build** command:

```
!Build <SRC.7.GALAXY> RET
[New]
!!preserve RET
!!Working 5000 RET
!!Permanent 5000 RET
!!RET
!
```

Follow the example to assign storage space and to end the **Build** procedure.

Build the directory <SRC.7.GALAXY.BUILD>, if it does not already exist:

```
!Build <SRC.7.GALAXY.BUILD> RET
```

The system displays either “[New]” if the directory does not already exist, or “[Old]” if the directory exists. The subcommand prompt, “!!” follows.

If the system displays “[Old]”, type <CTRL/C> to stop the build process: the directory already exists. Otherwise, create it. Type the indicated subcommands to give an allocation of disk space to the new directory. Type RET on an empty subcommand line to finish the **Build** command:

```
!Build <SRC.7.GALAXY.BUILD> RET
[New]
!!preserve RET
!!Working 2000 RET
!!Permanent 2000 RET
!!RET
!
```

If it does not already exist, build the directory <SRC.7.GALAXY.RUNTIME>:

```
!Build <SRC.7.GALAXY.RUNTIME> 
```

The system displays either “[New]” if the directory does not already exist, or “[Old]” if the directory exists. The subcommand prompt, “!!” follows.

If the system displays “[Old]”, type <CTRL/C> to stop the build process: the directory already exists. Otherwise, create it. Type the indicated subcommands to give an allocation of disk space to the new directory. Type on an empty subcommand line to finish the **Build** command:

```
!Build <SRC.7.GALAXY.RUNTIME> 
[New]
!!preserve 
!!Working 1000 
!!Permanent 1000 
!!
!
```

To restore the GALAXY files from tape, mount the Ship tape on the tape drive you assigned. Be sure the drive is on line. In the command examples that follow, we assume that you are using **MTA0:**. If that is not your situation, substitute the device name of the tape that you are using wherever “**MTA0:**” appears below.

Skip the first five files on the Ship tape. Perform the following commands:

```
!Rewind MTA0: 
!Set Tape Format High 
!Copy MTA0:,MTA0:,MTA0:,MTA0:,MTA0: NUL:  Copy five files to NUL:
!
```

You must run the **DUMPER** program to restore the GALAXY source files from tape. To run this program, type **Dumper**. The **DUMPER** program prints the **Dumper>** prompt:

```
!Dumper 
DUMPER>
```

Tell the **DUMPER** program which tape drive you are using; tell it to use the system-default account:

```
DUMPER>Tape MTA0: 
DUMPER>Account system-default 
DUMPER>
```

To restore the GALAXY sources, type the **Restore** command shown below. If you are restoring to a structure other than **PS:**, use the name of that structure where “**PS:**” appears in the **Restore** command that follows:

```
DUMPER>Restore DSK*:<SRC.7.GALAXY.*>*.*.* PS:<*>*.*.* 
Saveset "SHIP: BOOT TAPE" 18-Mar-96 1400
Loading files into PS:<SRC.7.GALAXY>
Loading files into PS:<SRC.7.GALAXY.BUILD>
Loading files into PS:<SRC.7.GALAXY.RUNTIME>
End of Tape.
Total files restored:      n
Total pages restored:     n
DUMPER>
```

The DUMPER prints a message identifying the saveset. Then you may have to wait 30 to 60 minutes, while DUMPER skips various files on the tape before coming to the GALAXY sources. When DUMPER comes to the end of the saveset, it prints the number of files and pages restored; then it prompts again.

Error: If DUMPER does not print a message identifying the saveset as “SHIP: BOOT TAPE”, the tape is not positioned at the correct saveset. If not already at the **DUMPER>** prompt, type **<CTRL/E>** and give the **Abort** command to stop DUMPER. Give the **Rewind** command to DUMPER; exit from DUMPER. Again, reposition the tape by using the **Copy** command.

If the message “**Loading files into PS:<7.SRC.GALAXY>**” does not appear, no files were restored from tape. (While DUMPER is running you may type **<CTRL/A>** to make it report on its progress.) Be sure that the correct tape is mounted on the tape drive you have assigned to your job. Then exit from DUMPER. Begin again at the start of this subsection. If errors persist, the tape may be bad.

Once the files have been restored from tape, stop the DUMPER program. To do so, follow the example to rewind the tape and exit from DUMPER. The EXEC prints its enabled prompt, “!”:

```
DUMPER>Rewind 
DUMPER>Exit 
!
```

To unload the tape from the tape drive, type the **EXEC** command:

```
!Unload MTA0: RET
```

The tape drive unloads the tape onto the source reel.

You can now remove the Ship tape from the tape drive.

After you unload the tape from the drive, return the tape drive to the pool of available resources by following the procedure described in Section 6.5, page 71.

E.1.2 Verify

To verify that you have restored all the source files for GALAXY type the command:

```
!Information disk-usage <SRC.7.GALAXY> RET
```

The system prints information about the amount of disk space used by the directory:

```
!Information Disk-Usage <SRC.7.GALAXY*> RET
PS:<SRC.7.GALAXY>
1862 Pages assigned
5000 Working pages, 5000 Permanent pages allowed

SHIP:<SRC.7.GALAXY.BUILD>
838 Pages assigned
2000 Working pages, 2000 Permanent pages allowed

SHIP:<SRC.7.GALAXY.RUNTIME>
107 Pages assigned
1000 Working pages, 1000 Permanent pages allowed

Total of 2807 Pages assigned, in 3 directories.
783651 Pages free on SHIP:, 157068 pages used.

!
```

Note

If you restored the GALAXY saveset to another directory, use that directory name in the “Information Disk-Usage” command.

If the number of pages assigned does not approximately match the number shown above, you may not have restored all the files necessary for creating a GALAXY system. Check to be sure you used the correct tape; then begin again at Section E.1.1.

E.2 Running the GALGEN Program

Run the GALGEN program to create the GALAXY configuration file, `PS:<BUILD>GALCNF.MAC`, which will contain the parameters describing the tailored GALAXY system you are building.

Be sure you run GALGEN Version 6, which is the latest version.

Connect to the build area:

```
!Connect <Src.7.Galaxy.Build> [RET]
!
```

Type `Galgen` and press `[RET]`. The GALGEN program responds by giving its name and version number; it then prints several paragraphs of information:

```
!Galgen [RET]
```

```
Galgen Version 6(7626)
```

```
[Starting Galaxy Generation Procedure for TOPS-20 System]
```

```
[Writing Galaxy Configuration File PS:<SRC.7.GALAXY.BUILD>GALCNF.MAC.1]
```

```
In the following dialog, all questions are of the form:
```

```
text of question (list or range of answers) [default answer]
```

The line starts with the actual text of the question. Following the question is a description of the possible answers enclosed in parentheses. This description might be a range of numbers, a list of keywords, or a textual description. Following this description is the default answer, enclosed in square brackets. The default will be assumed if you only press the RETURN key.

You have the choice of carrying on a LONG dialog in which an explanation of each question is provided automatically, or a SHORT dialog in which it is assumed that you are familiar with the Galaxy generation procedure. In either case, you can always obtain the help text by typing a question mark (?) in response to any question.

Answer the following question with either LONG (for a long dialog) or SHORT (for a short one). Simply pressing the RETURN key implies SHORT.

```
Dialog Length (SHORT, LONG) [SHORT]
```


You have the option of a long dialogue, in which GALGEN explains the choices it offers you, or a short one, in which only the choices themselves are displayed. If you are an experienced user of GALGEN, you may want the short dialogue, which is faster; in this case, press the **RET** key and skip ahead.

To request the long dialogue with GALGEN, type **LONG** and press the **RET** key. GALGEN responds with some information; then it asks the first question about GALGEN parameters:

```
Dialog Length (SHORT, LONG) [SHORT] LONG RET
-----
Administrative Controls and Parameters
-----

The questions in this section determine absolute Galaxy
parameters, which cannot be overridden by user or operator
commands.

The system logs all operator interactions in a disk file in the
spooling area. Your answer to the following question specifies
the name to be used for this file.

Operator log filename(3-20 Characters) [OPERATOR-SYSTEM]
```

Answer the questions in the GALGEN dialogue.

GALGEN presents a series of questions, one at a time, for you to answer. If you asked for a long dialogue, an explanation of each question is included. Otherwise, you will see only the questions, the range of choices, and the default answers. Default answers are shown in brackets ([]). This value (the default) is used if you answer the question by pressing the **RET** key.

Press the **RET** key when you want the default answer shown in brackets and are ready to see the next question in the GALGEN dialogue. Otherwise, type your answer and then press the **RET** key to see the next question.

In the sample dialogue below, the default answer is taken for most questions except the last question in the dialogue.

Note

You must supply the correct values for your own system.

The following is what you would see if you requested the “long” dialogue:

Operator log filename(3-20 Characters) [OPERATOR-SYSTEM]

Users can specify a priority for their batch and spooling requests with the /PRIORITY switch. The allowable values on this parameter range between 1 (representing lowest priority) and 63 (representing the highest). The following two questions relate to this switch. The first question requests the maximum priority which may be specified by a non-privileged user.

Maximum priority non-privileged users may specify(1-63) [20]

You may specify the value to be used if the user does not specify a /PRIORITY switch. This value must be between 1 and the maximum that you specified in the previous question.

Default for /PRIORITY on batch and spooling requests(1-63) [10]

This value is to control the number of objects (printers, batch streams, etc.) that can be controlled by a single command in OPR. This value should be at least the maximum number of objects of a single type you may have. The value must be at least 1.

Maximum number of objects that can be controlled with one command (1-100) [6]

Specify the highest number that a batch stream can be on the system. The value must be greater than zero and less than or equal to 10000.

Highest number a batch stream can be on the system (1-10000) [100]

The priority of a process and the permissible range of queues it may run in is determined by the scheduler priority control word.

A system process gets a higher priority than all user jobs and gets as much time as is necessary for execution.

Do you wish QUASAR to run as a system process?(YES,NO) [YES]

Do you wish ORION to run as a system process?(YES,NO) [YES]

Do you wish NEBULA to run as a system process?(YES,NO) [YES] No

Do you wish BATCON to run as a system process?(YES,NO) [YES]

Do you wish LPTSPL to run as a system process?(YES,NO) [YES]

The following parameter determines whether CLUSTER Galaxy should be ENABLEd or DISABLEd at Galaxy startup. The default is YES.

Do you want to ENABLE CLUSTER Galaxy at Galaxy start up?(YES,NO) [YES]

The following parameters allow a user with SEMI-OPR privilege to be able to control certain devices.

Do you want SEMI-OPR to be able to control the printers?(YES,NO) [YES]

Do you want SEMI-OPR to be able to control the plotters?(YES,NO) [YES]

Do you want SEMI-OPR to be able to control the card readers?(YES,NO) [YES]

Do you want SEMI-OPR to be able to control the card punch?(YES,NO) [YES]

Do you want SEMI-OPR to be able to control the paper tape punch?(YES,NO) [YES]

Do you want SEMI-OPR to be able to control the tape drives?(YES,NO) [YES]

Applications

The following section establishes parameters for applications supported by GALAXY (for example, NCP). These applications will be accessible through OPR, the operator interface.

You can enable applications through your answers to the following questions.

Will you be running applications?(YES,NO) [YES] NO

Each application must be associated with an application table name (for example, NCPTAB).

Application name or carriage return to exit(1-20 characters) NCP

Table name for NCP(1-6 characters) NCPTAB

Is there an internal ORION application for this application?(YES,NO) [NO]

Application name or carriage return to exit(1-20 characters) LCP

Table name for NCP(1-6 characters) LCPTAB

Is there an internal ORION application for this application?(YES,NO) [NO] yes

Application module name(1-6 characters) LCPORN

Application name or carriage return to exit(1-20 characters)

Note

If you are running applications supported by GALAXY, see the appropriate documentation to determine the application table names.

Batch Job Defaults

The questions in this section are used to define default values for the various options that a user can specify when submitting a batch job.

The batch user can specify a maximum runtime for his batch job using the /TIME switch. If he does not specify this switch, the following default will be used:

Default batch job runtime limit(5-9999 seconds) [300]

Every batch job has maximum limits for each type of spooled output. These limits can be set by the batch user with appropriate switches to the SUBMIT command. If the user doesn't specify these switches, the following defaults will be used:

Line printer output limit(0-9999 pages) [200]

Card punch output limit(0-9999 cards) [1000]

Paper tape punch output limit(0-5000 feet) [200]

Plotter output limit(0-5000 minutes) [30]

The user can specify whether or not the batch log file should be printed after the job terminates by using the /OUTPUT switch to the SUBMIT command. The action is specified by: LOG to always print the log file, NOLOG to never print the log file, and ERROR to print the log file only if the job ended with an unhandled error. If the user doesn't specify this switch, the following value will be used.

Default for batch SUBMIT/OUTPUT (LOG,NOLOG,ERROR) [LOG]

Line Printer Defaults and Parameters

The questions in this section are used to set default values for the line printer spooler.

Job banner pages printed at the beginning of each print request. The user's name and any /NOTE value is printed on them.

Number of job banner pages (0-5) [2]

Job trailer pages are printed at the end of each print request. The user's name and any spooler checkpoint information is printed on them.

Number of job trailer pages(0-5) [2]

File header pages are printed before each file or copy in each print request. The file name is printed in BOLD letters on each header page.

Number of file header pages(0-5) [2]

The user can specify a special form for a print request with the /FORMS switch. If this switch is not specified, the following default will be used.

Name for standard output forms(4-6 Characters) [NORMAL]

The number of characters that uniquely identify a form is compared with the current form name to determine if a forms change is required for a specific user request.

Number of characters which uniquely identifies form(2-6) [4]

Cluster LPTSPL requires a directory that must be shared between all the systems in the cluster that send print requests to a Cluster LPTSPL. The default logical name is LPTXFR.

Cluster LPTSPL shared directory logical name (2-40 characters)
[LPTXFR]

Cluster printers require the LISSPL server to be running on the node where the actual physical printer is located. LISSPL has server forks that pick up the print request from remote nodes. The default number of server forks is 2.¹

Number of LISSPL server forks (1-4) [2] 0

Tape Label Action

The system administrator can allow non-privileged users to bypass label processing.

Allow bypassing for non-privileged users?(YES,NO) [NO]

NEBULA (Cluster GALAXY message router) options

The NEBULA ACK/ACK RESPONSE message option ensures that NEBULA actually has DECnet communication with nodes that believes it has communication with by sending an ACK message every 5 minutes to each node. If an ACK RESPONSE message is not received from a node after 5 minutes, then NEBULA assumes that it no longer has communication with that node. This covers the case where the NI of a remote node is reloaded. In this case, the local NEBULA is not interrupted so it assumes that it still has communication

¹The TOAD-1 System does not support LISSPL.

with the remote node when in fact it does not.

Do you want NEBULA ACK/RESPONSE ACK messages enabled?(YES,NO) [NO]

Miscellaneous Defaults and Parameters

The questions in this section refer to general GALAXY parameters.

The following section establishes default bias and disk page to spooler unit conversion factors.

Do you want the standard limit computation(YES,NO) [YES]

The default action to be taken when output exceeds the specified limit for an output spooler may be to ABORT the job and ignore all remaining output, ASK the operator what to do, or PROCEED to finish processing the job.

Default output-limit-exceeded action(ABORT,ASK,PROCEED) [ASK] Abort

[End of Galgen Dialog]

!

E.3 Assembling the New GALAXY System

In the last step, you completed the GALGEN program, which produced the GALAXY parameter file, **GALCNF.MAC**. This file contains your choice of options for a tailored batch and spooling system. To create this batch and spooling system, you must submit the GALAXY control file, **GALAXY.CTL**, to your present batch system. **GALAXY.CTL** reads **GALCNF.MAC** and generates the files and programs for a tailored GALAXY system.

While still connected to the directory <SRC.7.GALAXY.BUILD>, perform the command:

```
!Take Batch.Cmd 
```

This command establishes a set of logical names through which the sources for the GALAXY system can be found. Then, to start the assembly process, submit the **GALAXY.CTL** file to the batch queue. After you submit the batch job, the system prints a message giving the job name, the request I.D., and the time limit of the batch job:

```
!Submit GALAXY.CTL/Time:0:30:00/Restartable 
[Batch job GALAXY queued, request #170, limit 0:30:00]
!
SJ 0:
SJ 0: 15:20:48      Batch-Stream 0      -- Started --
```

```

SJ 0:
SJ 0: 15:20:48      Batch-Stream 1    -- Started --
SJ 0:
SJ 0: 15:20:48      Batch-Stream 2    -- Started --
SJ 0: OPR>
SJ 0: 15:20:49      Batch-Stream 0    -- Begin --
SJ 0:                Job GALAXY Req #170 for OPERATOR
!
```

Error: If the file `GALAXY.CTL` cannot be found, it may be because you neglected to `!Take Batch.Cmd` RET.
If so, do the `Take` now; resubmit the batch request.

While the `GALAXY` system assembly proceeds (this may take an hour), the system prints messages for each successful assembly it completes. The messages are similar to the following example:

```

SJ 0: OPR>
SJ 0: 15:39:07      Batch-Stream 2 JOB #7 -- Message from Batch User --
SJ 0:                Job PLEASE Req #6 for OPERATOR
SJ 0:                PLEASE PLEASE Assembly Successful
SJ 0: OPR>
SJ 0: 15:39:15      Batch-Stream 2    -- End --
SJ 0:                Job PLEASE Req #6 for OPERATOR
SJ 0: OPR>
SJ 0: 15:39:16      Batch-Stream 2    -- Begin --
SJ 0:                Job LPTSPL Req #8 for OPERATOR
```

Note

If an assembly is unsuccessful, examine the appropriate `.LOG` file to see if you can find the problem. If you can, correct it (which might involve editing the control file), delete the `.LOG` file, and resubmit the specified `.CTL` file. If you are unable to determine what caused the failure, consult your systems programmer or software specialist.

Once the assembly completes, you can run the new batch system. However, before you proceed to shut down the old `GALAXY` system, you should check the `.LOG` files to be sure that no problems arose during batch system generation. It is conceivable that a %-type (warning) error occurred. Such an error might not cause processing of `GALAXY.CTL` to be unsuccessful, but it could create problems later, when you attempt to use the new batch system. Therefore, it is important that you correct any problems now, while you are still running the old batch system.

E.4 Shutting Down the Old GALAXY System

Once you have checked the .LOG files to be sure that no errors occurred during the build procedure, you are ready to stop all old GALAXY programs and start the new ones. Perform the following steps to do so.

Note

During the steps in this section, it is important **NOT** to have timesharing users logged in to the system, as their jobs could initiate batch or spooling requests during the critical period of changeover. However, it is acceptable for requests to be present in the queues at the time you shutdown each GALAXY object, as long as the old GALAXY system is a Release 6 system.

Return to the OPR program so you can stop the old GALAXY objects:

```
!OPR RET
OPR>
```

To determine which GALAXY objects are in use, give the “Show Status” command. The OPR program responds with a list of GALAXY objects:

```
OPR>Show Status RET
OPR>
16:32:03      -- System Device Status --
Batch-Stream Status:
  Strm      Status
  ----      -
    0  Idle
    1  Idle
    2  Idle
Printer Status:
  Unit      Status
  ----      -
    0  Idle
OPR>
```

For each GALAXY object shown by the “Show Status” command, enter a “Shutdown” command:


```

OPR>Shutdown Printer 0 
OPR>
16:32:26      Printer 0      -- Shutdown --
SJ  0: OPR>
SJ  0: 16:32:26      Printer 0      -- Shutdown --
OPR>Shutdown Batch-Stream 0:2 
OPR>
16:32:37      Batch-Stream 0      -- Shutdown --
16:32:37      Batch-Stream 1      -- Shutdown --
16:32:37      Batch-Stream 2      -- Shutdown --
SJ  0: OPR>
SJ  0: 16:32:37      Batch-Stream 0      -- Shutdown --
SJ  0:
SJ  0: 16:32:37      Batch-Stream 1      -- Shutdown --
SJ  0:
SJ  0: 16:32:37      Batch-Stream 2      -- Shutdown --
OPR>

```

Error: If you see a message similar to “There are no devices started”, you can ignore this message and go on to the next Shutdown command. The message means only that your site was not using the specified device.

To verify that all parts of the GALAXY system are shut down, repeat the “Show Status” command:

```

OPR>Show Status 
OPR>
16:32:47      -- There are no devices started --
OPR>

```

To return to TOPS-20 command level so you can stop the GALAXY components, type the Exit command; the EXEC prints its enabled prompt “!”:

```

OPR>Exit 
!

```

To stop the GALAXY components, use the command:

```

!       the first character typed is <CTRL/E>

```

The EXEC responds “[Please type SYSJOB commands - end with ^Z]”, and waits for your commands.

```
!^ Espeak 
[Please type SYSJOB commands - end with ^Z]
```

You should stop all the components referenced in the `SYSTEM:SYSJOB.RUN` file. You should also stop `BATCON`, which normally runs as a separate job under `SYSJOB`. Follow the example to stop the `GALAXY` components.

```
[Please type SYSJOB commands - end with ^Z]
KILL QUASAR 
Kill ORION 
KILL MOUNTR 
KILL BATCON 
KILL LPTSPL 
KILL TCPSPL 
CCJOB 1 
STATUS 
```

To return to TOPS-20 command level, type `<CTRL/Z>`. The system prints the EXEC's enabled prompt, "!", followed by an echo of your commands to `SYSJOB`:

```
^Z
!KILL QUASAR
KILL ORION
KILL MOUNTR
KILL BATCON
JOB 1 # RESET *           Kill LPTSPL and TCPSPL
#
CCJOB 1
STATUS
SYSJOB status at 31-May-87 1639
  0 INFO  DISMS at 1013  31-May-87 1639
  1 MAPPER DISMS at 266  31-May-87 1639
  5 MAILER DISMS at 1225 31-May-87 1639
  7 LPTSPL DISMS at 415322 31-May-87 1639
0: Job 1, TTY 160, PTYCON
1: Job 2, TTY 161, EXEC
 Type  to see the next prompt.
!
```

You should create a directory to store your old `GALAXY` system before copying your new one into `SYSTEM:.` To do so, use the `Build` command:

```
!Build PS:<OLD-GALAXY> 
[New]
!! 
!
```

This creates a directory with default characteristics, after which you return to TOPS-20 command level.

Now copy the old GALAXY components to the newly-built directory:

```
!COPY SYSTEM:QUASAR.EXE PS:<OLD-GALAXY>* 
PS:<SYSTEM>QUASAR.EXE.1 => PS:<OLD-GALAXY>QUASAR.EXE.1 [OK]
!
```

Repeat the Copy command as many times as necessary to copy all the files of our present GALAXY system into PS:<OLD-GALAXY>. You must copy all of the following files, if they exist:

```
BATCON.EXE
LPTSPL.EXE
MOUNTR.EXE
NEBULA.EXE
ORION.EXE
QUASAR.EXE
TCPSPL.EXE
```

The following files reside on SYS:, usually in PS:<SUBSYS>. They must be moved as well:

```
GLXLIB.EXE
GLXLIB.REL
OPR.EXE
PLEASE.EXE
```

E.5 Starting up the New GALAXY System

Now you are ready to replace the old GALAXY system on SYS: with your new tailored system.

Note

If you restored the GALAXY source files to a directory other than PS:<SRC.7.GALAXY.BUILD>, use that directory name in the following steps.

Copy the QUASAR.EXE component of the new GALAXY system into the system area:

```
!COPY PS:<SRC.7.GALAXY.BUILD>QUASAR.EXE PS:<SUBSYS>QUASAR.EXE [RET]
PS:<BUILD>QUASAR.EXE.1 => PS:<SUBSYS>QUASAR.EXE.1 [OK]
!
```

Repeat the Copy command as many times as it is necessary to copy all the files of your new GALAXY system into PS:<SUBSYS>. You must copy all the files listed at the end of Section E.4.

To start the new GALAXY system you must enter the ^ESpeak command. The system prints “[Please type SYSJOB commands - end with ^Z]”.

```
!^ESpeak [RET]
[Please type SYSJOB commands - end with ^Z]
```

To start all of the components of the GALAXY system, you must type a Run command for each component referenced in the file SYSJOB.RUN. Follow the example to start the GALAXY components. End the list by typing <CTRL/Z>:

```
[Please type SYSJOB commands - end with ^Z]
RUN SYSTEM:QUASAR [RET]
RUN SYSTEM:ORION [RET]
RUN SYSTEM:MOUNTR [RET]
RUN SYSTEM:BATCON [RET]
JOB 1 / [RET]
RUN SYSTEM:LPTSPL [RET]
RUN SYSTEM:TCPSPL [RET]
/ [RET]
STATUS [RET]
^Z
```

```

!RUN SYSTEM:QUASAR
RUN SYSTEM:ORION
RUN SYSTEM:MOUNTR
RUN SYSTEM:BATCON
JOB 1 /
RUN SYSTEM:LPTSPL
RUN SYSTEM:TCPSPL
/

STATUS
SYSJOB status at 31-May-87 1653
  0 INFO   DISMS at 415322 31-May-87 1653
  1 MAPPER DISMS at 415322 31-May-87 1653
  5 MAILER DISMS at 415322 31-May-87 1653
  7 LPTSPL DISMS at 415322 31-May-87 1653
!

```

You must return to OPR command level to start the components of your new GALAXY system:

```

!Opr RET
OPR>

```

To start all the components of the new GALAXY system, use the Take command to read commands from the file SYSTEM:SYSTEM.CMD:

```

OPR>Take SYSTEM:SYSTEM.CMD RET
OPR>
16:54:37 Batch-Stream 0   -- Set Accepted --
16:54:37 Batch-Stream 1   -- Set Accepted --
16:54:37 Batch-Stream 2   -- Set Accepted --
16:54:37 Batch-Stream 0   -- Startup Scheduled --
16:54:37 Batch-Stream 1   -- Startup Scheduled --
16:54:37 Batch-Stream 2   -- Startup Scheduled --
OPR>Take SYSTEM:PRINTER.CMD RET
OPR>
16:54:37 Printer 0       -- Set Accepted --
16:54:37 Printer 0       -- Startup Scheduled --
OPR>

```

To verify that each component of your GALAXY system is running properly, you should now type a Show Status command:

```

OPR>Show Status RET
OPR>
16:54:57          -- System Device Status --
Batch-Stream Status:
  Strm      Status
  ----      -
      0 Idle
      1 Idle
      2 Idle
Printer Status:
  Unit      Status
  ----      -
      0 Idle
OPR>

```

To end the OPR program, type the Exit command:

```

OPR>Exit RET
!

```

E.6 Making Your New GALAXY System Permanent

Once your new system is successfully running, you may want to delete all the files from `PS:<SRC.7.GALAXY.BUILD>` and `PS:<OLD-GALAXY>`, to make more disk space available.

Note

Be certain your new GALAXY system is running properly **before** deleting the old files.

Use the Delete command to remove the old GALAXY files from disk in order to make more disk space available. The system prints a message for each file it deletes:

```

!Delete PS:<SRC.7.GALAXY.BUILD>*.* ,PS:<OLD-GALAXY>*.* RET
.
.
.
!

```

Note

If your GALAXY files are in a directory other than PS:<SRC.7.GALAXY.BUILD>, use that directory name in place of “PS:<SRC.7.GALAXY.BUILD>” in the Delete command.

To supervise your new GALAXY system, run OPR:

```
!Opr RET  
OPR>
```


Appendix F

TDBOOT Command Summary

The contents of this appendix have been created mechanically from the online help texts contained within the TDBOOT program. Hopefully, this approach encourages the accuracy and completeness of both this printed form and the online help texts. Because this is automatically generated, the typographical conventions used in this appendix vary slightly from those found elsewhere in this manual. This appendix reflects TDBOOT Version 3.1(111)-1.

F.1 Macro-console commands

The commands that are processed by the macro-console are listed below. (Micro-console commands, all of which start with the period (“.”) character, are listed in the next section.)

The syntax “C[ONTINUE]” in a command or other keyword means that “C” is an explicit shorthand for “CONTINUE”, even though it might otherwise be ambiguous.

BOOT Load an executable file into memory and start it. The optional argument can specify a path to a file on a TOPS-20 or TOPS-10 structure, a network node, a specific SCSI tape device nexus, an operating system tape specifier, or a sequential SCSI tape name.

BOOT path

The path argument can have one of the following formats:

<code>str:<dir.sub>name.typ.gen</code>	To specify a TOPS-20 disk file
<code>str:name.ext[p,pn,sub]</code>	To specify a TOPS-10 disk file
<code>node::'remote-file'</code>	To specify a network file (not implemented yet)
<code># slot bus target [unit]</code>	To specify an explicit SCSI nexus
<code>MTAnnn:</code>	To specify a SCSI tape by its sequential nexus position
<code>channel, controller, unit</code>	Operating system tape specifier

If no argument is specified, the default boot string is used (see the “DEFINE BOOT” command). If no structure, node, “#”, “MTAnnn:”, or operating system tape specifier is specified, a TOPS-20 structure is located whose AUTOMATIC-STARTUP parameter has been set to this system’s ID number. If none is found, the TOPS-10 structure with the name beginning with “DSK” and the lowest letter “x” (starting at “A”) is used. If a TOPS-20 structure is used, the remaining file fields are defaulted from the following file specification:

```
<SYSTEM>MONITR.EXE.0
```

or if a TOPS-10 structure is used, the remaining file fields are defaulted from the following file specification:

```
SYSTEM.EXE[1,4]
```

For a network file, specify a node name terminated by two colons. Following the node name should be a remote file specification enclosed in either single or double quotes (either ‘ or ”). The interpretation of the specification is left to the remote system.

*** Network files are not yet implemented.

For an explicit SCSI nexus, enter a pound sign (“#”), followed by the slot number of a mass-storage controller, a SCSI bus number, a SCSI device ID, and finally, an optional SCSI logical unit number. This method cannot be used to boot from a structure, since a structure could include more than one SCSI disk.

For a sequential SCSI tape, specify “MTAnnn:” where “nnn” is the sequential number of the desired tape. The tapes are numbered starting at zero as they are encountered while scanning the mass-storage controllers, SCSI buses, SCSI devices, and SCSI logical units, in that order. Thus MTA2: specifies the third SCSI tape encountered during the scan.

For an operating system tape specifier, specify three decimal numbers: channel, controller, and unit separated by commas. The decimal channel number is constructed by counting buses. Buses 0 thru 3 on the mass-storage controller in the lowest numbered slot number are channels 0 thru 3; buses 0 thru 3 on the mass-storage controller in the next lowest slot number are channels 4 thru 7; the next would be 8 thru 11, and so on. For SCSI devices with LUN zero, the controller number can be -1 and the unit number is the decimal target number. Otherwise, the controller number is the decimal target number, and the unit number is the LUN. For example, if there are two mass-storage controllers in the system, the tape with target number 8 and LUN 0 on bus 1 in slot 7 (the higher numbered mass-storage controller slot) would be accessed by:

```
5,-1,8
```

BOOT switches

Switches should be entered after all other arguments.

```
/CACHE
```

Enable caching of loaded pages. This switch implies `/CST`. The cachable bit is set in all CST entries. Do not use this option if the loaded program uses direct I/O between system memory and the XRH or XNI unless the program clears the appropriate cachable bits in the CST first. This switch must be specified with the first program loaded into memory.

`/CST`

When loading the program into memory, create a CST. This CST will be used when the program is run (by `/START` now, or `START` later). This switch must be specified with the first program loaded into memory.

`/CORE-DUMP`

Set the data mode to Core Dump format. In this mode, each 36-bit word corresponds to five 8-bit bytes. The first four 8-bit bytes contain bits 0-7, 8-15, 16-23, and 24-31, respectively. The last byte contains bits 32-35 right justified. This switch applies only to sequential-access devices (tapes).

`/DEBUG`

Set `CF%DBG` in control flags (`WCTRLF/RCTRLF`) when starting program. This can be used to tell a program (such as the monitor) that it should run in debug mode.

`/DDT`

Load `TDBoot` and its copy of `DDT` into memory and start `DDT` with symbols set up for the loaded program. This switch must be specified with the first program loaded into memory.

`/HIGH-DENSITY`

Set the data mode to High-Density format. In this mode, two 36-bit words correspond to nine 8-bit bytes. The first four 8-bit bytes contain bits 0-7, 8-15, 16-23, and 24-31 of the first word, respectively. The fifth byte contains bits 32-35 of the first word in the high-order 4 bits and bits 0-3 of the second word in the low-order 4 bits. The last four 8-bit bytes contain bits 4-11, 12-19, 20-27, and 28-35 of the second word, respectively. This switch applies only to sequential-access devices (tapes).

`/MERGE`

Merge the specified program with any programs already in memory.

`/NOLONG-TRANSFERS`

Do not use multi-page transfers. This switch applies only to direct-access devices (disks). By default, `TDBoot` will attempt to combine transfers involving sequential pages on disk into large groups called "long transfers". This switch forces each page to be processed as a separate transfer.

`/PROTECT`

Load this program into protected locations in high physical memory along with the paging data. This is useful for keeping a program out of the way of a program loaded into low memory.

/REWIND

Rewind the SCSI device before performing the requested operation.

/START:address

After loading the specified program, start it at the address supplied. If the address begins with a “+” is interpreted as an entry-vector offset.

/W

“/W” is a synonym for “/REWIND”, provided for backwards compatibility.

CLEAR Clear various system variables.

CLEAR ADDRESS-BREAK

Clear any address break set for the current program.

CLEAR CACHE

Clear CPU cache. Any modified CPU cache entries are written back to memory, and all cache entries are invalidated. If the /INVALIDATE switch is present, the CPU cache is merely invalidated without writing modified entries back to memory.

CLEAR CONFIGURATION slot

Clear the configuration for the specified slot. Until it is configured again the slot will be treated as if no device is present.

CLEAR MEMORY

Clear all or selected system memory modules so that they contain zeros and good parity. The STATIC and COMBINED memory tests are performed first to insure proper operation (see the TEST MEMORY command for more details on these tests).

CLEAR MEMORY arguments

The optional decimal slot number argument (0–15) indicates a specific memory to clear. No argument will clear all memories in the system. Switches may follow the slot number.

CLEAR MEMORY switches

Switches should be entered after all other arguments.

/NOCACHING

Disable use of the cache to accelerate the operation. This will force direct memory accesses even though they are slower. Use this if you think there are problems with the cache. Caching will automatically be disabled if the CACHE-TEST startup parameter is disabled, or if there were initialization errors. Since using the cache requires using the pager, caching is implicitly disabled if paging is disabled (PAGER-TEST is disabled, or /NOPAGING was specified).

/NOPAGING

Disable use of paging to accelerate the operation. This will force the use

of PMOVEM even though it is slower. Use this if you think there are problems with the pager. Paging will automatically be disabled if the PAGER-TEST startup parameter is disabled, or if there were initialization errors. Since using the cache requires using the pager, caching is implicitly disabled if paging is disabled.

CLEAR NVRAM

Clear entire saved system configuration and reinitialize.

C[ONTINUE] [address]

Resume running an interrupted program. The optional argument specifies the new PC at which to continue. With or without the argument, the flags remain unchanged.

DAYTIME Print the current date and time as stored in the hardware timebase. If the hardware timebase does not appear to be set properly, a message to that effect is printed instead of the date and time.

DDT Load TDBoot into memory, mapped in the same section it normally runs at, and enter DDT via an unsolicited breakpoint. You may type \$P (escape, then "P") to proceed TDBoot running in system memory.

DEFINE Change certain static information which is saved in nonvolatile RAM. (The LIST command displays these values.)

DEFINE AUTO-BOOT-DELAY

Define the auto-boot delay for this processor. The argument is the decimal delay time in seconds (range 0-255). This is how long this processor waits prior to initiating an automatic boot after power-on. This value is saved in nonvolatile RAM.

DEFINE AUXILIARY-PORT

Define the state of the auxiliary terminal port. The state is saved in nonvolatile RAM and is used to set up the auxiliary port during initialization.

DEFINE AUXILIARY-PORT OFF

Turn off the auxiliary port.

DEFINE AUXILIARY-PORT ON

Turn on the auxiliary port.

DEFINE BOOT-DEFAULTS arguments

Define the default BOOT command arguments to be the remainder of the command line. See the BOOT command for the usage of this string. This value is saved in nonvolatile RAM.

DEFINE CONFIGURATION {slot | *}

Clear the saved system configuration for the specified slot, poll that slot, and save the new configuration. With "*" as the argument, clear, poll, and save the configuration for all system slots. Unlike other commands, the "*" is not the default and must be explicit. The new configuration information is saved in

nonvolatile RAM.

DEFINE DAYLIGHT-SAVINGS

Define the default daylight savings handling. When the processor is initialized, this value is copied into dynamic storage (same as the “SET DAYLIGHT-SAVINGS” command). This value is saved in nonvolatile RAM.

DEFINE DAYLIGHT-SAVINGS ALWAYS

Always process date and time with daylight savings time in effect. This may be useful if the AUTOMATIC setting is not appropriate for your site.

DEFINE DAYLIGHT-SAVINGS AUTOMATIC

Automatically determine when to process date and time with daylight savings time in effect. The determination is based on the rules in effect in most of the USA at the time of this writing. If this does not seem to be appropriate, consider using the ALWAYS or NEVER settings.

DEFINE DAYLIGHT-SAVINGS NEVER

Never process date and time with daylight savings time in effect. This may be useful if the AUTOMATIC setting is not appropriate for your site.

DEFINE DUMP-DEFAULTS arguments

Define the default DUMP command arguments to be the remainder of the command line. See the DUMP command for the usage of this string. This value is saved in nonvolatile RAM.

DEFINE IP-ADDRESS a.b.c.d

Define the Internet Protocol Address for a network port. The first argument is the decimal slot number (1–15) of a network controller. The second argument is the port number (0–3) of a network connection. The optional third argument is an IP address in dot format (a.b.c.d), where each letter is replaced by a decimal number (0–255). If the third argument is not supplied, the IP address is cleared. These settings are not currently used, but are reserved for diagnostics.

DEFINE SCSI-ID slot bus {id | OFFLINE}

Define the SCSI target number for the mass-storage controller connected to a SCSI bus. The arguments specify the mass-storage controller, SCSI bus, and what ID to use (or to set the bus off line). These values are saved in nonvolatile RAM. The “slot” is the decimal slot number (1–15) of a mass-storage controller. The “bus” is the SCSI bus number (0–3). The “id” is the decimal SCSI target number (0–15) which the mass-storage controller should use to identify itself on the specified bus. If you expect to communicate with any 8-bit SCSI devices, do not use SCSI target numbers above 7, as these devices which don’t implement target numbers greater than 7. The OFFLINE keyword will set the specified bus off line. That is, it will not be used for any purpose until a valid SCSI target number has again been set up for it.

DEFINE SYNC-DELAY seconds

Define the synchronization delay for this processor. The argument is the deci-

mal delay time in seconds (range 0-255). This value is used for multi-processor synchronization. This value is saved in nonvolatile RAM.

DEFINE TIMEZONE hh:mm

Define the local timezone as the specified number of decimal hours and minutes different from Greenwich. The value may be between -12:00 and 12:00 inclusive. Timezones west of Greenwich are negative, while timezones east of Greenwich are positive. The values -12:00 and 12:00 are the same time, but on opposite sides of the international date line, and thus differ by 1 day. For example, the United States eastern timezone would be -5:00 (standard time is 5 hours and 0 minutes earlier than Greenwich). When the processor is initialized, this value is copied into dynamic storage (same as the "SET TIMEZONE" command). Users should note that the sign of this value differs from the equivalent one used in TOPS-20 (in the 7-SETSPD program when read from the 7-CONFIG.COMD file). This value is saved in nonvolatile RAM.

D[EPOSIT]

Deposit into various system entities.

D[EPOSIT] A-MEMORY address

Store data in the microcode private memory (MemA). Supply an address (0-17777) followed by a 36-bit data word.

D[EPOSIT] CACHE a1 a2 a3

Store data in the CPU cache. See the description of the DWRC SH instruction for the format of the three arguments.

D[EPOSIT] DEVICE-REGISTER slot address data

Perform a device-control cycle (also known as a "write to I/O space") to write data into a device register. The first argument is the slot number of the desired device. The second argument is the in-module address of the register. The third argument is the data to be written.

D[EPOSIT] FLAGS-AND-CONTEXT data

Store new values for the processor flags, and current-context and previous-context context AC blocks, and previous-context section. The only argument contains the processor flags in bits 0-17, the current AC block number in bits 18-20, the previous-context AC block number in bits 21-23, and the previous-context section in bits 24-35. Care should be used when manipulating the processor flags and context, because this may adversely affect program operation.

D[EPOSIT] MEMORY slot address data

Using a physical memory address, write data into system memory. The first argument is the slot number of the desired device. The second argument is the in-module memory address. The third argument is the data to be written.

D[EPOSIT] NVRAM address data

Store data in the nonvolatile RAM. Supply an address (0-17777) followed by an 8-bit data byte.

D[EPOSIT] PC address

Set the current program counter. The argument is the address at which to continue program execution. Care should be used when manipulating the program counter, because this may adversely affect program operation.

D[EPOSIT] REGISTER n data

Write data into the current-context registers. The first argument is an octal register number (0–17). The second argument is the data to be written to the register. Following the second argument, you may type `/BLOCK:n` to specify the AC block number to examine. The default is the current AC block as specified in the flags and context word.

/BLOCK:n

Specify the AC block (0–7) to deposit in. The default is the current AC block as specified in the flags and context word.

D[EPOSIT] SYSTEM-VIRTUAL address data

Using a system virtual address, write data into system memory. The first argument is the system virtual address. The second argument is the data to be written. The `/DETAIL` switch prints the details of the virtual address resolution.

/DETAIL

Include a detailed printout of the virtual address resolution process while performing the requested operation. This is useful for determining the source of a page fault or other error when referencing the specified virtual address.

D[EPOSIT] TRANSLATION-BUFFER a1 a2 a3

Store data in the CPU page-translation buffer (also known as the pager). See the description of the `DWRPTB` instruction for the format of the three arguments. Since the microcode clears the pager upon exit from the macro-console, this command is of limited utility. It can be used more effectively from a copy of `TDBoot` running in system memory.

D[EPOSIT] USER-VIRTUAL address data

Using a user virtual address, write data into system memory. The first argument is the user virtual address. The second argument is the data to be written. The `/DETAIL` switch prints the details of the virtual address resolution.

/DETAIL

Include a detailed printout of the virtual address resolution process while performing the requested operation. This is useful for determining the source of a page fault or other error when referencing the specified virtual address.

DIRECTORY [path]

List all selected files in a TOPS-20 or TOPS-10 directory. The optional argument can specify a path to the directory, as well as which files to list. There are also switches available to select what information will be printed for each file. This command is valid

only for disk devices. The path for a TOPS-20 disk has the following format:

```
str:<dir.sub>name.typ.gen
```

Any unspecified file fields are default from the following file specification:

```
<SYSTEM>MONITR.EXE.*
```

The path for a TOPS-10 structure has the following format:

```
str:name.ext[p,pn,sub]
```

Any unspecified file fields are default from the following file specification:

```
SYSTEM.EXE[1,4]
```

/CREATION

Print date and time of creation for each file.

/LENGTH

Print length for each file. For TOPS-20, this is the number of bytes and the byte size; for TOPS-10, this is the number of words.

/L[ONG]

This is a shorthand for /SIZE, /LENGTH, and /WRITE.

/READ

Print date and time of last read for each file.

/SIZE

Print size for each file. For TOPS-20, the size is in pages; for TOPS-10, the size is in blocks.

/WRITE

Print date and time of last write for each file.

DISABLE parameter-list

Disable various system startup parameters. Provide a list of parameter names separated by commas.

The parameter names and their meanings are listed with the ENABLE command.

DUMP

Copy the entire physical system memory image (and some of the processor state) to the file specified by the optional path argument. The optional argument can specify a path to a file on a TOPS-20 or TOPS-10 structure, a network node, a specific SCSI tape device nexus, a sequential SCSI tape name, or an operating system tape specifier. The default "path" is specified by the "DEFINE DUMP" command. If the target device is a disk structure, the given file must exist and it must be be capacious enough to hold all of system memory.

DUMP path

The syntax of the path argument to the DUMP command is the same as for the BOOT command. The default DUMP string is specified via the DEFINE DUMP command.

If a structure is used, the remaining file fields are defaulted from the following file specification for TOPS-20:

```
<SYSTEM>DUMP.EXE.0
```

or the following file specification for TOPS-10:

```
CRASH.EXE[1,4]
```

DUMP switches

Switches should be entered after all other arguments.

/ALL-MEMORY

Dump or save all of physical memory, if possible.

/CORE-DUMP

Set the data mode to Core Dump format. In this mode, each 36-bit word corresponds to five 8-bit bytes. The first four 8-bit bytes contain bits 0-7, 8-15, 16-23, and 24-31, respectively. The last byte contains bits 32-35 right justified. This switch applies only to sequential-access devices (tapes).

/HIGH-DENSITY

Set the data mode to High-Density format. In this mode, two 36-bit words correspond to nine 8-bit bytes. The first four 8-bit bytes contain bits 0-7, 8-15, 16-23, and 24-31 of the first word, respectively. The fifth byte contains bits 32-35 of the first word in the high-order 4 bits and bits 0-3 of the second word in the low-order 4 bits. The last four 8-bit bytes contain bits 4-11, 12-19, 20-27, and 28-35 of the second word, respectively. This switch applies only to sequential-access devices (tapes).

/MAPPED-MEMORY

Dump or save only exec mapped memory.

/NOLONG-TRANSFERS

Do not use multi-page transfers. This switch applies only to direct-access devices (disks). By default, TDBoot will attempt to combine transfers involving sequential pages on disk into large groups called "long transfers". This switch forces each page to be processed as a separate transfer.

/REWIND

Rewind the SCSI device before performing the requested operation.

/W

“/W” is a synonym for “/REWIND”, provided for backwards compatibility.

ENABLE Enable various system startup parameters. Provide a list of parameter names separated by commas.

ENABLE AUTO-BOOT

The AUTO-BOOT startup parameter. If this parameter is enabled at startup, after a short delay (see DEFINE AUTO-BOOT-DELAY), the system will attempt to boot automatically using the defaults set up via the DEFINE BOOT command.

ENABLE BUS-POLL

The BUS-POLL startup parameter. If this parameter is enabled at startup, the system will poll the bus to determine the hardware configuration.

ENABLE CACHE-TEST

The CACHE-TEST startup parameter. If this parameter is enabled at startup, the system will perform a diagnostic check of the cache. This parameter must be enabled to automatically utilize the cache during some memory tests (see TEST MEMORY).

ENABLE CLEAR-MEMORY

The CLEAR-MEMORY startup parameter. If this parameter is enabled at startup, the system will test memory, which leaves all of memory with good parity. If TEST-MEMORY is disabled and CLEAR-MEMORY is enabled, all memory is cleared at startup. This prevents any latent parity errors from affecting system behavior. Prior to clearing memory, the static memory tests are performed (see TEST MEMORY/STATIC). If the PAGER-TEST and CACHE-TEST parameters are enabled, then the combined tests are performed (see TEST MEMORY/COMBINED).

ENABLE CONFIGURE-MEMORY

The CONFIGURE-MEMORY startup parameter. If this parameter is enabled at startup, the system will configure all available memory. Prior to configuring memory, each memory may also be tested (see TEST-MEMORY) or cleared (see CLEAR-MEMORY).

ENABLE MASS-STORAGE-CONFIGURATION

The MASS-STORAGE-CONFIGURATION startup parameter. If this parameter is enabled at startup, the system will report the mass-storage controller device configuration and start the disks.

ENABLE PAGER-TEST

The PAGER-TEST startup parameter. If this parameter is enabled at startup, the system will perform a diagnostic check of the pager. This parameter must be enabled to automatically utilize the pager during some memory tests (see TEST MEMORY), and when clearing memory (see CLEAR MEMORY). Since the pager is needed to use the cache during the TEST MEMORY and CLEAR MEMORY commands, the use of the cache is implicitly disabled when this parameter is disabled.

ENABLE TEST-MEMORY

The TEST-MEMORY startup parameter. If this parameter is enabled at startup, the system will perform a diagnostic check of the system memory. Prior to other testing, the static memory tests are performed (see TEST MEMORY/STATIC). If the PAGER-TEST and CACHE-TEST parameters are enabled, then the combined tests are performed (see TEST MEMORY/COMBINED).

ENABLE *

This is a simple abbreviation used to ENABLE or DISABLE all startup parameters.

E[XAMINE]

Examine various system entities.

E[XAMINE] A-MEMORY address

Print the contents of the specified location in the microcode private memory (MemA). The argument is the octal address (0-17777).

E[XAMINE] CACHE set-and-line-index

Print the contents of the specified CPU cache line. See the description of the DRDCSH instruction for the format of the argument.

E[XAMINE] DEVICE-REGISTER slot address

Perform a device status request cycle (also known as a “read from I/O space”) to read data from a device register. The next argument is the in-module address of the register.

E[XAMINE] FLAGS-AND-CONTEXT

Print current program flags and context information.

E[XAMINE] INTERRUPT

Print current interrupt system status.

E[XAMINE] MASS-STORAGE slot

Print the contents of a location in one of the memory spaces in the specified mass-storage controller. The first argument is the slot number of the desired device. The second argument is a register number (0-17), or a keyword describing the name of the space to be examined.

E[XAMINE] MASS-STORAGE slot register

Print the result of reading the specified mass-storage controller register. This argument is the octal number of the desired register. The optional next argument specifies an octal address to be supplied in the register address bits as appropriate for the register.

E[XAMINE] MASS-STORAGE slot ALU

Print the contents of a location in the ALU of a mass-storage controller. The next argument is the location to read in the mass-storage controller ALU.

E[XAMINE] MASS-STORAGE slot COMMUNICATIONS-REGION

Print the contents of the mass-storage controller communications region register. The optional argument specifies an offset (0-7) in the communications region whose contents is printed as well.

E[XAMINE] MASS-STORAGE slot DRAM

Print the contents of the specified DRAM address of the specified mass-storage controller. The next argument is the address of the entry to read in the mass-storage controller DRAM.

E[XAMINE] MASS-STORAGE slot ERROR-ADDRESS

Print the contents of the error-address register of the specified mass-storage controller.

E[XAMINE] MASS-STORAGE slot HASH-TABLE

Print the contents of the specified hash-table entry (two words) in the DRAM of the specified mass-storage controller. The next argument is the hash-table entry to read in the specified mass-storage controller DRAM. The actual words read from DRAM are the address times two and the address times two plus one.

E[XAMINE] MASS-STORAGE slot MICROCODE-VERSION

Print the contents of the microcode version register of the specified mass-storage controller.

E[XAMINE] MASS-STORAGE slot SERIAL-NUMBER

Print the contents of the serial number register of the specified mass-storage controller.

E[XAMINE] MASS-STORAGE slot SRAM

Print the contents of a location in the SRAM of the specified mass-storage controller. The next argument is the location to read in the SRAM.

E[XAMINE] MASS-STORAGE slot STATUS

Print the contents of the status register of the specified mass-storage controller.

E[XAMINE] M[EMORY] slot address

Print the contents of the specified physical memory location. The next argument is the in-module memory address.

E[XAMINE] NETWORK slot

Print the interpreted contents of a register in the specified Network Controller. The first argument is the slot number of the desired network controller. The second argument is the name of the register to be examined.

E[XAMINE] NETWORK slot BUS-STATUS

Prints the contents of the bus status register of the specified network controller.

E[XAMINE] NETWORK slot STATUS

Print the contents of the status register of the specified network controller.

E[XAMINE] NEXT [n]

Print the logically next value of the preceding EXAMINE or DEPOSIT command, whichever was most recent. The optional argument is a decimal repeat count. This is the default if no arguments are specified after the EXAMINE command.

E[XAMINE] NVRAM address

Print the contents of the specified byte in the nonvolatile RAM. The argument is an octal address (0-17777).

E[XAMINE] PC

Print current program PC.

E[XAMINE] PREVIOUS [n]

Print the logically previous value from the most recent EXAMINE or DEPOSIT command. The optional argument is a decimal repeat count.

E[XAMINE] REGISTER {n | *}

Examine the specified register (octal argument) or all registers (“*” argument). Following the argument, you may type /BLOCK:n to specify the AC block number to examine. The default is the current AC block as specified in the flags and flags and context word.

/BLOCK:n

Specify the AC block (0-7) to examine. The default is the current AC block as specified in the flags and context word.

E[XAMINE] SYSTEM-VIRTUAL address

Print the contents of the specified location in system memory. The argument is a system virtual address. The /DETAIL switch prints the details of the virtual address resolution.

/DETAIL

Include a detailed printout of the virtual address resolution process while performing the requested operation. This is useful for determining the source of a page fault or other error when referencing the specified virtual address.

E[XAMINE] TRANSLATION-BUFFER set-and-line-index

Print the contents of the specified CPU page-translation buffer entry. See the description of the DRDPTB instruction for the format of the argument. Since the microcode clears the pager upon exit from the macro-console, this command will find only valid data in the pager on the first macro-console command following the placing of that data in the pager. It can be used more effectively from a copy of TDBoot running in system memory.

E[XAMINE] USER-VIRTUAL address

Print the contents of the specified location in system memory. The argument is a

user virtual address. The `/DETAIL` switch prints the details of the virtual address resolution.

/DETAIL

Include a detailed printout of the virtual address resolution process while performing the requested operation. This is useful for determining the source of a page fault or other error when referencing the specified virtual address.

E[XAMINE] *

Examine the macro PC, flags, context, PI, and current register block.

EXIT Resume running an interrupted program. If debugging TDBoot, this command will HALT the program.

FORMAT Format a disk for the operating system.

This command performs the SCSI Mode Select and Format Unit operations needed to set the disk for 512-word (2304 byte, long) physical records or 128-word (576 byte, short) physical records. After “hard” formatting the disk, this command goes on to perform other bookkeeping operations (“soft formatting”) that are needed to make the disk ready for use by the operating system.

Restrictions: this command takes a long time. Disk types known to the program can be formatted; unknown disks may be formatted if they comply with any version of “standard behavior”.

FORMAT arguments

The arguments specify which disk to format as:

slot bus target-id [unit]

Where “slot” is the decimal slot number (1–15) of the mass-storage controller, “bus” is the SCSI bus number (0–3), “id” is the decimal SCSI target number (0–15) of the desired disk drive, and “unit” is the optional SCSI logical unit number (0–7) within the target.

FORMAT switches

Switches should be entered after all other arguments.

/HARD

Perform a hard format followed by a soft format.

/LONG

Format using only long (2304 byte/512 word) blocks.

/SHORT

Format using only short (576 byte/128 word) blocks.

/SOFT

Perform only a soft format. The unit must already have an acceptable hard format.

/TYPE:keyword

Select a specific format algorithm. The following algorithms are available:

GENERIC	Default format algorithm
HP	Algorithm applied to Hewlett-Packard drives
SEAGATE	Algorithm applied to Seagate drives

GET [path]

GET is an alternate name for the LOAD command.

H[ALT] Stop the currently running program. Certain commands are allowed only when the program is halted. The program may be resumed using the CONTINUE command.

HELP The HELP command with no argument can be used to get a list of commands and brief descriptions. To get a more detailed description of a particular command, give the command name as the argument to the HELP command (e.g., "HELP HALT"). This may be extended for commands with multiple keywords (e.g., "HELP EXAMINE PC"). Commands with switches will provide help information by typing the command name followed by "switches" and one of the switches (e.g., "HELP BOOT switches /START:").

INTERRUPT

Set the interrupt request flag (AP%INT) in the processor flags accessible by RDAPR and continue the program. A program can use this to receive interrupt requests by enabling an interrupt when this bit is set or by sampling the bit periodically.

LIST Display various static system parameters stored in nonvolatile RAM. (These are the parameters that are changed by the DEFINE, ENABLE, and DISABLE commands.)

LIST AUTO-BOOT-DELAY

Print the auto-boot delay in seconds for this processor stored in nonvolatile RAM.

LIST AUXILIARY-PORT

Print the state of the auxiliary terminal port stored in nonvolatile RAM.

LIST BOOT-DEFAULTS

Print the default BOOT command argument stored in nonvolatile RAM.

LIST CONFIGURATION

Print the static bus configuration stored in nonvolatile RAM.

LIST DAYLIGHT-SAVINGS

Print the static DAYLIGHT-SAVINGS setting stored in nonvolatile RAM.

LIST DUMP-DEFAULTS

Print the default DUMP command argument stored in nonvolatile RAM.

LIST IP-ADDRESSES

Print the Internet Protocol Address for each network connection in the system stored in nonvolatile RAM.

LIST PARAMETERS

Print the saved parameter settings stored in nonvolatile RAM.

LIST SCSI-IDS

Print the static SCSI target ID number for each SCSI bus on each mass-storage controller in the system stored in nonvolatile RAM.

LIST SYNC-DELAY

Print the synchronization delay in seconds for this processor stored in nonvolatile RAM.

LIST TIMEZONE

Print the static local TIMEZONE value stored in nonvolatile RAM.

LIST *

Print the state of all information stored in nonvolatile RAM. This is the default if no arguments are specified after the LIST command.

L[OAD] Load an executable file into memory, but do not start it. The optional argument can specify a path to a file on a TOPS-20 or TOPS-10 structure, a network node, a specific SCSI tape device nexus, a sequential SCSI tape name, or an operating system tape specifier. GET is an alternate name for LOAD.

L[OAD] path

The path argument for LOAD has the same format as described for the path argument to the BOOT command.

L[OAD] switches

The switches for LOAD are the same as those described in the BOOT command. However, the /DEBUG and /START switches do not apply to the LOAD command.

REINITIALIZE

Clear the dynamic configuration data and rescan the system configuration (if enabled). This will also cause the cache and pager tests to be run, as well as testing or clearing memory, if enabled.

RESET Place the specified components and/or devices in their initial state. A slot number or a keyword may be entered.

RESET slot

The "slot" argument is a decimal slot number (0-15) indicating the single slot to reset. Slot 0 refers to this CPU.

/HARD

Reset the specified device(s) in a hard manner. A hard reset may not wait for data transfers to be completed, and thus could discard data destined for the disk or network.

/SHUTDOWN

Shut down the specified mass-storage controllers. The mass-storage controllers will attempt to complete all outstanding requests and flush their caches back to disk.

RESET BUS

Reset all bus devices except this CPU.

/HARD

Reset the specified device(s) in a hard manner. A hard reset may not wait for data transfers to be completed, and thus could discard data destined for the disk or network.

/SHUTDOWN

Shut down the specified mass-storage controllers. The mass-storage controllers will attempt to complete all outstanding requests and flush their caches back to disk.

RESET CPU

Reset only CPUs.

RESET CPU arguments

The optional decimal slot number argument (0–15) indicates a specific CPU to be reset (0 is a synonym for this CPU). A missing argument or “*” will reset all CPUs in the system.

RESET CPU switches

Switches should be entered after all other arguments.

/HARD

Reset the specified device(s) in a hard manner. A hard reset may not wait for data transfers to be completed, and thus could discard data destined for the disk or network.

RESET MASS-STORAGE

Reset only mass-storage controllers.

RESET MASS-STORAGE arguments

The following arguments may be used to reset a subset of the mass-storage devices:

slot bus target-id

The “slot” argument specifies a mass-storage controller, the “bus” argument specifies a SCSI bus, and the “id” argument specifies a SCSI target id number. A missing argument or “*” will select all slots, buses, or ids corresponding to the argument.

RESET MASS-STORAGE switches

Switches should be entered after all other arguments.

/HARD

Reset the specified device(s) in a hard manner. A hard reset may not wait for data transfers to be completed, and thus could discard data destined for the disk or network.

/SHUTDOWN

Shut down the specified mass-storage controllers. The mass-storage controllers will attempt to complete all outstanding requests and flush their caches back to disk.

RESET MEMORY

Reset only memories.

RESET MEMORY arguments

The optional decimal slot number argument (0-15) indicates a specific memory to be reset. A missing argument or "*" will reset all memories in the system.

RESET MEMORY switches

Switches should be entered after all other arguments.

/HARD

Reset the specified device(s) in a hard manner. A hard reset may not wait for data transfers to be completed, and thus could discard data destined for the disk or network.

RESET NETWORK

Reset only network controllers.

RESET NETWORK arguments

The following arguments may be used to reset a subset of the network ports:

slot port

The "slot" argument specifies a network controller, and the "port" argument specifies a network port. A missing argument or "*" will select all slots or ports corresponding to the argument.

RESET NETWORK switches

Switches should be entered after all other arguments.

/HARD

Reset the specified device(s) in a hard manner. A hard reset may not wait for data transfers to be completed, and thus could discard data destined for the disk or network.

RESET *

Reset all system devices, including this CPU. This is the default if no arguments

are specified after the RESET command.

/HARD

Reset the specified device(s) in a hard manner. A hard reset may not wait for data transfers to be completed, and thus could discard data destined for the disk or network.

/SHUTDOWN

Shut down the specified mass-storage controllers. The mass-storage controllers will attempt to complete all outstanding requests and flush their caches back to disk.

REWIND scsi-device-specifier

Rewind the specified SCSI device (usually a tape). The device is specified using one of the following formats:

# slot bus target [unit]	To specify an explicit SCSI nexus
MTAnnn:	To specify a SCSI tape by its sequential nexus position
channel, controller, unit	Operating system tape specifier

R[UN] [path]

RUN is an alternate name for the BOOT command.

SAVE Save the current program as an executable image to the file specified by the optional path argument. The optional argument can specify a path to a file on a TOPS-20 or TOPS-10 structure, a network node, a specific SCSI tape device nexus, a sequential SCSI tape name, or an operating system tape specifier. There is no default path. If the target device is a disk structure, the given file must exist and it must be capacious enough to hold the program. *** Not implemented yet ***

SAVE path

The syntax of the path argument to the SAVE command is the same as for the BOOT command. There is no default path string for the SAVE command; a path must be supplied.

If a structure is used, the remaining file fields are defaulted (note: no default filename). For TOPS-20 and TOPS-10, the default file type (extension) is ".EXE". For TOPS-20, the default generation is 0.

SAVE switches

The switches for the SAVE command are the same as those described for the DUMP command.

SCAN Check various memory elements for data patterns and/or errors.

SCAN CACHE

Check the CPU cache for data patterns and/or errors.

SCAN CACHE DATA value [mask]

Check the CPU cache for data patterns. The first argument is the search data to be matched against any cache data entry. The optional second argument is the mask to be anded with the cache data entry before comparing it to the search data. If the second argument is not specified, it defaults to all ones.

SCAN CACHE ERRORS

Check the CPU cache memory for errors. Each cache line, set, and word is examined and any cache data or tag parity, or both cache sets match errors are printed. This is the default if no arguments are specified after the SCAN CACHE command.

/REPAIR

When an error is encountered during the scan, an attempt is made to repair it. In the case of a parity error, this means rewriting the tag or data with good parity. In the case of both sets matched, one of the sets will be cleared if it has not been modified.

SCAN CACHE TAG value [mask]

Check the CPU cache memory for tag patterns. The first argument is the search tag to be matched against any cache tag entry. The optional second argument is the mask to be anded with the cache tag entry before comparing it to the search tag. See the description of the DRDCSH instruction E+1 word for the format of the tag. If the second argument is not specified, it defaults to all ones.

SCAN MEMORY

Check system memory for data patterns and/or errors.

SCAN MEMORY DATA value [mask]

Check system memory for data patterns. The first argument is the search data to be matched against any pager data entry. The optional second argument is the mask to be anded with the pager data entry before comparing it to the search data. If the second argument is not specified, it defaults to all ones. Only configured memories are scanned.

SCAN MEMORY ERRORS

Check system memory for parity errors. All memory modules which are configured are read and any parity errors are printed. This is the default if no arguments are specified after the SCAN MEMORY command.

/REPAIR

When an error is encountered during the scan, an attempt is made to repair it. In the case of a parity error, this means rewriting the data with good parity.

SCAN PAGER

Check the CPU pager for data patterns and/or errors.

SCAN PAGER DATA value [mask]

Check the CPU pager for data patterns. The first argument is the search data

to be matched against any pager data entry. The optional second argument is the mask to be anded with the pager data entry before comparing it to the search data. If the second argument is not specified, it defaults to all ones.

SCAN PAGER ERRORS

Check the CPU pager for errors. Each pager line and set is examined and any pager data or tag parity, or both pager sets match errors are printed. This is the default if not arguments are specified after the SCAN PAGER command.

/REPAIR

When an error is encountered during the scan, an attempt is made to repair it. In the case of a parity error, this means rewriting the tag or data with good parity. In the case of both sets matched, one of the sets will be cleared.

SCAN PAGER TAG value [mask]

Check the CPU pager for tag patterns. The first argument is the search tag to be matched against any pager tag entry. The optional second argument is the mask to be anded with the pager tag entry before comparing it to the search tag. If the second argument is not specified, it defaults to all ones. See the description of the DRDPTB instruction E+1 word for the format of the tag.

SET Change certain dynamic information saved in MemA (in a manner less permanent than the DEFINE command). (See also the SHOW command.)

SET AUXILIARY-PORT {ON | OFF}

Set the state of the auxiliary terminal port. The new state is applied immediately to the auxiliary port. The keywords and their meanings are listed with the DEFINE AUXILIARY-PORT command.

SET ADDRESS-BREAK

Set an address break after loading a program into memory. If you set an address break and then load the first program, the implicit program reset that is done will clear the break.

SET ADDRESS-BREAK arguments

The single argument is a 30-bit virtual address on which to break.

SET ADDRESS-BREAK switches

Switches should be entered after all other arguments.

/EXECUTE

Select an address break when an instruction is fetched from the specified address.

/READ

Select an address break when data is read from the specified address.

/USER

Select an address break when the specified address is accessed in user mode.

/WRITE

Select an address break when data is written to the specified address.

SET CONFIGURATION

Explicitly configure various parts of the machine.

SET CONFIGURATION slot

The “slot” argument is a decimal slot number (0–15) indicating the single slot to configure. That slot will be polled and the results saved in the dynamic configuration database for that slot. Any discrepancies between the saved configuration and the current configuration for that slot will be printed.

SET CONFIGURATION BUS

Explicitly poll each slot in the machine and initialize the dynamic configuration. It will also print any discrepancies between the saved configuration and the current configuration for each slot.

SET CONFIGURATION MASS-STORAGE

Set up the dynamic configuration of all mass-storage controllers in the system. Start all disks and report device configuration.

SET CONFIGURATION MEMORY

Configure memory. Each memory is tested for proper operation and configured for use.

/CLEAR

Enable clearing memory after configuration and testing. The default setting for this switch is the same as the previous “CONFIGURE MEMORY” command. At powerup, it is initialized to the same setting as that of the CLEAR-MEMORY startup parameter.

/FORCE

Reconsider any memory units that have been put off line. Such a unit will have to pass some tests before it is accepted. This setting is remembered as part of the dynamic configuration and will be used until the processor is reset or TDBoot is REINITIALIZED.

/FORWARD

Configure the memory in the forward direction; that is, with the lowest physical slot numbers having the lowest addresses. This setting is remembered as part of the dynamic configuration and will be used until the processor is reset or TDBoot is REINITIALIZED.

/NOCACHING

Disable use of the cache to accelerate the operation. This will force direct

memory accesses even though they are slower. Use this if you think there are problems with the cache. Caching will automatically be disabled if the `CACHE-TEST` startup parameter is disabled, or if there were initialization errors. Since using the cache requires using the pager, caching is implicitly disabled if paging is disabled (`PAGER-TEST` is disabled, or `/NOPAGING` was specified).

`/NOCLEAR`

Disable clearing memory after configuration and testing. The default setting for this switch is the same as the previous “`CONFIGURE MEMORY`” command. At powerup, it is initialized to the same setting as that of the `CLEAR-MEMORY` startup parameter.

`/NOFORCE`

Do not reconsider any memory units that have been taken off line. This setting is remembered as part of the dynamic configuration and will be used until the processor is reset or TDBoot is `REINITIALIZED`.

`/NOPAGING`

Disable use of paging to accelerate the operation. This will force the use of `PMOVEM` even though it is slower. Use this if you think there are problems with the pager. Paging will automatically be disabled if the `PAGER-TEST` startup parameter is disabled, or if there were initialization errors. Since using the cache requires using the pager, caching is implicitly disabled if paging is disabled.

`/NOTEST`

Disable memory testing after configuration. The default setting for this switch is the same as the previous “`CONFIGURE MEMORY`” command. At powerup, it is initialized to the same setting as that of the `TEST-MEMORY` startup parameter.

`/REVERSE`

Configure the memory in the highest physical slot number to have the lowest addresses. This setting is remembered as part of the dynamic configuration and will be used until the processor is reset or TDBoot is `REINITIALIZED`.

`/TEST`

Enable memory testing after configuration. The default setting for this switch is the same as the previous “`CONFIGURE MEMORY`” command. At powerup, it is initialized to the same setting as that of the `TEST-MEMORY` startup parameter.

`SET CONFIGURATION NETWORK`

Set up the dynamic configuration of all network connections in the system. There is currently no dynamic configuration to setup for the network controllers. This command is reserved for future expansion.

SET CONFIGURATION *

Perform the default system configuration which includes the BUS, MEMORY, and MASS-STORAGE. This is the default if no arguments are specified after the SET CONFIGURATION command.

SET DAYLIGHT-SAVINGS

Set default daylight savings handling.

SET DAYLIGHT-SAVINGS ALWAYS

Always process date and time with daylight savings time in effect. This may be useful if the AUTOMATIC setting is not appropriate for your site.

SET DAYLIGHT-SAVINGS AUTOMATIC

Automatically determine when to process date and time with daylight savings time in effect. The determination is based on the rules in effect in most of the USA at the time of this writing. If this does not seem to be appropriate, consider using the ALWAYS or NEVER settings.

SET DAYLIGHT-SAVINGS NEVER

Never process date and time with daylight savings time in effect. This may be useful if the AUTOMATIC setting is not appropriate for your site.

SET SCSI-ID slot bus {id | OFFLINE}

Set the initiator target ID for a connected SCSI bus. The arguments specify the mass-storage controller, SCSI Bus, and what ID to use (or to set the bus off line). This command changes only the dynamic configuration of the SCSI interface. The "SET CONFIGURATION BUS" and "SET CONFIGURATION *" commands will set the SCSI bus states from their static values saved via DEFINE SCSI-ID. The "slot" is the decimal slot number (1-15) of a mass-storage controller. The "bus" is the SCSI bus number (0-3). The "id" is the decimal SCSI target number (0-15) which the SCSI controller should use to identify itself on the specified bus. If you expect to communicate with any 8-bit SCSI devices, do not use SCSI target numbers above 7, as these devices which don't implement target numbers greater than 7. The OFFLINE keyword will set the specified bus off line. That is it will not be used for any purpose until a valid SCSI target number has again been set up for it.

SET TIMEZONE hh:mm

Set the current local timezone as the specified number of decimal hours and minutes different from Greenwich. The value may be between -12:00 and 12:00 inclusive. Timezones west of Greenwich are negative, while timezones east of Greenwich are positive. The values -12:00 and 12:00 are the same time, but on opposite sides of the international date line, and thus differ by 1 day. For example, the United States eastern timezone would be -5:00 (standard time is 5 hours and 0 minutes earlier than Greenwich). Users should note that the sign of this value differs from the equivalent one used in TOPS-20 (in the 7-SETSPD program when read from the 7-CONFIG.CMD file).

SHOW Show various system settings.

SHOW ADDRESS-BREAK

Print the contents of the program address break register.

SHOW AUXILIARY-PORT

Print the current state of the auxiliary terminal port.

SHOW CAPACITY

This command prints the capacity of all or selected direct-access devices. This is similar to the “SHOW CONFIGURATION MASS-STORAGE ... /CAPACITY” command, except that it includes only direct-access devices.

SHOW CAPACITY arguments

The following arguments may be used to include a subset of the mass-storage devices:

slot bus target-id

The “slot” argument specifies a mass-storage controller, the “bus” argument specifies a SCSI bus, and the “id” argument specifies a SCSI target id number. All arguments are in decimal. A missing argument or “*” will select all slots, buses, or ids corresponding to the argument.

SHOW CONFIGURATION

Print the dynamic system configuration. A slot number or a keyword may be entered.

SHOW CONFIGURATION slot

The “slot” argument is a decimal slot number (0–15) indicating the single slot for which to show the configuration. Slot 0 refers to this CPU. The slot number may be followed by switches appropriate to the type of device in that slot.

SHOW CONFIGURATION CPU

Show only the configuration of CPUs.

SHOW CONFIGURATION CPU arguments

The optional decimal slot number argument (0–15) indicates a specific CPU to include (0 is a synonym for this CPU). A missing argument or “*” will include all CPUs in the system.

SHOW CONFIGURATION CPU switches

Switches should be entered after all other arguments.

/LONG

Print a detailed configuration.

/SHORT

Print a minimum configuration.

SHOW CONFIGURATION MASS-STORAGE

Show only the configuration of mass-storage controllers.

SHOW CONFIGURATION MASS-STORAGE arguments

The following arguments may be used to include a subset of the mass-storage devices:

slot bus target-id

The “slot” argument specifies a mass-storage controller, the “bus” argument specifies a SCSI bus, and the “id” argument specifies a SCSI target id number. All arguments are in decimal. A missing argument or “*” will select all slots, buses, or ids corresponding to the argument.

SHOW CONFIGURATION MASS-STORAGE switches

Switches should be entered after all other arguments.

/CAPACITY

Print the capacity of direct-access devices. The /LONG switch is set implicitly by this switch.

/HOME-BLOCKS

Print the home block information used by TDBoot for direct-access devices. The /LONG switch is set implicitly by this switch.

/LONG

Print a detailed configuration.

/SHORT

Print a minimum configuration.

SHOW CONFIGURATION MEMORY

Show only the configuration of memory modules.

SHOW CONFIGURATION MEMORY arguments

The optional decimal slot number argument (0–15) indicates a specific MEMORY to include. A missing argument or “*” will include all memories in the system.

SHOW CONFIGURATION MEMORY switches

Switches should be entered after all other arguments.

/LONG

Print a detailed configuration.

/SHORT

Print a minimum configuration.

SHOW CONFIGURATION NETWORK

Show only the configuration of network controllers.

SHOW CONFIGURATION NETWORK arguments

The following arguments may be used to include a subset of the network ports:

slot port

The “slot” argument specifies a network controller, and the “port” argument specifies a network port. A missing argument or “*” will select all slots or ports corresponding to the argument.

SHOW CONFIGURATION NETWORK switches

Switches should be entered after all other arguments.

/LONG

Print a detailed configuration.

/SHORT

Print a minimum configuration.

SHOW CONFIGURATION SYSTEM

Show only the system ID and options, and the backplane serial number and options.

SHOW CONFIGURATION *

Show the configuration of all devices in the system. This is the default if no arguments are specified after the SHOW CONFIGURATION command.

/LONG

Print a detailed configuration.

/SHORT

Print a minimum configuration.

SHOW DAYLIGHT-SAVINGS

Print the current DAYLIGHT-SAVINGS setting.

SHOW HOME-BLOCKS

This command prints the home blocks of all or selected direct-access devices. This is similar to the “SHOW CONFIGURATION MASS-STORAGE ... /HOME-BLOCKS” command, except that it includes only direct-access devices.

SHOW HOME-BLOCKS arguments

The following arguments may be used to include a subset of the mass-storage

devices:

slot bus target-id

The “slot” argument specifies a mass-storage controller, the “bus” argument specifies a SCSI bus, and the “id” argument specifies a SCSI target id number. All arguments are in decimal. A missing argument or “*” will select all slots, buses, or ids corresponding to the argument.

SHOW MEMORY-STATUS [slot]

Print all or selected memory status. The optional decimal slot number argument (0–15) indicates a specific memory controller whose status should be printed. No argument indicates that status for all memory controllers should be printed. The printout includes both the status and error-summary registers. Note: reading the error-summary register clears the “Parity Error Detected” bit in the status register and releases the error-summary register to capture another error.

SHOW PAGE-FAIL

Print the data saved by the most recent page fail(s) in a user readable form. If this was the result of a recursive page fail, the preceding page fail information is also printed. Be careful that there are no intervening page fails or you will not get the results you expect.

SHOW PAGE-FAIL CONSOLE

Show page fail data which preceded the most recent recursive page failure which attempted to trap via the console.

SHOW PAGE-FAIL IO

Show page fail data which preceded the most recent recursive page failure which attempted to trap via the EPT.

SHOW PAGE-FAIL LAST

Show page fail data from the most recent page fail.

SHOW PAGE-FAIL ROM

Show page fail data which preceded the most recent recursive page failure which attempted to trap via the ROM.

SHOW POWER-STATUS

Print the current power status.

SHOW PROGRAM

Print information associated with the current program in memory.

SHOW PROGRAM ENTRY-VECTOR

Print the current program’s entry-vector location and length.

SHOW PROGRAM MEMORY-MAP

Print all or some of the contents of the current program’s memory map. By

default, all executive pages (those mapped via the EPT) will be printed.

/EXEC

Limit information printout to executive pages (those mapped via the EPT).

/NOPAGE-INFO

Suppress printout of information on pages in each section. With this switch, only information about sections is printed.

/SECTION:n

Limit printout to the specified section. The argument is an octal section number (0-7776).

/USER

Limit information printout to user pages (those mapped via the UPT).

SHOW PROGRAM PDVAS

Print the current program's program data vector addresses (PDVAs).

SHOW SCSI-IDS

Print the dynamic initiator target ID number for each SCSI bus on each mass-storage controller in the system.

SHOW STRUCTURE str:

Print information about the units in the specified structure. The argument is a 1-6 character structure name followed by a colon (":").

SHOW TIMEZONE

Print the current local TIMEZONE offset.

SHOW VERSION

Print the version numbers of the BOOT ROM program and the current CPU microcode.

SHOW ZONE-TABLE slot bus target [lun]

Print the zone table for the specified disk drive. The first argument is the decimal slot number (1-15). The second argument is the bus number (0-3). The third argument is the decimal target ID (0-15). The optional fourth argument is the logical unit number (LUN, 0-7, default 0).

SHUTDOWN

Set the shutdown request flag (AP%SHT) in the processor flags accessible by RDAPR and continue the program. A program can use this to receive shutdown requests by enabling an interrupt when this bit is set or by sampling the bit periodically.

S[TART]

Set the program counter and continue the program. The optional argument can specify the absolute virtual address to load into the PC or a "+" followed by an entry-vector offset. If no argument is present, entry-vector offset 0 is used. The program flags and

context are zeroed before starting, which starts in EXEC mode. If you do not wish the flags and context to be zeroed, consider using the “DEPOSIT FLAGS <n>” and “CONTINUE <pc>” commands. The optional switch /DEBUG will set CF%DBG in the control flags (WCTRLF/RCTRLF), and can be used to tell a program (such as the monitor) to run in debug mode.

TEST Perform testing of specific processor components and/or devices. A slot number or a keyword may be entered.

TEST slot

The “slot” argument is a decimal slot number (0–15) indicating the single slot to test. Slot 0 refers to this CPU. The slot number may be followed by switches appropriate to the type of device in that slot.

TEST BUS

Perform device specific tests on each slot except this CPU.

/REPEAT:n

Repeat the specified tests. The argument is the number of times to repeat.

TEST CACHE

Perform the cache tests on this CPU.

/REPEAT:n

Repeat the specified tests. The argument is the number of times to repeat.

TEST CPU

Perform only CPU tests. This currently includes the cache and pager tests.

TEST CPU arguments

The optional decimal slot number argument (0–15) indicates a specific CPU to test (0 is a synonym for this CPU). A missing argument or “*” will test all CPUs in the system.

TEST CPU switches

Switches should be entered after all other arguments.

/REPEAT:n

Repeat the specified tests. The argument is the number of times to repeat.

TEST INTERVAL-TIMER

Run the interval timer at various interval values and compute the actual interval using RDTIME. Prints the interval being tested followed by the mean and variance for that interval.

/INTERVAL:n

Select a specific interval to test. The argument specifies the interval from 256 to 32768 microseconds. It will be rounded to the nearest multiple of 128.

/LOOP-ON-ERROR

Continue testing after printing an error.

/QUIET

Suppress informational output during testing. You will still get output if an error occurs.

/SAMPLES:n

The argument specifies the number of samples to take before computing average and variance. The default value is 100.

TEST MASS-STORAGE

Perform the mass-storage controller tests on all or selected mass-storage controllers and SCSI devices.

TEST MASS-STORAGE arguments

The following arguments may be used to select a subset of the mass-storage devices:

slot bus target-id unit

The “slot” argument specifies a mass-storage controller, the “bus” argument specifies a SCSI bus, the “id” argument specifies a SCSI target id number, and the “unit” argument specifies a SCSI logical unit number. A missing argument or “*” will select all slots, buses, ids, or units corresponding to the argument.

TEST MASS-STORAGE switches

Switches should be entered after all other arguments.

/BYTE-OFFSET:n

Specify the byte offset to use for data transfers. This can be used to test transfers which end in the middle of a word. (Byte-offset is effective only during tests of the industry-compatible data format.)

/COMPARE

During data tests, do write, read, and compare.

/CORE-DUMP

Test core dump (40-bit) data format.

/DEVICE-BUFFERS

Test the SCSI BUS. This consists of a WRITE BUFFER followed by a READ BUFFER and verify to each selected SCSI device which supports these commands.

/DRAM

Test mass-storage controller DRAM (internal buffer memory). Data is written from system memory into the DRAM and then read back and compared. If none of /DEVICE-BUFFERS, /DRAM, /MEDIA-TEST, /TIMING, or /VERIFY are specified, then /DEVICE-BUFFERS and /DRAM are the default.

/ENDING-OFFSET:n

Specify the ending word offset to use for data transfers. The argument is the offset backward from the end of the cache line of the last word to transfer (e.g., `/ENDING-OFFSET:7` would transfer only the first word of the cache line at the end of the transfer). This can be used to test transfers which end in the middle of a cache line.

/FLOATING-ZEROS

Test using a floating-zeros data pattern.

/FLOATING-ONES

Test using a floating-ones data pattern.

/FONES

Test using a floating-ones data pattern.

/FZEROS

Test using a floating-zeros data pattern.

/HIGH-DENSITY

Test high-density (36-bit) data format.

/INDUSTRY-COMPATIBLE

Test industry-compatible (32-bit) data format.

/JOHNSON-COUNTER

Test using a Johnson counter data pattern.

/MEDIA-TEST

Test the medium on the SCSI device by reading each block into system memory. For direct-access devices (disks), the entire medium is read.

/NOCOMPARE

During data tests, do write and read, but no compare.

/NOLONG-TRANSFERS

Do not use long transfers during media test. This switch is meaningful only in conjunction with the `/MEDIA-TEST` switch.

/ONES

Test using a data pattern of all ones.

/REPEAT:n

Repeat the specified tests. The argument is the number of times to repeat.

/RETRIES:n

Retry after errors when establishing the communications region. The argument is the number of times to retry before giving up.

/RONLY

During data tests, do read, but no write or compare.

/STARTING-OFFSET:n

Specify the starting word offset to use for data transfers. The argument is the offset forward from the beginning of the cache line of the first word to transfer (e.g., **/BEGINNING-OFFSET:7** would begin the transfer with the last word of the cache line). This can be used to test transfers which start in the middle of a cache line.

/TIMING

Test various timing parameters of selected SCSI devices.

/VERIFY

Test the medium on the SCSI device by issuing the SCSI VERIFY command. Since not all devices support this command, it may fail. If so, use the **/MEDIA-TEST** option, which will read each block into memory. For direct-access devices (disks), the entire medium is read.

/WONLY

During data tests, do write, but no read or compare.

/ZEROS

Test using a data pattern of all zeros.

TEST MEMORY

Test system memories.

TEST MEMORY arguments

The optional decimal slot number argument (0-15) indicates a specific memory to test. A missing argument or "*" will test all memories in the system.

TEST MEMORY switches

Switches should be entered after all other arguments.

/ADDRESS

Select the memory address test. If no memory tests are selected, all are performed. An "A" is printed to indicate that the address test is being initialized. During initialization, each word is written with its bus address word (BAW). An "a" is printed to indicate that the address-test compare has started. Each word of memory is checked to verify that it still contains the correct data.

/COMBINED

Select the combined memory tests. If no memory tests are selected, all are performed. A "C" is printed to indicate combined testing is in progress. A sequence of data patterns and addresses is used which will exercise the cache refill, writeback, and flush operations.

/DATA

Select the memory data test. If no memory tests are selected, all are performed. A “D” is printed to indicate that the data test is being initialized. During initialization, each word is written with the negative of its bus address word (BAW) minus two. A “d” is printed to indicate that the data-test compare has started. Each word of memory is checked to verify that it still contains the correct data.

/NOCACHING

Disable use of the cache to accelerate the operation. This will force direct memory accesses even though they are slower. Use this if you think there are problems with the cache. Caching will automatically be disabled if the CACHE-TEST startup parameter is disabled, or if there were initialization errors. Since using the cache requires using the pager, caching is implicitly disabled if paging is disabled (PAGER-TEST is disabled, or /NOPAGING was specified).

/NOPAGING

Disable use of paging to accelerate the operation. This will force the use of PMOVEM even though it is slower. Use this if you think there are problems with the pager. Paging will automatically be disabled if the PAGER-TEST startup parameter is disabled, or if there were initialization errors. Since using the cache requires using the pager, caching is implicitly disabled if paging is disabled.

/PROCEED-ON-ERROR

Causes the memory test to proceed despite errors. Errors will continue to be printed, and a failing memory will not be taken off line.

/REPEAT:n

Repeat the specified tests. The argument is the number of times to repeat.

/R[EADONLY]

Perform a READONLY memory test. This test is performed before any others to avoid changing the contents of memory. A “R” is printed to indicate readonly testing is in progress. Each word of memory is read to determine if it has good parity. This test makes use of the pager and cache to accelerate operation. See the /NOPAGING and /NOCACHING switches for more information.

/STATIC

Select the static memory tests. If no memory tests are selected, all are performed. An “S” is printed to indicate static testing is in progress. Each static test is performed on a cache line in memory. Data patterns of all ones, all zeros, a floating-one, and a floating-zero are written and compared in each word of the line. Cache line addresses of all zeros and all ones are performed first. This is followed by writing and detecting bad parity in cache line zero. Finally, cache line addresses first of a floating-one and then of a floating-zero are tested.

TEST NETWORK

Perform diagnostic tests on the specified network controllers.

TEST NETWORK arguments

The following arguments may be used to test a subset of the network ports:

slot port

The “slot” argument specifies a network controller, and the “port” argument specifies a network port. A missing argument or “*” will select all slots or ports corresponding to the argument.

TEST NETWORK switches

Switches should be entered after all other arguments.

/REPEAT:n

Repeat the specified tests. The argument is the number of times to repeat.

/TIMING

Measure the amount of time it takes to reset the network controller.

TEST PAGER

Test this CPU’s pager.

/REPEAT:n

Repeat the specified tests. The argument is the number of times to repeat.

TEST POWER-FAIL

Loop checking power status and reporting changes. Type ^C to terminate.

TEST TIME-BASE

Check RDTIME timebase for monotonicity and no large changes. Prints elapsed time approximately every 3 seconds.

/LOOP-ON-ERROR

Continue testing after printing an error.

/QUIET

Suppress informational output during testing. You will still get output if an error occurs.

TEST *

Test this CPU and each other system slot. This is the default if no arguments are specified after the TEST command.

/REPEAT:n

Repeat the specified tests. The argument is the number of times to repeat.

UNLOAD scsi-device-specifier

Unload the specified SCSI device (usually a tape or other removable medium). The device is specified using one of the following formats:

<code># slot bus target [unit]</code>	To specify an explicit SCSI nexus
<code>MTAnnn:</code>	To specify a SCSI tape by its sequential nexus position
<code>channel, controller, unit</code>	Operating system tape specifier

`VDIRECTORY [path]`

This is a shorthand for the “`DIRECTORY args /LONG`” command. See the “`DIRECTORY`” command for a full description of the arguments.

F.2 Micro-console commands

A command beginning with a “.” is actually processed by the micro-console. Help for the micro-console commands is provided here as a convenience to the user. Micro-console commands consist of the initial “.” followed by one or more letters and a variable number of numeric arguments. All numeric arguments are in octal. The following is a summary of the available commands:

Command	Description
<code>.B offset</code>	Start TDBoot
<code>.C</code>	Continue program
<code>.D l [adr] data</code>	Deposit things
<code>.E [l [adr]]</code>	Examine things
<code>.H</code>	Halt program
<code>.I</code>	Initialize microcode
<code>.M</code>	Enable macro-console
<code>.R cnt cmd</code>	Repeat command
<code>.S adr</code>	Start program
<code>.T l [args]</code>	Test things
<code>.U</code>	Disable macro-console
<code>.V</code>	Type version
<code>.W data</code>	Write control flags
<code>.B</code>	Start TDBoot as a normal program. The TDBoot code runs out of ROM and parses its own commands. When running in this mode, micro-console commands are not available. To get to the micro-console, type control-backslash (<code>^\</code>). Terminate this mode of TDBoot by typing “EXIT” or control-Z (<code>^Z</code>).
<code>.C</code>	Continue the current program if the PC is valid. After this command, the program will run until it executes a HALT instruction or is halted by a console command.
<code>.D</code>	Deposit things in various parts of the machine. The next letter specifies where to deposit. The last letter used is remembered and is used as the default thing to examine in the

next “E” command. The following is a summary of the deposit commands:

Command	Description
.D A adr data	Deposit MemA
.D D baw data	Deposit device-control
.D F data	Deposit flags and context
.D I data	Deposit PIR
.D M baw data	Deposit physical memory
.D N adr data	Deposit NVram
.D O om adr data	Diag write, om -> cache mode; adr -> E; data -> IABUS
.D P data	Deposit PC
.D R reg data	Deposit current-context register
.D S adr data	Deposit system virtual memory
.D U adr data	Deposit user virtual memory

.D A **address data**

Deposit into MemA. The first argument is the 13-bit address (0-17777). The second argument is the 36-bit data to deposit.

.D D **baw data**

Perform a device-control cycle on the system bus (this is also known as a write to I/O space). The first argument is the bus address word (BAW). The second word is the data to write. Bit 0 of the address controls cache access. It should never be set when accessing I/O space.

.D F **data**

Store into the internal processor flags and context word. The argument is a 36-bit number specifying the processor flags in bits 0-12, the current AC block in bits 18-20, the previous AC block in bits 21-23, and the previous section in bits 24-35.

.D I **data**

Store into the internal priority-interrupt register (as read by RDPI). Changing these values may cause certain inconsistencies in the internal machine state; use with great care.

.D M **baw data**

Perform a word write on the system bus (this is also know as a write to memory space). The first argument is the bus address word (BAW). The second word is the data to write. Bit 0 of the address controls cache access. If bit 0 is clear, the cache will be bypassed. If bit 0 is set, the cache will be flushed and/or loaded as needed and the data written into the proper cache line. Care should be taken to use the same cache access as is currently being used for the specified address; otherwise cache inconsistencies may occur.

.D N **address data**

Write one byte into the nonvolatile RAM (NVRAM). The first argument is a 13-bit address (0-17777). The second argument is the 8-bit data (0-377) to be written. Use care when changing NVRAM, because it controls various pieces of

system configuration.

.D O mode address data

Perform a diagnostic write cycle. The first argument is loaded into the cache mode. The second argument is loaded into the internal DPM address register. The third argument is loaded into the IABUS latches. A DPM write cycle is performed.

.D P address

Store into the macro PC. This is similar to the .S command, but the macro program is not continued.

.D R n data

Store into the registers selected by the current AC block context setting. The first argument is the 4-bit register number. The second argument is the 36-bit data to write.

.D S address data

Write to a system virtual address. The first argument is the 30-bit address. The second argument is the 36-bit data to write. A system virtual address is one which is mapped starting at EPT offset 540 plus the super-section number.

.D U address data

Write to a user virtual address. The first argument is the 30-bit address. The second argument is the 36-bit data to write. A user virtual address is one which is mapped starting at UPT offset 540 plus the super-section number.

.E Examine things in various parts of the machine. The next letter specifies what to examine. The following is a summary of the examine commands:

Command	Description
.E	Repeats previous examine with next address or examines the same address as the previous deposit, whichever occurred most recently.
.E A adr	Examine MemA
.E D baw	Examine device-control
.E F	Examine flags and context
.E I	Examine PIR and HW interrupt register
.E L adr	Examine lookup ROM (<0:12>, <31:35>, or <18:35>)
.E M baw	Examine physical memory
.E N adr	Examine NVram
.E O A om adr	Diag read, om -> cache mode; adr -> E; print IABUS
.E O D om adr	Diag read, om -> cache mode; adr -> E; print DPM
.E O N om adr	Diag read, om -> cache mode; adr -> E; no print
.E P	Examine PC
.E R reg	Examine current-context register
.E S adr	Examine system virtual memory
.E U adr	Examine user virtual memory

.E A address

Examine locations in MemA. The argument is the 13-bit address (0-17777).

.E D baw

Perform a status request cycle on the system bus (also know as an I/O space read). The argument is the bus address word (BAW) to read. Bit 0 controls cache access. Bit 0 should not be set with this command.

.E F

Examine the internal processor flags and context. The processor flags are in bits 0-12, the current AC block is in bits 18-20, the previous AC block is in bits 21-23, and the previous section number is in bits 24-35.

.E I

Examine the priority-interrupt register and the hardware interrupt register. The priority-interrupt register contains the same data as read with RDPI. The hardware interrupt register has hardware bits available to the microcode.

.E L address

Examine the internal lookup ROM used for byte instructions. The argument specifies the address in the ROM to examine. First, if bits 0-17 are zero, copy bits 18-30 to bits 0-12. The address is then formed from bits 0-12 and bits 31-35.

.E M baw

Perform a word-read cycle on the system bus (also know as a memory-space read). The argument is the bus address word (BAW) to read. Bit 0 controls cache access. If bit 0 is clear, the cache is bypassed. If bit 0 is set, the cache will be examined to see if the specified address is cached there. If not, the corresponding cache line will be loaded from system memory into the cache and the desired will be word returned.

.E N address

Examine a byte of nonvolatile RAM (NVRAM). The argument is the 13-bit address of the desired byte.

.E O

Perform a diagnostic-read. The first argument is a letter specifying what to type out upon completion. The second argument is loaded into the cache mode. The third argument is loaded into the internal DPM address register. The following is a summary of the diagnostic-read commands:

Command	Description
.E O A om adr	Diag read, om -> cache mode, adr -> E, print IABUS
.E O D om adr	Diag read, om -> cache mode, adr -> E, print DPM
.E O N om adr	Diag read, om -> cache mode, adr -> E, no print

.E O A mode address

Perform a diagnostic-read cycle and print the resulting contents of IABUS. The first argument is loaded into the cache mode; the second argument is loaded into the internal DPM address register.

.E O D mode address

Perform a diagnostic-read cycle and print the resulting contents of DPM. The first argument is loaded into the cache mode; the second argument is loaded into the internal DPM address register.

.E O N mode address

Perform a diagnostic-read with no output. The first argument is loaded into the cache mode; the second argument is loaded into the internal DPM address register. diagnostic-read with no output is sometimes useful for scoping a processor problem.

.E P

Print the current macro PC.

.E R n

Examine the registers selected by the current AC block context setting. The argument is the 4-bit register number (0-17).

.E S address

Read a system virtual address. The first argument is the 30-bit address. A system virtual address is one which is mapped starting at EPT offset 540 plus the super-section number.

.E U address

Read a user virtual address. The first argument is the 30-bit address. A user virtual address is one which is mapped starting at UPT offset 540 plus the super-section number.

.H Halt the currently running program. The PC remains valid, and the program can be continued where it left off using the `.C` command.

.I Restart the CPU microcode at the beginning. The CPU cannot tell the difference between this and starting from power-on.

.M Initialize the macro-console by starting it at its entry-vector offset 3. Whenever the micro-console passes a command to the macro-console, it clears the macro-console enable bits stored in MemA. If the macro-console terminates without setting this bit again, it is effectively disabled (i.e., the micro-console will not pass commands to it). The `.M` command gives the macro-console a chance to re-enable itself. (The macro-console is also disabled by the `.U` command or by installing option jumper J0.)

.R n command

Repeat any other micro-console command a specified number of times. The argument following `.R` is a number specifying the number of times to repeat the command. A value of zero will cause command to repeat indefinitely. Following the repeat count, you should give the command you wish repeated, but without the initial dot (“.”). For example, the

following commands examine all 16 (decimal) current-context registers:

```
.E R 0
.R 17 E
```

The first command examines register 0, while the second command examines the next 17 (octal) registers in sequence.

.S address

Start the processor executing instructions. The argument is a 30-bit address specifying the initial PC.

.T Perform various low-level tests which are difficult or impossible to perform using macro-code, or when macro-code execution is not operational. The following test commands are available:

Command	Description
.T A d	W/R alternate MemA locs with “d” and “d” with its low order bit complemented, respectively.
.T B	Read BP power status and ROMs
.T C b	Test cache interaction with memory at BAW “b”
.T M b s p c	Test system memory: start at BAW “b”, for “s” words, with pattern “p”. “c” has control bits.
.T S d s m	Rotate “d” by “s”, AND with “m”
.T U b	Test cache invalidate/flush functions
.T X h1 l1 h2 l2	Do 72-bit add of h1,l1 and h2,l2

.T A data

Test MemA. The argument is a data pattern. Two locations in MemA are used. The data is written to the first location. Then the data is written to the second location with the low-order bit complemented. This changes the parity bit. Both locations are then read to verify parity. This process repeats until a character is entered on the console.

.T B

Read the backplane power status, serial number and system ID ROMs. Then read the CPU serial number ROM into predefined locations in MemA. Print the power status (it appears in the hi-order 8 bits of value printed). The bits are interpreted

as follows:

Bit	Mask	Description
0	400000000000	AC fault
1	200000000000	Thermal fault
2	100000000000	Battery bad
3	040000000000	Battery low
4	020000000000	Need power asserted on backplane
5	010000000000	Not used, should be zero
6	004000000000	Backplane wedged
7	002000000000	Always set

.T C baw

Perform a variety of cache/memory interaction tests. The argument is the base bus address word (BAW) of the memory locations to use. The addresses used are BAW through BAW+37777777. Since the page offsets are fixed for mapped addresses, the BAW should have bits <27:35> set to zero. If a test fails, the address, actual, and expected data are printed and the test is terminated. The test can also be terminated by typing on the console.

.T M baw count pattern control

Perform a variety of system memory tests. The first argument is the base bus address word (BAW) of the locations to be tested. The second argument is the count of locations to test (0 will count for a long time). The third argument is a data pattern. The fourth argument is the function, encoded as follows:

Bits	Mask	Function
33:35	0	Use the data pattern unchanged
	1	Complement the pattern on each pass
	2	Increment the pattern on each pass
	3	Rotate the pattern left one bit on each pass
18	400000	Continue on error
15:17	0000000	Init pass only (places pattern in memory)
	1000000	Init only, pattern XOR'd with BAW
	2000000	Write/read/compare using pattern
	3000000	Write/read/compare, pattern XOR'd with BAW
	4000000	Write only (no read or compare)
	5000000	Read only (no write or compare)
	6000000	Read/write (no compare)
7000000	XOR/read/write (no compare)	

When comparing and stopping on errors (bit 18 clear), an error will terminate testing with a printout of the address, actual data, and expected data. The test can also be terminated at the end of a pass by typing on the console.

.T S data rotate mask

Test the internal shifter/masker hardware. The first argument is the data. The

second argument is the shift amount. The third argument is the mask.

.T U baw

Perform the invalidate and flush tests on the CPU cache. The argument is the base bus address word (BAW) of the memory locations to use. The addresses used are BAW through BAW+377777. Since this test uses the right half of the BAW as a counter, it should have bits <18:35> set to zero. If a test fails, the address, actual, and expected data are printed and the test is terminated. The test can also be terminated by typing on the console.

.T X h1 l1 h2 l2

Test the internal 72-bit adder. The first argument is the high-order word loaded into the B register. The second argument is the low-order word loaded into the B register. The third argument is the high-order word loaded into the A register. The fourth argument is the low-order word loaded into the A register. The two registers are added and the two word result is printed, high-order word first.

.U Disable the macro-console. This may be useful to prevent inadvertent running of macro-code by entering a command without an initial dot (“.”). Use the **.M** command to re-enable the macro-console. (The macro-console is also disabled by installing option jumper J0.)

.V Print the processor description string and microcode version number.

.W data Perform a WCTRLF instruction using the argument. This is provided to allow manipulation of the auxiliary port when the macro-console may not be running. The “data” argument contains the WCTRLF bits. The bits which control the auxiliary port are:

Name	Value	Function
CF%SET	1B0	Set selected bits
CF%CLR	1B1	Clear selected bits
CF%DTR	200	Auxiliary Data Terminal Ready
CF%APE	10	Auxiliary Port Enable
CF%RTS	2	Auxiliary Request To Send

To enable the auxiliary port, type “.W400000000212”; to disable the auxiliary port, type “.W200000000212”.

Index

- Accounts, 30, 81, 83
- ACJ program, 91
- ACTGEN program, 81
- Argument, viii
- Auto-Boot Structure, *see* Boot Structure
- Auxiliary Console, 9, 32

- Backup policy, 58
- Battery installation, 6
- Boot Structure, 18, 24, 40, 48, 76, 88, 103, 104, 109
- Bootstrapping, 17
- BS:, 76

- Cables, external tape drive, 11
- Cables, SCSI, 11
- Capabilities, *see* Privileges
- Channel Number, 42
- CHECKD program, 18, 41, 103
- CHECKD program, 33
- Circuit breakers, 8
- Command Recognition, vii
- Configuration, network, 28
- Configuration, system, 25, 75
- Connected Directory, 30
- Connector, console, 9
- Connector, RS-232, 10
- Connectors, Ethernet, 14
- Connectors, SCSI, 10
- Console Commands, 22
- Console ports, 32
- Console Terminal, 91
- Contributed Program, 44
- Cooling, 2

- Data Back Up, 45
- Data Recovery, 46
- DBEDIT program, 31
- Directories, 30, 32, 43
- Directory, Login, 30
- Disconnecting Means, 1

- Disk File Structures, 41
- Disks, 10
- Disks, formatting, 57

- Emergency Power Off, 1, 9
- Ethernet standards, 14

- File Access
 - Group, 43
 - Other, 43
 - Owner, 43
- File Back Up, 45
- File protection, 43
- File Structures, 41
- FINGER Program, 31

- GALAXY system, 113
- GALGEN program, 120
- Group File Access, 43
- Groups, 44
- Guide Words, viii

- Indicators, 14
- Installation, 5

- Labeled Tape, 85
- Labeled Tapes, 27
- Lamps, diagnostic, 14
- Login Directory, 30, 88
- Login Procedure, 30
- Login Structure, 88, 103
- lpd Protocol, 38
- LPTSPL program, 37

- Mail, and user-names, 32
- Maintenance Privilege, 31
- Modem, connection, 9, 33
- MONITR.EXE file, 40

- Network addresses, 28
- Network connection, 14
- Network connection, indicators, 16

- Operator Privilege, 31
- Operator, system, 97
- OPR program, 97
- Other File Access, 43
- Owner File Access, 43

- Parameter, viii
- Password, 33
- Password, operator, 22, 55
- Password, user, 30, 33
- Passwords, 89
- Power, AC, 1
- Power, emergency cutoff, 1
- Printer, UNIX, 38
- Printers, network, 3, 35
- Printers, shared, 3, 35
- Privileges, 31, 80
- PS:, 76
- Public Structure, 76, 88

- Rack mounting, 6
- Recognition, Command, vii

- SCSI cables, 11
- SCSI connectors, 10
- SCSI standard, 13
- SCSI Target IDs, 13
- Security, 33
- SETSPD program, 75
- Stand-Alone Operation, 40
- Structure Manipulation, *see* CheckD program
- Structure, File, 41
- Swapping Structure, 25, 76
- SYSDPY program, 77
- System Structure, 25, 76, 87, *see also* *Boot Structure, 91*

- Tape Drive Allocation, 85*
- Tape Labels, 27, 85*
- Tape, 9-track, 59*
- TCPSPL program, 38*
- TDBOOT program, 22*
- Terminal, Console, 9*
- Terminal, network, 3*

- User, 30*
- User IDs, 30*
- User-Name, 30, 88*

- Wheel Privilege, 31*