

UNICOS[®] Installation Guide for
CRAY Y-MP[®] EL Computer Systems

SG-5201 7.0

Cray Research, Inc.

UNICOS® Installation Guide for
CRAY Y-MP® EL Computer Systems

SG-5201 7.0

Cray Research, Inc.

Copyright © 1993 Cray Research, Inc. All Rights Reserved. This manual or parts thereof may not be reproduced in any form unless permitted by contract or by written permission of Cray Research, Inc.

Autotasking, CF77, CRAY, Cray Ada, CRAY Y-MP, CRAY-1, HSX, MPGS, SSD, SUPERSERVER, UniChem, UNICOS, and X-MP EA are federally registered trademarks and CCL, CFT, CFT2, CFT77, COS, CRAY APP, Cray C++ Compiling System, CRAY S-MP, CRAY X-MP, CRAY XMS, CRAY-2, Cray/REELlibrarian, CRInform, CR1/TurboKiva, CSIM, CVT, Delivering the power . . . , Docview, EMDS, IOS, OLNETH, RQS, SEGLDR, SMARTE, SUPERCLUSTER, SUPERLINK, and Trusted UNICOS are trademarks of Cray Research, Inc.

AMPEX is a trademark of Ampex Corporation. Ethernet is a trademark of Xerox Corporation. EXABYTE is a trademark of EXABYTE Corporation. Heurikon is a trademark of Heurikon Corporation. HYPERchannel is a trademark of Network Systems Corporation. Sabre is a trademark of Seagate Technology, Inc. Sequent is a trademark of Sequent Computer Systems, Inc. STK is a trademark of Storage Technology Corporation. UNIX is a trademark of UNIX System Laboratories. Viper and Anaconda are trademarks of Archive Technology, Inc. VT is a trademark of Digital Equipment Corporation. WYSE is a trademark of Wyse Technology, Inc.

The UNICOS operating system is derived from the UNIX System Laboratories, Inc. UNIX System V operating system. UNICOS is also based in part on the Fourth Berkeley Software Distribution (BSD) under license from The Regents of the University of California.

Requests for copies of Cray Research, Inc. publications should be sent to the following address:

Cray Research, Inc.
Distribution Center
2360 Pilot Knob Road
Mendota Heights, MN 55120
USA

Order desk (612) 681-5907
Fax number (612) 681-5920

Cray Research Software Documentation Map

The illustration on the following pages highlights the major body of documentation available for Cray Research (CRI) customers. The illustration is organized into categories by audience designation:

<u>Audience</u>	<u>Description</u>
General users	Those who use the UNICOS operating system, products, applications, or linking software
Application and system programmers	Those who write or modify system or application code on a CRI system for the purpose of solving computer system, scientific, or engineering problems
System administrators	Those who perform system administration tasks, such as installation, configuration, and basic troubleshooting
System analysts	Those who perform advanced troubleshooting, tuning, and customization
Operators	Those who perform operational functions, such as performing system dumps, and those who administer an operator workstation

To use the map, find the audience designation closest to your specific needs or role as a CRI system user. Note that manuals under other audiences may also be of interest to you; manuals are listed only once, underneath the audience to which they most directly apply. Some manual titles are abbreviated. The date in the footer tells you when the information was last revised.

For more information

In addition to the illustration, you can use the following publications to find documentation specific to your needs:

- *Software Documentation Ready Reference*, publication SQ-2122, serves as a general index to the CRI documentation set. The booklet lists documents and man pages according to topic.
- *Software Overview for Users*, publication SG-2052, introduces the UNICOS operating system, its features, and its related products. It directs you to documentation containing user-level information.
- *User Publications Catalog*, publication CP-0099, briefly describes all CRI manuals available to you, including some not shown on the map, such as training workbooks.

Ordering

To obtain CRI publications, order them by publication number from the Distribution Center:

Cray Research, Inc.
Distribution Center
2360 Pilot Knob Road
Mendota Heights, MN 55120
USA

Order desk (612) 681-5907
Fax number (612) 681-5920

GENERAL USERS

Introductory

UNICOS Primer
(SG-2010)*
Software Overview
(SG-2052)*

General

Software Documentation
Ready Reference*
(SQ-2122)
UNICOS Message
Reference (SR-2200)
User Commands
Reference (SR-2011)†

User Commands Ready
Reference (SQ-2056)

UNICOS Shells Ready
Reference (SQ-2116)

UNICOS Environment
Variables Ready
Reference (SQ-2117)

UNICOS Index for Man
Pages (SR-2049)

General (continued)

Docview Guide
(SG-2109)*
Visual Interfaces Guide
(SG-3094)*
Tape Subsystem Guide
(SG-2051)*
TCP/IP and OSI Network
Guide (SG-2009)*
NQS Guide (SG-2105)*
Security (MLS) Guide
(SG-2111)
Kerberos User's Guide
(SG-2409)

Text Editing

Text Editors Primer
(SG-2050)
v.i Reference Card
(SQ-2054)
ed Reference Card
(SQ-2055)

MVS Link

SUPERLINK General
Information Manual
(SI-0194)
SUPERLINK Messages
(SI-0176)
SUPERLINK User's
Guide (SI-0195)
MVS Station Messages
(SI-0108)
Station Reference
(SI-2066)
Station Ready Reference
(SI-0104)

RQS User's Guide
(SG-2405)

NOS/VE Link

NOS/VE Reference
(SC-0270)

UNIX Link

RQS User's Guide
(SG-2119)
CLS-UX User's Guide
(SU-3121)

VAX/VMS Link

SUPERLINK User's Guide
(SV-3153)
RQS User's Guide
(SV-3151)
Station Primer (SV-0361)
Station Reference
(SV-0020)
Station Ready Reference
(SV-0102)

VM Link

RQS VM User's
Guide (SI-0170)
Station Primer (SI-0167)
Station Reference
(SI-0168)
Station Messages and
Codes (SI-0165)
Station Reference
Summary (SI-0169)

* Available online with Docview

† Man pages available with the **man** command

APPLICATION AND SYSTEM PROGRAMMERS

C

Cray Standard C Reference (SR-2074)*

Cray Standard C Ready Reference (SQ-2076)

Ada

Cray Ada Reference (SR-3014)

Cray Ada Programming Guide (SR-3082)

Pascal

Pascal Reference (SR-0060)*

Fortran 77

CF77 Fortran Reference (SR-3071)*

CF77 Compiler Message Manual (SR-3072)

CF77 Vectorization Guide (SG-3073)*

CF77 Parallel Processing Guide (SG-3074)*

CF77 Ready Reference (SQ-3070)

UNICOS Libraries

System Calls (SR-2012)†

Fortran Library (SR-2079)†

C Library (SR-2080)†

Math & Scientific Library (SR-2081)†

Specialized Libraries (SR-2057)†

I/O User's Guide (SG-3075)*

Advanced I/O Guide (SG-3076)*

Loaders

Loader Reference (SR-0066)*

SEGLDR Reference Card (SQ-0303)

Performance Utilities

Performance Utilities Reference (SR-2040)*

Debuggers

CDBX Debugger Reference (SR-2091)*

CDBX Debugger User's Guide (SG-2094)

Cray Assembly Language (CAL)

CAL Reference (SR-2003)*

CAL for CRAY Y-MP and CRAY Y-MP C90

Reference (SR-3108)

Symbolic Machine Instructions (SR-3109)

Ready Reference (SQ-3110)

CAL for CRAY X-MP and CRAY X-MP EA

Macros and Opdefs Reference (SR-0012)

Symbolic Machine Instructions (SR-0085)

Ready Reference (SQ-0083)

CAL for CRAY-2

CAL Ready Reference (SQ-2002)

Macros and Opdefs Reference (SR-2082)†

Linking Software

SUPERLINK MVS AAC Reference (SI-0197)

VAX/VMS Station Common Access Facilities (SN-0362)

SUPERLINK Programmer's Guide VAX/VMS (SV-3155)

Source Control

USM User's Guide (SG-2097)*

Networking

RPC Reference (SR-2089)

Visualization

UNICOS X Window System Reference (SR-2101)*

Other

Support Tools Guide (SG-2016)*

UNICOS Message System Programmer's Guide (SG-2121)*

Compiler Information File (CIF) Reference (SM-2401)

SYSTEM ADMINISTRATORS

General

UNICOS Installation Guide (SG-2112)

UNICOS Installation Guide for CRAY Y-MP EL Systems (SG-5201)

UNICOS System Administration (SG-2113)*

Administrator Commands Reference (SR-2022)†

CRAY Y-MP EL-specific Administrator Commands Reference (SR-2408)†

Defining and Compiling Terminal Definitions (SN-2067)

Docview Writer's Guide (SG-2118)*

C2 Functionality on MLS Systems (SN-2407)

fy Driver Administrator's Guide (SG-2132)

IOS Models B - D

IOS Guide (SG-0307)

IOS Messages (SR-2240)

MVS Link

Station Installation (SI-0078)

SUPERLINK MVS Installation, Tuning, & Customization (SI-0188)

RQS Administrator's Guide (SG-2406)

VM Link

Station Installation and Maintenance (SI-0162)

SUPERLINK Administrator's Guide (SI-0171)

VAX/VMS Link

Station Installation (SV-0100)

Station Administration (SV-0363)

RQS Administrator's Guide (SV-3152)

SUPERLINK Installation (SG-5091)

SUPERLINK Administrator's Guide (SV-3154)

UNIX Link

RQS Administrator's Guide (SG-2120)

CLS-UX Installation and Configuration (SU-3123)

NOS/VE Link

NOS/VE Operator and Administrator Guide (SC-0271)

* Available online with Docview

† Man pages available with the **man** command

SYSTEM ANALYSTS

General

File Formats and Special Files Reference (SR-2014)[†]

Data Migration MSP Writer's Guide (SN-2098)*

UNICOS Tuning Guide (SR-2099)

System-specific Differences in the User Interface (SN-2104)

Installation and Configuration Tool Reference (SR-3090)

CSIM

User's Guide (SG-2059)

Ready Reference (SQ-2031)

IOS Models B - D

Table Descriptions (SM-0007)

Internal Reference (SM-0046)

USCP

Front-end Protocol Internals (SM-0042)*

USCP Optimization (SN-2103)

OPERATORS

Peripheral Expander

IOS Operator's Guide (SG-2005)

OWS-E

OWS-E Reference (SR-3077)[†]

OWS-E Ready Reference (SQ-3080)

OWS-E Operator's Guide (SG-3078)

OWS-E Administrator's Guide (SG-3079)

OWS

OWS Reference (SR-3030)[†]

OWS Ready Reference (SQ-3066)

OWS Operator's Guide (SG-3042)

OWS Administrator's Guide (SG-3038)

Linking Software

CLS-UX (SU-3122)

SUPERLINK MVS (SI-0196)

MVS Station (SI-0037)

* Available online with Docview

[†] Man pages available with the `man` command

Record of Revision

The date of printing or software version number is indicated in the footer. Changes are noted by revision bars along the margin of the page.

<i>Version</i>	<i>Description</i>
2.2	March 1992 – Original printing. This document describes the software installation procedures for the Cray Research, Inc. (CRI) UNICOS 6.1.5 release and the CRI entry-level (EL) systems IOS 9.1 release on Cray Research EL computer systems.
2.3	June 1992 – This document has been reorganized and rewritten to describe the software installation procedures for the CRAY Y-MP EL system software 2.3 release. The 2.3 release consists of the Cray Research UNICOS 6.1.6 release and the Cray Research entry-level (EL) systems IOS 9.3 release software.
7.0	February 1993 – This document describes the software installation procedures for the UNICOS 7.0 release and the IOS 10.1 release on CRAY Y-MP EL computer systems.

This publication is for system administrators and operators of CRAY Y-MP EL systems. It provides the procedures that are necessary to accomplish the following:

- Boot the preinstalled CRAY Y-MP EL system software
- Upgrade to a new release of the CRAY Y-MP EL system software
- Reinstall the CRAY Y-MP EL system software

This guide also provides information on customizing your configuration and performing basic troubleshooting procedures.

Related publications

Because CRAY Y-MP EL systems run the UNICOS operating system, you will find the entire UNICOS manual set to be helpful. However, because some differences exist that are especially important to system administrators and operators, the following Cray Research, Inc. (CRI) publications help you run UNICOS successfully on CRAY Y-MP EL systems:

- *UNICOS Release Overview for CRAY Y-MP EL Systems*, publication RO-5200
- *CRAY Y-MP EL System-specific Administrator Commands Reference Manual*, publication SG-2408
- *CRAY Y-MP EL Basic Administration Guide*, publication SG-2416
- *A Brief Overview of Your Cray Research Entry-level (EL) Computer System*, publication SG-2410

- *CRAY Y-MP EL Computer Systems Online Diagnostic Maintenance Manual*, publication SPM-1025[†]

Conventions

This manual uses the following typographical conventions:

<u>Convention</u>	<u>Meaning</u>
typewriter font	Denotes literal items such as command names, file names, routines, directory names, path names, signals, messages, and programming language structures.
<i>italic font</i>	Denotes variable entries and words or concepts being defined.
bold typewriter font	In screen drawings of interactive sessions, denotes literal items entered by the user. Output is shown in nonbold typewriter font.
[]	Brackets enclose optional parts of a command line.
< >	Angle brackets indicate that you must use at least one of the enclosed options.

In this publication, *Cray Research*, *CRI*, and *Cray* refer to Cray Research, Inc. and/or its products. To avoid redundancy or awkwardness, *Cray Research* may occasionally be shortened to *Cray* (for example, *Cray disk* or *Cray job*).

[†] CRAY RESEARCH PROPRIETARY. Dissemination of this document to non-CRI personnel requires approval of the appropriate vice president and that the recipient sign a nondisclosure agreement. Export of technical information in this category may require an export license.

Ordering publications

The *User Publications Catalog*, publication CP-0099, lists all Cray Research hardware and software manuals that are available to users.

To order a manual, either call the Distribution Center in Mendota Heights, Minnesota, at (612) 681-5907 or send a facsimile of your request to fax number (612) 681-5920. Cray Research employees may choose to send electronic mail to `order.desk` (UNIX system users) or `order.desk` (HPDesk users).

Reader comments

If you have comments about the technical accuracy, content, or organization of this manual, please tell us. You can contact us in any of the following ways:

- Send us electronic mail from a UNICOS or UNIX system, using the following UUCP address:

`uunet!cray!publications`

- Send us electronic mail from any system connected to Internet, using the following Internet addresses:

`pubs5201@timbuk.cray.com` (comments specific to this manual)

`publications@timbuk.cray.com` (general comments)

- Contact your Cray Research representative and ask that a Software Problem Report (SPR) be filed. Use PUBLICATIONS for the group name, PUBS for the command, and NO-LICENSE for the release name.
- Call our Software Information Services department in Eagan, Minnesota, through the North American Support Center, using either of the following numbers:

(800) 950-2729 (toll free from the United States and Canada)

(612) 683-5600

- Send a facsimile of your comments to the attention of “Software Information Services” in Eagan, Minnesota, at fax number (612) 683-5599.
- Use the postage-paid Reader’s Comment form at the back of this manual.

We value your comments and will respond to them promptly.

Contents

<i>Page</i>		<i>Page</i>	
v	Preface	21	sn.h configuration file
v	Related publications	21	conf.c configuration file
vi	Conventions	23	config.mh configuration file
vii	Ordering publications	36	config.h configuration file
vii	Reader comments	42	Customizing root (/) and /usr file systems
1	Overview [1]	42	Changing to multiuser mode
2	Organization	43	Adding tape devices
3	Before you begin	43	Boot procedures completed
5	Booting Preinstalled UNICOS [2]	43	Installing Fortran and C compilers
5	Loading the IOS kernel	43	Configuring MLS
7	Setting the IOS clock	45	Upgrade Installation [3]
7	Booting UNICOS	45	Preparing to upgrade
8	Configuring disk space	46	Saving your 6.1 configuration files
8	Configuration specification language (CSL) overview	47	Upgrading to a new release level
9	CSL recommendations	55	Network Queueing System (NQS)
9	Balancing I/O	55	Network file system (NFS) UID maps
10	Defining the root (/), /usr, and /usr/src file systems	56	Upgrade procedures completed
10	Configuring the swap device	57	Reinstalling UNICOS [4]
11	Configuring stripe groups	57	Tape and tape device
11	Using a buffer memory resident (BMR) device	59	Create the IOS file system
12	Disk types	60	Install the IOS software
12	Sample CSL settings	61	Install the scan files
18	Modify the parameter file config.uni	61	Copy site-dependent IOS files
19	Relink the UNICOS kernel	61	IOS /config file
19	Relinking procedures	64	IOS /sys/unicos.ymp file
20	Example configuration files	65	IOS /bin/boot script
		65	Automatic initial installation procedure
		68	Edit the boot script

- 68 Set the IOS clock
- 69 Reboot the UNICOS system
- 69 Relink the UNICOS kernel
- 70 Configure the peripherals
 - 70 Configuration specification language (CSL) overview
 - 71 CSL recommendations
 - 72 Balancing I/O
 - 72 Defining the root (/), /usr, and /usr/src file systems
 - 72 Configure the swap device
 - 72 Configure the stripe groups
 - 73 Using a buffer memory resident (BMR) device
 - 74 Disk types
 - 74 Sample CSL settings
- 75 **Online Tape System [A]**
 - 76 Configuring your online tape system
 - 76 Components of character special files
 - 78 Tape special file naming convention
 - 79 Tape special file mknod fields
 - 80 Sample character special files
- 83 **Panics and File System Crashes [B]**
 - 83 Types of panics
 - 84 IOS assertion panic
 - 84 IOS dumps
 - 87 UNICOS dumps
 - 87 Using the `mf dump` command on CRAY Y-MP EL systems
 - 91 Recovery from a root (/) file system crash
 - 92 Alternative root (/) file system recovery procedure
- 93 **Console Terminal Settings [C]**
 - 93 Defining your console settings
- 95 **DAS Fault-resilient Features [D]**
 - 95 DAS initialization
 - 96 Viewing threshold and error counts
 - 97 Changing threshold values
 - 97 Bad sector remapping
 - 97 DAS hot-spare switching
- 99 **UNICOS Multilevel Security [E]**
 - 99 Configuring MLS on your pre-installed UNICOS 7.0 system
 - 111 Upgrading your MLS system to 7.0
- 119 **Configuring Network Interfaces [F]**
 - 119 UNICOS kernel configuration
 - 119 The `comm_info` structure
 - 121 Determining the interface name
- 123 **UNICOS Configuration Menu System [G]**
 - 123 Menu screens
 - 125 Terminology
 - 126 Menu selections
 - 127 Menu prompts
 - 128 General menu keys
 - 129 Movement keys
 - 130 Input keys
 - 131 Accelerator keys
 - 131 Form-list keys
 - 132 Record-line keys
 - 133 Log file
 - 134 Entering numbers
 - 134 Signal processing
 - 135 Top three menu levels

Page**Figures**

- 3 Figure 1. CRAY Y-MP EL system front-panel configuration
- 48 Figure 2. Physical definition of an example `config.uni` file before modifications
- 48 Figure 3. Physical definition of an example `config.uni` file after modifications
- 49 Figure 4. Logical definition of an example `config.uni` file before modifications
- 49 Figure 5. Logical definition of an example `config.uni` file after modifications
- 58 Figure 6. Anaconda QIC tape device
- 64 Figure 7. IOS Heurikon boards
- 77 Figure 8. Special file name and `mknod` components
- 121 Figure 9. The `comm_info` structure
- 122 Figure 10. Examples of interface entry numbers
- 135 Figure 11. Top three levels of the menu system

Tables

- 10 Table 1. Disk space on `/usr/src` required for UNICOS and other products
- 12 Table 2. CRAY Y-MP EL system disk types
- 47 Table 3. Disk space on `/usr/src` required for UNICOS and other products
- 74 Table 4. CRAY Y-MP EL system disk types
- 81 Table 5. Example special character file command lines
- 130 Table 6. Input keys
- 134 Table 7. Numeric types in the menu system

This document is for system administrators and operators of CRAY Y-MP EL computer systems. It explains the procedures for booting the preinstalled CRAY Y-MP EL system software 7.0 release. This guide also provides information about how to upgrade to a new revision level of the CRAY Y-MP EL system software and how to reinstall the software from tape.

There are several noteworthy changes in release 7.0 for CRAY Y-MP EL systems. A new package format includes absolute binary files instead of the relocatable files that were shipped in previous releases. The following are the effects of this change:

- Absolute binaries are loaded from the package directly into the root (/) and /usr file systems instead of relocatable binaries being loaded into /usr/src to be linked with segldr and cpset into their destinations. The exception to this is the UNICOS kernel build environment, which is still delivered into /usr/src/uts.
- The time required to install a UNICOS package has been substantially reduced because no builds are necessary except for the UNICOS kernel.
- The amount of space required for /usr/src has been substantially reduced. The recommended default is now 50,000 blocks.
- The minimum space required for the root (/) and back-up root (/bkroot) file systems is 90,000 blocks.
- The minimum space required for the /usr and back-up /usr (/bkusr) file systems is 150,000 blocks.
- The UNICOS Installation and Configuration menu system (the menu system) has limited support in this release. You can use the menu system to install MLS, but you cannot use it to install UNICOS on CRAY Y-MP EL systems. See "UNICOS Configuration Menu System," page 123, for information about using the menu system.

A version of the CRAY Y-MP EL system software is installed and configured on your CRAY Y-MP EL system. The software consists of the CRAY Y-MP EL I/O subsystem (IOS) and the Cray Research UNICOS operating system.

Before you begin any of the procedures outlined in this guide, you should read through the section that relates to the procedure you intend to perform. For example, if you intend to use the preinstalled software, you should read “Booting Preinstalled UNICOS,” page 5, in its entirety before beginning the procedures.

You should have available the following documentation:

- *CRAY Y-MP EL Basic Administration Guide*, publication SG-2416
- *UNICOS User Commands Reference Manual*, publication SR-2011
- *CRAY Y-MP EL System-specific Administration Commands Reference Manual*, publication SR-2408
- *UNICOS Administrator Commands Reference Manual*, publication SR-2022.

Organization

1.1

Unless you are upgrading to a new release level or intend to reinstall the CRAY Y-MP EL system software (UNICOS), you should follow the procedures in “Booting Preinstalled UNICOS,” page 5, for booting the preinstalled UNICOS.

“Upgrade Installation,” page 45, is for sites currently running UNICOS which plan to upgrade to a new release level.

“Reinstalling UNICOS,” page 57, explains how to perform an initial installation of UNICOS from tape.

Appendixes A through G provide information on the following:

- Online tape system
- Panics and file system crashes
- Console terminal settings
- Disk array subsystem (DAS)
- UNICOS multilevel security (MLS)
- Configuring network interfaces

- Configuring network interfaces
- UNICOS Installation and Configuration Menu System

Before you begin

1.2

After your CRAY Y-MP EL system is powered up, you must manually reset both the IOS and central processing unit (CPU) logic before booting the preinstalled UNICOS system or installing it from tape. If you are upgrading to a new revision level of UNICOS on a CRAY Y-MP EL system, you do not need to reset the IOS and CPU logic.

To reset both the IOS and CPU logic, depress the two manual reset buttons on the front panel of your CRAY Y-MP EL system. Figure 1 shows this panel.

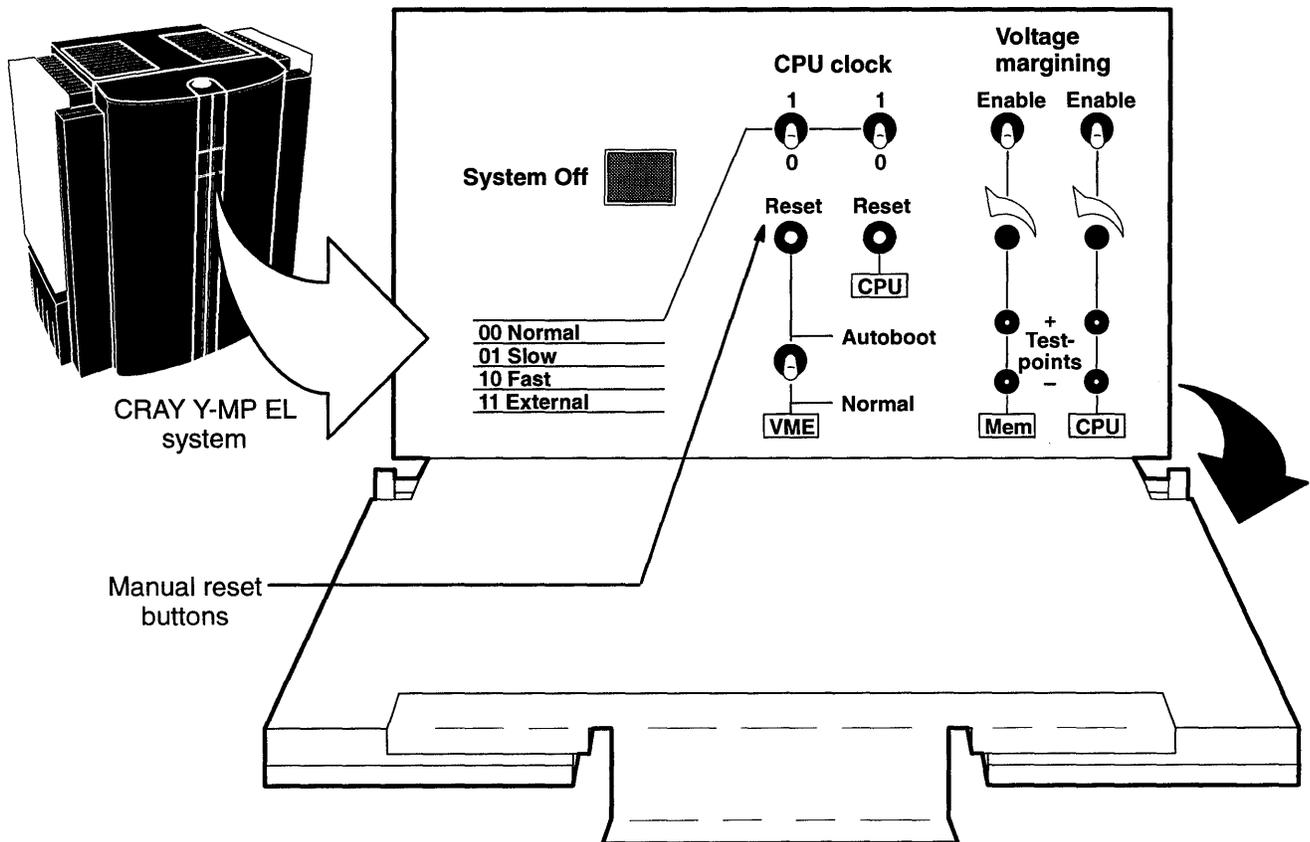


Figure 1. CRAY Y-MP EL system front-panel configuration

Section 4 of the *UNICOS Release Overview for CRAY Y-MP EL Systems*, publication RO-5200, outlines software compatibility issues and differences between previous EL releases and the UNICOS 7.0 release. Before proceeding with the upgrade to UNICOS 7.0, you should familiarize yourself with this information.

Booting Preinstalled UNICOS [2]

This section describes the procedures for booting the preinstalled CRAY Y-MP EL system software that is configured to match the hardware configuration of your CRAY Y-MP EL system.

The preinstalled UNICOS file systems let you run UNICOS with a generic file system configuration. You may decide to reconfigure the file systems to match your site requirements and disk space. The procedures to make these changes are outlined in this section.

Loading the IOS kernel

2.1

When your system is powered up, you will see the programmable read-only memory (PROM) prompt, `BOOT>`, on your system console. At this prompt, enter the following command to boot the preinstalled IOS kernel:

```
BOOT> load
```

Output similar to that shown on the following page is displayed on your system console when the IOS kernel is loaded. The length of time required for the `load` command to execute depends on the number of IOSs and devices that are configured on your CRAY Y-MP EL system. After the IOS kernel is loaded successfully, your prompt will change from `BOOT>` to `IOS>`.

BOOT> load

Loading /IOS/IOS...
Loaded 200422 bytes to location 0x3010000.
RC> call 03010020
IFX Version 1.10
RTSCOPE 68020 v1.24

No FPU Present

Copyright (c) 1991, 1992, 1993 by Entry Level Systems, Cray Research, Inc. This is confidential and proprietary software. All rights reserved.

Entry Level Systems IOS version 10.1 93/01/29 11:36:36

SCSI winchester disk: 1 drive(s) found
Master IOS: IOP Serial Number 1509
VME SCSI: Controller 0 - detected
Ethernet: Controller 0 - detected
ESDI ctlr 0 unit 0 [E00], type = DD3
ESDI ctlr 0 unit 1 [E01], type = DD3
ESDI ctlr 1 unit 0 [E10], type = DD3
ESDI ctlr 1 unit 1 [E11], type = DD3
DAS ctlr 0 bank 0 [D00], type = DDAS2
SCSI ctlr 0 unit 0 lun 0 [rss000], type = TD3
SCSI ctlr 0 unit 1 lun 0 [rss010], type = TD3S
SCSI ctlr IOP unit 1 lun 0 [rpq01], type= QT1
SCSI ctlr IOP unit 2 lun 0 [rpe02], type= EX2
PERTEC ctlr 0 unit 0 [rmt00], type = TD2
PERTEC ctlr 1 unit 0 [rmt10], type = TD2

Architecture: YMP-EL, cpu 0 configured.
This is the MASTER IOS.

IOS>

Setting the IOS clock

2.2

To verify that the IOS clock is set for your local date and time and to adjust this clock, use the `time` command, as follows:

```
IOS> time
```

The time reported and set by this command is the time that UNICOS will boot with; therefore, be sure it is correct, so you do not have to change it later for UNICOS and then have your UNICOS clock be out of sync with your IOS clock.

For more information on the `time` command, see the *CRAY Y-MP EL System-specific Administration Commands Reference Manual*, publication SR-2408.

Booting UNICOS

2.3

The UNICOS kernel (`/sys/unicos.ymp`) and configuration file (`/sys/config.uni`) reside on the IOS Small Computer System Interface (SCSI) disk device. These files are used within the boot script when booting the system to single-user mode. This method provides flexibility to change variables such as `ROOTDEV` and `PIPEDEV` from the IOS prior to booting the UNICOS system.

With the IOS kernel loaded, boot UNICOS to single-user mode by entering the following command at the `IOS>` prompt:

```
IOS> boot
```

After booting UNICOS to single-user mode, you must use the `fsck` command to check the root (`/`) file system for inconsistencies. See *UNICOS System Administration*, publication SG-2113, for information about the `fsck` command.

After successfully correcting the `/root` file system, `fsck` prints the number of files on the `/root` file system, the number of used and free blocks, and the percentage of fragmentation. To check the `/root` file system, enter the following command line at the UNICOS prompt (`#`):

```
# fsck /dev/dsk/root
```

Note

`CONTROL-a` toggles between the IOS and UNICOS consoles.

Configuring disk space

2.4

This subsection describes how to configure your peripherals by editing your configuration specification language (CSL) variables in the `/sys/config.uni` file that resides on the IOS disk.

Using CSL, you will define run-time system configurable variables, your disk peripherals, and file systems.

To familiarize you with CSL, this subsection provides an overview, which includes a description of CSL, parameter setting recommendations, and a sample configuration file. “Modify the parameter file `config.uni`,” page 18, describes the specific procedures for editing and reinstalling your new `/sys/config.uni` file. See *UNICOS System Administration*, publication SG-2113, for more information about CSL.

Note

System performance is significantly impacted by disk configuration. See *UNICOS System Administration*, publication SG-2113, for details about file system placement.

Configuration specification language (CSL) overview

2.4.1

CSL describes configuration and run-time parameter settings that are interpreted when you boot UNICOS. The use of CSL statements permits you to extend or override the default configuration by letting you do the following:

- Define the physical attributes of your disk devices
- Define the logical grouping of your physical disk slices
- Define system devices
- Configure logical device cache
- Define memory size (required)
- Specify the number of IOSs, CPUs, and channels

Your IOS `/sys/config.uni` file contains the definitions of your physical and logical system disk devices. Optionally, `sys/config.uni` can contain CSL statements to modify or override specific UNICOS kernel parameters and the additional statements required to configure UNICOS disk striping.

Each time the UNICOS operating system is booted, UNICOS interprets and sequentially processes the CSL statements; therefore, the order of your CSL statements is very important. After processing the CSL file, UNICOS automatically creates the following parameter files that reflect the current system configuration:

<u>File name</u>	<u>Description</u>
/IOS-param	This file is an exact copy of your CSL file.
/CONFIGURATION	This file contains CSL-defined definitions and compiled UNICOS kernel definitions.
/unicos	This file is a copy of the running UNICOS kernel.

CSL recommendations 2.4.2

When configuring UNICOS during the installation process, you must use CSL to define the size and location of the disk region used for the root (/), /usr, and /usr/src file systems. You must also use CSL to define the size and location of the UNICOS swap and dump devices, /bkroot (back-up root), /bkusr (back-up /usr), /tmp, and /core.

Note

Disk and buffer memory resident (BMR) partitions are defined in units of blocks (1 block = 4096 bytes) on CRAY Y-MP EL systems.

Balancing I/O 2.4.2.1

To balance your I/O, place the root (/) and /usr file systems on different disk drives. If possible, also isolate the root (/) and /usr file systems on different disk controllers. The recommended disk drive for the root (/) and /usr file systems is the DD-4 disk drive; use the DD-3 and DDESDI disk types as alternatives.

*Defining the root (/),
/usr, and /usr/src file
systems*
2.4.2.2

You must define the root (/) and back-up root (/bkroot) file systems as the same size; they require a minimum of 90,000 blocks. You must also define the /usr and back-up /usr (/bkusr) file systems as the same size; they require a minimum of 150,000 blocks.

The disk space requirements for /usr/src on CRAY Y-MP EL systems have been dramatically reduced for the UNICOS 7.0 release. The recommended default value is 50,000 blocks. This is sufficient to hold all of the files necessary to relink UNICOS kernels and to build the CF77 and SCC compilers. Table 1 shows the required disk space in /usr/src for UNICOS and other products on CRAY Y-MP EL systems:

Table 1. Disk space on /usr/src required for UNICOS and other products

Product	Size in 4 Kbyte blocks
UNICOS 7.0	25,000
CF77 5.0	11,500
SCC 3.0	9,500
Pascal 4.2	2,300
Data migration	15,400
DWB	9,100
OSI 2.0	44,700

*Configuring the swap
device*
2.4.2.3

Generally, you should configure your swap device to be 3 to 8 times larger than your UNICOS main memory. If your system includes a disk array subsystem (DAS), use it for your swap device, because the DAS is best suited for large-block sequential I/O. The DAS is also an excellent location for seldom-used file systems or application disk space.

If you have no DAS, configure your swap device on available DD-4 or DD-3 disk drives. Locate your swap device on different disk drives than those used for your user's /home directories and applications.

Configuring stripe groups 2.4.2.4

Stripe group members must be placed on the same disk type (for example, DD-4), be the same length, and start at the same block. When possible, place stripe group members on separate controllers. You can also stripe across multiple IOSs.

A striped swap configuration example of `stripe_20` and `stripe_30` on IPI disk drives follows:

```
STRIPED := DDSTR (
    stripe_20 + stripe_30 /* 150000 blocks
*/
);
swap := (                /* 150000 blocks
*/
    stripe_20
);
```

Using a buffer memory resident (BMR) device 2.4.2.5

When using a BMR device, you should adhere to the following rules:

- Striped file systems cannot contain BMR partitions.
- Banded file systems (physically adjacent file systems on a disk) can be composed of BMR partitions from more than one IOS.
- BMR and disk partitions can be mixed in file systems that are not striped.
- The swap device can be defined to include partitions of BMR from more than one IOS.
- A BMR device defined as `ldcache` can be used only to cache file systems that are wholly composed of disk partitions that reside within the same IOS as that BMR device.
- The CSL `BMRDEV` statement cannot be defined to include partitions of a BMR device from more than one IOS.

Disk types
2.4.3

Table 2 shows the disk types that can be configured on your CRAY Y-MP EL system:

Table 2. CRAY Y-MP EL system disk types

Disk type	Description	Number of user blocks per device	Sectors/track
DDESDI	Old ESDI drive	170,100	9
<i>1.36g</i> *DD-3	New ESDI drive	334,200	10
DR-1	Removable ESDI	334,200	10
<i>2.67g</i> *DD-4	IPI-Sabre 7	653,000	28
DDLDA	Old DAS	1,269,114	65
DDAS2	New DAS	2,502,000	75
DDBMR	VME RAM device	Multiple of 32,768	—
DDRAM	Central memory resident	Configuration dependent	—

Sample CSL settings
2.4.4

The sample /sys/config.uni file, beginning on page 13, contains CSL for a CRAY Y-MP EL system with two CPUs and two IOSs. This file exemplifies a configuration for the basic file systems and 10 sample file systems (fs1...fs10). Each file system name has a suffix of *_dxyz*, where each letter has the following meaning:

<u>Component</u>	<u>Meaning</u>
<i>d</i>	Device type
<i>x</i>	IOS number
<i>y</i>	Controller number
<i>z</i>	Unit number

For clarity, the sample `/sys/config.uni` file is partitioned by text that explains each configurable parameter. CSL comments start with `/*` and end with `*/`.

```
CONFIG {
    AUTOGEN OFF;

    /* ----- IOS settings ----- */

    /* The number of IOSs must be set if more than 1. CHANNEL sets */
    /* channel number for IOSs beyond IOS 0. */

    IOS=2;
    CHANNEL 040 = MIOP IOS = 1;

    /* ----- CPU settings ----- */

    /* The number of CPUs must be set if more than 1.*/

    CPUS = 2;

    /* NLDCH is the number of ldcache headers. LDCHCORE is the number */
    /* of 4-Kbyte blocks of type MEM ldcache. */

    /*      NLDCH = 1024; */
    /*      LDCHCORE = 8192; */

    /*-----*/
    /*----- IOS 0 physical devices settings -----*/
    /*-----*/

    /* IOS 0 ESDI controller (MIOP) 0 */

    ESDI_000 := DD3 MIOP 0 UNIT=0 (
        0:root_e000      /* 90000 blocks */
        96000:fs1_e000  /* 244200 blocks */
    );
```

(continued)

```

ESDI_001 := DD3 MIOP 0 UNIT=1 (
    0:fs2_e001      /* 184200 blocks */
    184200:tmp_e001/* 150000 blocks */
);
ESDI_002 := DD3 MIOP 0 UNIT=2 (
    0:fs3_e002      /* 184200 blocks */
    184200:tmp_e002/* 150000 blocks */
);
ESDI_003 := DD3 MIOP 0 UNIT=3 (
    0:fs4_e003      /* 184200 blocks */
    184200:tmp_e003/* 150000 blocks */
);

/* IOS 0 ESDI controller (MIOP) 1 */

ESDI_010 := DD3 MIOP 1 UNIT=0 (
    0:usr_e010      /* 150000 blocks */
    150000:src_e010/* 150000 blocks */
    300000:fs4_e010/* 34200 blocks */
);
ESDI_011 := DD3 MIOP 1 UNIT=1 (
    0:fs3_e011      /* 184200 blocks */
    184200:tmp_e011/* 150000 blocks */
);

ESDI_012 := DD3 MIOP 1 UNIT=2 (
    0:fs2_e012      /* 184200 blocks */
    184200:tmp_e012/* 150000 blocks */
);
ESDI_013 := DD3 MIOP 1 UNIT=3 (
    0:fs1_e013      /* 184200 blocks */
    184200:tmp_e013/* 150000 blocks */
);

/* IOS 0 DAS controller (MIOP) 8 */
/* Note: DAS MIOPs are referenced as MIOP 8 */

DAS_000 := DDAS2 MIOP 8 UNIT=0 (
    0:dump_d000     /* 75000 blocks */
    75000:swap_d000 /* 350000 blocks */
    425000:bkroot_d000 /* 90000 blocks */

```

(continued)

```
515000:bkusr_d000 /* 150000 blocks */
665000:core_d000 /* 150000 blocks */
815000:local_d000 /* 250000 blocks */
1065000:tmp_d000 /*1470000 blocks */
);

/*-----*/
/*---- IOS 1 Physical devices settings ----*/
/*-----*/

/* IOS 1 IPI controller (MIOP) 10 */
/* Note: IPI MIOPs are referenced as MIOP 10 */

IPI_100 := DD4 MIOP 10 UNIT=0 IOS=1 (
    0:fs5_i100 /* 653000 blocks */
);
IPI_101 := DD4 MIOP 10 UNIT=1 IOS=1 (
    0:fs6_i101 /* 653000 blocks */
);
IPI_102 := DD4 MIOP 10 UNIT=2 IOS=1 (
    0:fs5_i102 /* 653000 blocks */
);
IPI_103 := DD4 MIOP 10 UNIT=3 IOS=1 (
    0:fs6_i103 /* 653000 blocks */
);

/* IOS 1 ESDI controller (MIOP) 0 */

ESDI_100 := DD3 MIOP 0 UNIT=0 IOS=1 (
    0:fs7_e100 /* 334200 blocks */
);
ESDI_101 := DD3 MIOP 0 UNIT=1 IOS=1 (
    0:fs8_e101 /* 334200 blocks */
);
ESDI_102 := DD3 MIOP 0 UNIT=2 IOS=1 (
    0:fs9_e102 /* 334200 blocks */
);
ESDI_103 := DD3 MIOP 0 UNIT=3 IOS=1 (
    0:fs10_e103 /* 334200 blocks */
);

/* ----- Stripe groups ----- */
```

(continued)

```
/*-----*/
/* ----- Logical device settings -----*/
/*-----*/

dump := (
    dump_d000          /* 75000 blocks */
);
swap := (
    swap_d000         /* 163840 blocks */
);
bkroot := (
    bkroot_d000       /* 90000 blocks */
);
bkusr := (
    bkusr_d000        /* 150000 blocks */
);
core := (
    core_d000         /* 150000 blocks */
);
local := (
    local_d000        /* 250000 blocks */
);
dastmp := (
    tmp_d000          /*1470360 blocks */
);
fs5 := (
    fs5_i100          /* 1306000 blocks */
    fs5_i102
);
fs6 := (
    fs6_i103          /* 1306000 blocks */
    fs6_i101
);
root := (
    root_e000         /*90000 blocks */
);
usr := (
    usr_e010          /* 150000 blocks */
);
src := (
    src_e010          /*150000 blocks */
);
```

(continued)

```
tmp := (          /* 900000 blocks */
    tmp_e001
    tmp_e011
    tmp_e002
    tmp_e012
    tmp_e003
    tmp_e013
);
fs1 := (          /* 422400 blocks */
    fs1_e000
    fs1_e013
);
fs2 := (          /* 368400 blocks */
    fs2_e012
    fs2_e001
);
fs3 := (          /* 368400 blocks */
    fs3_e002
    fs3_e011
);
fs4 := (          /* 218400 blocks */
    fs4_e010
    fs4_e003
);
fs7 := (          /* 334200 blocks */
    fs7_e100
);
fs8 := (          /* 334200 blocks */
    fs8_e101
);
fs9 := (          /* 334200 blocks */
    fs9_e102
);
fs10 := (         /* 334200 blocks */
    fs10_e103
);
/*-----*/
/* ----- System devices ----- */
/*-----*/

    SWAPDEV = swap;
    ROOTDEV = root;
    PIPEDEV = root;
}
```

Modify the parameter file `config.uni`

2.5

If you want to change your system's disk configuration, use the following procedure:

1. Make a back-up copy of your current `config.uni` file by using the following command

```
install# CONTROL-a (toggles to the IOS)
IOS> cp /sys/config.uni /sys/config.bak
```

2. Transfer the `/sys/config.uni` file from the IOS to UNICOS by entering the following:

```
IOS> CONTROL-a RETURN (toggles back to UNICOS)
install# exdf -i sys/config.uni > new_config
```

3. Edit the `/new_config` file to match your configuration by entering the following:

```
install# vi /new_config
```

4. Check the configuration file for errors by entering the following:

```
install# bconfig new_config
```

5. If no errors are found, write the file back to the IOS SCSI disk, as follows:

Caution

The following command line overwrites the CSL file on the IOS hard disk that was used when UNICOS was previously booted.

```
install# exdf -o sys/config.uni < new_config
```

6. Reboot using the new configuration, as follows:

```
install# sync; sync; sync
CONTROL-a (toggles to the IOS)
IOS>
IOS> reload (wait for the IOS load to finish)
IOS> /bin/boot
```

7. Create the mount table, then check and mount the `/usr` and `/usr/src` file systems by entering the following commands:

```
# /etc/devnm / | /etc/setmnt
# /etc/fsck /dev/dsk/usr
# /etc/mount /dev/dsk/usr /usr
# /etc/fsck /dev/dsk/src
# /etc/mount /dev/dsk/src /usr/src
```

8. Use the `mkfs` command to check any new file systems and to recreate file systems that have changed size due to the new configuration.

Relink the UNICOS kernel

2.6

You can relink a UNICOS kernel for reasons that are specific to your site. When you relink a UNICOS kernel, you can modify the following kernel configuration files (*SN* is your system's serial number):

- `/etc/config/config.mh`
- `/usr/src/uts/cf.SN/sn.h`
- `/usr/src/uts/cf.SN/conf.c`
- `/usr/src/uts/cf.SN/config.h`

This subsection provides procedures for relinking a UNICOS kernel, lists examples of the kernel configuration files, and tells how they can be modified.

Relinking procedures

2.6.1

Use the following procedures to relink the UNICOS kernel:

1. Before starting this procedure, the UNICOS system must be in single-user mode with the root (`/`), `/usr`, and `/usr/src` file systems mounted. (The state it is in after completing "Modify the parameter file `config.uni`," page 18.)

2. Ensure that the `sn.h`, `conf.c`, and `/etc/config/config.mh` files are correct (see examples on pages 21 through 42), and relink the kernel by entering these commands:

```
# cd /usr/src/uts
# /usr/bin/nmake rmbin rmexe
# /usr/bin/nmake install
```

3. Save the old kernel and move the new kernel to the IOS disk by entering the following:

```
# cd /usr/src/uts/cf.SN
# CONTROL-a (toggles to the IOS console)
IOS> mv /sys/unicos.ymp /sys/unicos.old
CONTROL-a RETURN (toggles back to the UNICOS console)
# /bin/exdf -o sys/unicos.ymp < unicos
```

4. Reboot the system by entering the following:

```
# cd /
# /etc/umountem
# sync; sync; sync
# CONTROL-a (toggles to the IOS console)
IOS> boot (boots UNICOS to single-user mode)
```

Caution

Booting the kernel without first reloading the IOS will not work if you have run UNICOS in multiuser mode or have started any of the network interfaces.

Example configuration files

2.6.2

This subsection lists examples of the kernel configuration files, and tells how they can be modified. The serial number 5102 is used for example purposes; yours will be different. The dates, times, and version numbers shown here may also be different from the ones in the files on your system.

sn.h configuration file
2.6.2.1

The following example shows the lines in the /usr/src/uts/cf.5102/sn.h configuration file that you can safely change. You can change the serial number (SN), the number of CPUs (NCPU), and the memory size (MEMORY). (Your /usr/src/uts/cf.5102/sn.h file can begin as a copy of the /usr/src/uts/c1/sys/sn.5102.h file.)

```
#define SN          5102          /* mainframe serial number */
#define NCPU        4            /* Number of CPUs */
#define MEMORY      128*MEGAWD-1 /* memory size in words */
```

conf.c configuration file
2.6.2.2

The following example shows the part of the /usr/src/uts/cf.5102/conf.c configuration file where network connections are defined. You can modify this part of the file to define your system's network connections by changing the comm_info structure to reflect your configuration. See "Configuring Network Interfaces," page 119, for more information about this. (Your /usr/src/uts/cf.5102/conf.c configuration file can begin as a copy of the /usr/src/uts/c1/cf/conf.5102.c file.)

```
/*
 *Network configuration section.
 *
 *The network configuration section defines the external interfaces to
 *the system. Each channel is configured using the following format:
 *
 * { driver, ios, iop, channel, devtype, paths, dependent }
 *
 * driver - which driver type to use for this channel
 *   COMM_HY_DRIVER - low speed driver
 *   COMM_SL_DRIVER - SUPERLINK driver
 *   COMM_HSX_DRIVER - HSX driver (includes HPPI)
 * ios - IOS number
 * iop - IOP number
 * channel - channel number (octal)
 * devtype - device type (see netdev.h for types)
 * paths - number of logical paths
 * dependent - driver dependent parameter
 *   COMM_HY_DRIVER - TCP interface information
 *   COMM_SL_DRIVER - not used at present
 *   COMM_HSX_DRIVER - definition flags
 */
```

(continued)

```
struct comm_info comminfo[] = {
    /* drv, ios, iop, chan, devtype, lpath, devword */
    {COMM_EN_DRIVER, 0, 0, 0x10, COMM_EN, 0, 0},
    {COMM_EN_DRIVER, 0, 0, 0x11, COMM_EN, 0, 0},
    {COMM_EN_DRIVER, 0, 0, 0x12, COMM_EN, 0, 0},
    {COMM_EN_DRIVER, 0, 0, 0x13, COMM_EN, 0, 0},
    {COMM_HY_DRIVER, 1, 0, 0x0, COMM_N400, 4, 0},
    {COMM_HY_DRIVER, 1, 0, 0x1, COMM_N400, 4, 0},
    {COMM_FDDI_DRIVER, 0, 0, 0x20, COMM_FDDI, 0, 0},
    {COMM_FDDI_DRIVER, 0, 0, 0x21, COMM_FDDI, 0, 0},
    {COMM_HSX_DRIVER, 4, 0, 000, COMM_HIPPI_SLB, 16, HXCF_HIPPI},
    {COMM_HSX_DRIVER, 4, 0, 001, COMM_HIPPI_SLB, 16, HXCF_HIPPI},
    {COMM_HSX_DRIVER, 4, 0, 016, COMM_HIPPI, 16, HXCF_HIPPI},
    {COMM_HSX_DRIVER, 4, 0, 017, COMM_HIPPI, 16, HXCF_HIPPI},
};

int ncomm = (sizeof(comminfo)/sizeof(struct comm_info));
int nhsx = 0;
int nhyl = 0;
int nsf = 0;
struct hx *hxdev;
struct hy *hydev;
struct sf *sfdev;
#ifdef(CRAYEL)
struct fddi *fddidev;
int nfddi = 0;
struct en *endev;
int nen = 0;
#endif /* CRAYEL */
```

config.mh *configuration*
file
2.6.2.3

The /etc/config/config.mh configuration file shown in the following example must contain the system serial number and node name. Some selections are optional.

```

/* USMID @(#)skl/etc/config/config.mh 70.13 05/13/92 08:52:54 */

/*
 * COPYRIGHT CRAY RESEARCH, INC.
 * UNPUBLISHED -- ALL RIGHTS RESERVED UNDER
 * THE COPYRIGHT LAWS OF THE UNITED STATES.
 */
/*
 * make/config.mh - Configuration file for installation of UNICOS
 */

/*****
**/
/*****
**/
/*
 *
 *          SECTION I
 *          -----
 *
 * Mandatory variables - Must be set on site for the Y-MP EL
 *
 */

/*
 * The serial number of the system to generate.
 */
#ifndef CONFIG_SN
#define CONFIG_SN 5102
#endif

/*
 * The name of the system. See uname(1).
 */
#ifndef CONFIG_SYS
#define CONFIG_SYS unicos
#endif

```

(continued)

```

/*
 * The node name of the system. The node name may be a name that the
 * system is known by to a communications network. See uname(1).
 */
#ifndef CONFIG_NODE
#define CONFIG_NODE unicos
#endif

/*
 * UNICOS Release Number
 * The following depict the major and minor numbers of this release.
 */
#ifndef CONFIG_RLS_MAJOR
#define CONFIG_RLS_MAJOR 7
#endif
#ifndef CONFIG_RLS_MINOR
#define CONFIG_RLS_MINOR 0
#endif
#ifndef CONFIG_RLS_REVISION
#define CONFIG_RLS_REVISION 4
#endif

/*****
**/
/*****
**/

/*
 *
 * SECTION II
 * -----
 *
 * Optional variables - May be either on or off for the Y-MP EL
 *
 */

/*
 * The version name of the system. If unspecified, it defaults to the
 * value of the environment variable LOGNAME. See uname(1).
 */
#ifndef CONFIG_VERSION
#define CONFIG_VERSION $LOGNAME
#endif

```

(continued)

```
/*
 * Name of the desired temporary-file directory to use.  If the
 * TMPDIR environment variable is already defined, the following
 * default is NOT used.  Reference tmpnam(3C) for more
 * information.
 */
#ifndef CONFIG_TMPDIR
#define CONFIG_TMPDIR /tmp
#endif

/*
 * Remote Process Control System (RPC)
 * CONFIG_TCP must be configured also if using this option.
 */
#ifndef CONFIG_RPC
#define CONFIG_RPC 1
#endif

/*
 * Network Remote File System (NFS)
 * CONFIG_TCP must be configured also if using this option.
 */
#ifndef CONFIG_NFS
#define CONFIG_NFS 1
#endif

/*
 * Kerberos
 * CONFIG_TCP must be configured also if using this option.
 */
#ifndef CONFIG_KERBEROS
#define CONFIG_KERBEROS 0
#endif

/*
 * Batch Management System (NQS)
 */
#ifndef CONFIG_NQS
#define CONFIG_NQS 1
#endif
```

(continued)

```
/*
 * Network testing tools
 */
#ifndef CONFIG_NETTOOLS
#define CONFIG_NETTOOLS 0
#endif

/*
 * Open Systems Interface
 */
#ifndef CONFIG_OSI
#define CONFIG_OSI 0
#endif

/*
 * File Transfer Agent (FTA)
 */
#ifndef CONFIG_FTA
#define CONFIG_FTA 0
#endif

/*
 * System Security
 */
#ifndef CONFIG_SECURE
#define CONFIG_SECURE 0
#endif

/*
 * If set to one (1), the CONFIG_TAPE variable turns on the online tape
 * capability in the kernel. If set to zero (0), the code is turned
 * off. Its default is one (1), or ON.
 */

#ifndef CONFIG_TAPE
#define CONFIG_TAPE 1
#endif

/*
 * Cray/REELlibrarian
 */
#ifndef CONFIG_CRL
#define CONFIG_CRL 0
#endif
```

(continued)

```

/*
 * Data Management/migration
 */
#ifndef CONFIG_DM
#define CONFIG_DM 0
#endif

/*
 * File quotas
 */
#ifndef CONFIG_FQUOTAS
#define CONFIG_FQUOTAS 1
#endif

/*
 * Runtime Control Ownership Restrictions Option
 * The following CONFIG_RC_OWNED is for restricting any .*rc file to be
 * owned by the user. For example, the .exrc file for vi. If this option
 * is set to on (1), vi will execute a .exrc file only if it is owned by the
 * user. This will prevent "trojan horses" being put into the .exrc file
 * by abusive users. If this option is off (0), the behavior of the vi and
 * the .exrc usage will be System V compatible. Note that other products
 * may implement this option at a later date.
 */
#ifndef CONFIG_RC_OWNED
#define CONFIG_RC_OWNED 0
#endif

/*****
/*****
/*
 *
 *          SECTION III
 *          -----
 *
 * Static variables - Do not change these for the Y-MP EL
 *
 */

#ifndef CONFIG_ID
#define CONFIG_ID UNICOS

#ifndef CONFIG_ELS
#define CONFIG_ELS 1
#endif

```

(continued)

```
/*
 * The target of the system to generate.  If null, the existing
 * default is used.  See target(1) for more information.
 */
#ifndef CONFIG_TARGET
#define CONFIG_TARGET host
#endif

/*
 * Yellow Pages
 */
#ifndef CONFIG_YP
#define CONFIG_YP 1
#endif

/*
 * HSX device driver (uts kernel)
 * Also required by ultranet driver.
 */
#ifndef CONFIG_HSX
#define CONFIG_HSX 1
#endif

/*
 * The following definition contains the full path of the directory
 * which contains the generation software.
 */
#ifndef CONFIG_GENBIN
#if architecture(cray)
#define CONFIG_GENBIN /usr/gen/bin
#else
#define CONFIG_GENBIN /bin
#endif
#endif

/*
 * The environment variable FPP is set to the value of
 * CONFIG_FPP.  The cf77(1) command uses this variable.
 */
#ifndef CONFIG_FPP
#if architecture(cray)
#define CONFIG_FPP CONFIG_GENBIN/fpp
#else
#define CONFIG_FPP
#endif
#endif
```

(continued)

```
/*
 * The environment variable FMP is set to the value of
 * CONFIG_FMP. The cf77(1) command uses this variable.
 */
#ifndef CONFIG_FMP
#if architecture(cray)
#define CONFIG_FMP CONFIG_GENBIN/fmp
#else
#define CONFIG_FMP
#endif
#endif

/*
 * The environment variable CF77PP is set to the value of
 * CONFIG_GPP. The cf77(1) command uses this variable.
 */
#ifndef CONFIG_GPP
#if architecture(cray)
#define CONFIG_GPP CONFIG_GENBIN/gpp
#else
#define CONFIG_GPP
#endif
#endif

/*
 * The environment variable CFT77 is set to the value of
 * CONFIG_CFT77. The cf77(1) command uses this variable.
 */
#ifndef CONFIG_CFT77
#if architecture(cray)
#define CONFIG_CFT77 CONFIG_GENBIN/cft77
#else
#define CONFIG_CFT77
#endif
#endif
```

(continued)

```
/*
 * The environment variable GEN_SEGDIR is set to the value of
 * CONFIG_GEN_SEGDIR. The segldr(1) command uses this variable.
 */
#ifndef CONFIG_GEN_SEGDIR
#if architecture(cray)
#define CONFIG_GEN_SEGDIR CONFIG_GENBIN/./lib/segdirs
#else
#define CONFIG_GEN_SEGDIR
#endif
#endif

/*
 * Set C default to the generation cc compiler
 */
#ifndef CONFIG_GCC
#define CONFIG_GCC CONFIG_GENBIN/cc
#endif

/* Set nmake's CPP variable to the generation cpp */
#ifndef CONFIG_CPP
#define CONFIG_CPP CONFIG_GENBIN/./lib/cpp
#endif

/*
 * The following definition contains a list of products that must exist as
 * executables. This list is verified to exist when any part of UNICOS is
 * regenerated via nmake(1).
 */
#ifndef CONFIG_GENCMDS
#define CONFIG_GENCMDS CONFIG_FPP CONFIG_FMP CONFIG_CFT77 CONFIG_GPP\
    CONFIG_GCC CONFIG_CPP
#endif

/*
 * The following sets the generation software directory paths. The
 * environment variable PATH is set to the value of CONFIG_PATH.
 */
#ifndef CONFIG_PATH
#if architecture(cray)
#define CONFIG_PATH CONFIG_GENBIN:/bin:/usr/bin:/usr/ucb
#else
#define CONFIG_PATH /bin:/usr/bin:/usr/ucb:/usr/local/bin
#endif
#endif
```

(continued)

```
/*
 * The following sets the message catalog directory paths.
 * The environment variable NLSPATH is set to the value of
 * CONFIG_NLSPATH.
 */
#ifndef CONFIG_NLSPATH
#if architecture(cray)
#define CONFIG_NLSPATH /lib/nls/En/%N.cat
#else
#define CONFIG_NLSPATH
#endif
#endif

/*
 * The following variable is intended to disable/enable the -o option
 * on a cpset. Most released software doesn't use the -o option thus
 * the following variable has no effect.
 */
#ifndef CONFIG_CPSAVE
#define CONFIG_CPSAVE 0
#endif

/*
 * TCP/IP Network System
 */
#ifndef CONFIG_TCP
#define CONFIG_TCP 1
#endif

/*
 * X11 Window Management System
 * CONFIG_TCP must be configured also if using this option.
 */
#ifndef CONFIG_X11
#define CONFIG_X11 1
#endif
```

(continued)

```
/*
 * Several Cray 1/XMP/YMP specific options.
 */

/*
 *
 * Cross targetted libraries (XLIBS).
 *
 * In many cases, users require alternate libraries for binary
compatibility.
 *
 * For example, on a Cray-YMP machine, you may need Cray-XMP compatible
 * (24 bit) libraries. On a Cray-Y16 machine, you may need Cray-YMP
 * compatible (vector length = 64) libraries.
 *
 * These libraries are termed "cross targetted" (X) libraries. The
definition
 * tags CONFIG_XLIBS and CONFIG_XLIBTARGET facilitates the activation of
this
 * feature. You must set CONFIG_XLIBTARGET to reflect the architecture you
 * are targetting the libraries for.
 */

/*
 * Build cross targetted libraries (XLIBS)
 *
 * If you want the cross targetted libraries built and installed,
 * set the CONFIG_XLIBS variable to 1.
 */

#ifndef CONFIG_XLIBS
#define CONFIG_XLIBS 0
#endif
```

(continued)

```
/*
 * The following sets the TARGET for the desired set of cross targeted
 * libraries, i.e. cray-xmp, cray-x4 or cray-ymp.
 *
 * Note that these are not required on Cray-2 generations,
 * and can be set to null (e.g. #define CONFIG_XLIBTARGET)
 * in this case.
 */
#ifndef CONFIG_XLIBTARGET
#define CONFIG_XLIBTARGET cray-x4
#endif

#ifndef CONFIG_DECNET
#define CONFIG_DECNET 1
#endif

/*
 * Guest-mode/Native mode UNICOS kernel
 */
#ifndef CONFIG_GOS
#define CONFIG_GOS 0
#endif

/*
 * Expander chassis peripheral device drivers (uts kernel)
 */
#ifndef CONFIG_EXPANDER
#define CONFIG_EXPANDER 1
#endif

/*
 * Bit matrix multiply functional unit (uts kernel)
 */
#ifndef CONFIG_BMM
#define CONFIG_BMM 0
#endif

/*
 * IOS Model E support.
 */
#ifndef CONFIG_IOS_E
#define CONFIG_IOS_E 0
#endif
#endif
```

(continued)

```
/*
 * IOS serial numbers
 */
#ifndef CONFIG_IOSA_SN
#define CONFIG_IOSA_SN 0
#endif
#ifndef CONFIG_IOSB_SN
#define CONFIG_IOSB_SN 0
#endif

/*
 * Number of IOS machines (0, 1, or 2).
 */
#ifndef CONFIG_NIOS
#define CONFIG_NIOS 1
#endif

/*
 * End of Cray 1/XMP/YMP specific options.
 */

/*
 * Cray-2 specific options.
 */

/*
 * If you want the Cray-2 flow libraries built and installed,
 * set CONFIG_FLOWLIBS to 1.
 */
#ifndef CONFIG_FLOWLIBS
#define CONFIG_FLOWLIBS 0
#endif

/*
 * End Cray-2 specific options.
 */
```

(continued)

```
/*
 * UNICOS Station Control Programs
 */
#ifndef CONFIG_USCP
#define CONFIG_USCP 0
#endif

/*
 * Network Monitor
 */
#ifndef CONFIG_NETMON
#define CONFIG_NETMON 0
#endif

/*
 * SuperLink
 */
#ifndef CONFIG_SL
#define CONFIG_SL 0
#endif

/*
 * If set to one (1), the CONFIG_IPI3 variable turns on the ipi3
 * capability in the kernel.  If set to zero (0), the code is turned
 * off.  Its default is zero (0), or OFF.  Note that IPI3 is supported
 * only on Model E machines.
 */

#if CONFIG_IOS_E
#ifndef CONFIG_IPI3
#define CONFIG_IPI3 0
#endif /* CONFIG_IPI3 */
#endif /* CONFIG_IOS_E */

/*
 * Ultra
 */
#ifndef CONFIG_ULTRA
#define CONFIG_ULTRA 0
#endif
```

(continued)

```

/*
 * Operator's workstation
 */
#ifndef CONFIG_OWS
#define CONFIG_OWS 0
#endif

/*
 * Enable repeatable relocatables
 * Clear timestamp in pdt.
 */
#ifndef CONFIG_GENPROD_RULES
#define CONFIG_GENPROD_RULES 1
#endif

/*
 * The online diagnostics require that the directory where the
 * online diagnostics are kept be defined. The following variable
 * must be specified as a full path name to a directory where you
 * want the online diagnostics to be placed.
 */
#ifndef CONFIG_DIAGDIR
#define CONFIG_DIAGDIR /ce
#endif

/*
 * Default umask(1) to be used during the build
 */
#ifndef CONFIG_UMASK
#define CONFIG_UMASK 022
#endif
#else
$( :E=The file make/config.mh has been included more than once)
#endif

```

config.h *configuration*
file
 2.6.2.4

For certain configurations, the default UNICOS kernel parameter settings may not be adequate. For example, configurations with more than one Ethernet controller may require a larger number of mbufs defined in the kernel. See *UNICOS System Administration*, publication SG-2113, for more information about optimal kernel parameter settings for your system.

```

/*
 * SN5102 - config.h - Edition 0 [Wed Nov 25 11:13:32 CST 1992]
 * Created by Configuration Generator Rev. 70.65
 */
/* USMID @(#)uts/c1/cf/config.h70.23 10/16/92 03:56:21 */
/* COPYRIGHT CRAY RESEARCH, INC.
 * UNPUBLISHED -- ALL RIGHTS RESERVED UNDER
 * THE COPYRIGHT LAWS OF THE UNITED STATES.
 */
#define NBUF_FCTR 30 /* default factor of cache to user memory */
#define NHBUF_FCTR 2 /* default factor of hash to cache */
#define NBLK_FCTR 20 /* factor for max blocks per request of cache */
#define NPBUF 200 /* number of physical (raw) buffer headers */
#define NASYN 400 /* number of async I/O headers */
#define MAXASYN 35 /* max number of outstanding async requests */
#define NSU_ASYNC 4 /* number of asyn headers reserved for su */
#define MAXPIPE 20 /* Max # of blocks per pipe */
#define MAXRAH 8 /* Max read aheads */
#define NCRBUF 4 /* #of chkpnt() and restart() buffers */
#define NLDMAP 250 /* number of filesystems open concurrently */
#define NMNT 250 /* number of slice structures */
#define NMT 16 /* number of ddal tape devices */
#define NLDCH 0 /* Number of SSD logical device cache headers */
#define LDCHCORE 0 /* Memory clicks reserved for ldcache blocks */
#define NSIDEBUF 1 /* Number of blocks for side door, if in use */
/* must be a multiple of disk i/o unit */

#define NPTY 128
#define NUSERS 200 /* max # of users (share) */
#define NPROC 450 /* Number of processes */
#define NSESS 300 /* #of session */
#define NCALL 50
#define NMOUNT 75
#define NFILE 1500
#define NINODE 1000
#define NC1INODE NINODE /* NINODE and NC1INODE must be equal */
#if FILE_QUOTAS
#define NQUOTA 0 /* Default number of in-core quota entries */
#else
#define NQUOTA 0
#endif
#define NTEXT 40
#define NCLIST 1000
#define NFLOCKS 100 /* #of file lock regions */
#define ACNICE HZ /* Save lowest nice val. after n secs.*/
#define CDLIMIT (1L<<23) /* max file size (2^23 blocks = 34*10^9 bytes)*/
#define MAX_UNLINKED_BYTES 26214400
#if ( (MODEL_E && BMX_CONF) || (MFSUBTYP == XMP6XX || MFSUBTYP == YMPEL) )
/*
 * The amount of total system buffer space for tapes will be
 * TAPE_MAX_PER_DEV * TAPE_MAX_CONF_UP bytes.
 */

```

```

#define TAPE_MAX_PER_DEV      65536  /* max number of bytes per device */
                                /* for buffered I/O */
                                /* This will be rounded up to the */
                                /* next 4K multiple */
#define TAPE_MAX_CONF_UP     16      /* max number of tapes that may be */
                                /* configured up */
#endif
/*
 * Data management (archive) parameters.
 * DMODE is a users default offline file retrieval mode.
 * DMODE = 0 selects manual only retrieval,
 * DMODE = 1 selects automatic as well as manual retrieval.
 */
#define DMODE                 1      /* default archived file retrieval mode */
/*
 * The following defines are SLIPC related.
 */
#define SL_IPCS               100    /* Maximum IPC devices must be greater than
                                SLMAXSERV for Superlink to operate.
                                Must be greater than 1 for the
                                kernel to compile. */
#define SL_KIPCMS             4*SL_IPCS /* Max kernel copies of IPC messages.
                                One for each outstanding IPC I/O */
#define SL_WLISTS             2*SL_IPCS /* max wait list entries used to skip
                                IPC write messages queued and waiting
                                for a matching read. */
#ifdef CRAY2
/*
 * restrict the number of pblks Superlink uses. This
 * may be further reduced to NPBUF/2
 * at boot time.
 */
#define SL_NPBUFS SL_IPCS
#else
#define SL_NPBUFS 2*SL_IPCS
#endif
#define SL_NBUFS 2*SL_IPCS /* Restrict the number of NBUF used by
                                Superlink to this define.
                                At boot time, this may further
                                be reduced to NBUF/2. */
#define SL_MAXMSGs           25     /* Maximum number of several informational
                                messages that are printed by SLIPC
                                and SLFS. There messages are printed
                                when there is a resource shortage.
                                If the messages get in the way, set
                                this define to zero, or if memory
                                is available, increase the resource. */
#define SL_ABSMAXMSG         4 * 65536 /* the absolute maximum a process
                                may set s_maxread */
/*
 * The following defines are SLFS related.

```

```

/*
 * SN5102 - config.h - Edition 0 [Wed Nov 25 11:13:32 CST 1992]
 * Created by Configuration Generator Rev. 70.65
 */
/* USMID @(#)uts/c1/cf/config.h70.23 10/16/92 03:56:21 */
/* COPYRIGHT CRAY RESEARCH, INC.
 * UNPUBLISHED -- ALL RIGHTS RESERVED UNDER
 * THE COPYRIGHT LAWS OF THE UNITED STATES.
 */
#define NBUF_FCTR 30 /* default factor of cache to user memory */
#define NHBUF_FCTR 2 /* default factor of hash to cache */
#define NBLK_FCTR 20 /* factor for max blocks per request of cache */
#define NPBUF 200 /* number of physical (raw) buffer headers */
#define NASYN 400 /* number of async I/O headers */
#define MAXASYN 35 /* max number of outstanding async requests */
#define NSU_ASYNC 4 /* number of asyn headers reserved for su */
#define MAXPIPE 20 /* Max # of blocks per pipe */
#define MAXRAH 8 /* Max read aheads */
#define NCRBUF 4 /* #of chkpnt() and restart() buffers */
#define NLDMAP 250 /* number of filesystems open concurrently */
#define NMNT 250 /* number of slice structures */
#define NMT 16 /* number of ddal tape devices */
#define NLDCH 0 /* Number of SSD logical device cache headers */
#define LDCHCORE 0 /* Memory clicks reserved for ldcache blocks */
#define NSIDEBUF 1 /* Number of blocks for side door, if in use */
/* must be a multiple of disk i/o unit */

#define NPTY 128
#define NUSERS 200 /* max # of users (share) */
#define NPROC 450 /* Number of processes */
#define NSESS 300 /* #of session */
#define NCALL 50
#define NMount 75
#define NFILE 1500
#define NINODE 1000
#define NCLINODE NINODE /* NINODE and NCLINODE must be equal */
#if FILE_QUOTAS
#define NQUOTA 0 /* Default number of in-core quota entries */
#else
#define NQUOTA 0
#endif
#define NTEXT 40
#define NCLIST 1000
#define NFLOCKS 100 /* #of file lock regions */
#define ACNICE HZ /* Save lowest nice val. after n secs.*/
#define CDLIMIT (1L<<23) /* max file size (2^23 blocks = 34*10^9 bytes)*/
#define MAX_UNLINKED_BYTES 26214400
#if ( (MODEL_E && BMX_CONF) || (MFSUBTYP == XMP6XX || MFSUBTYP == YMP6EL) )
/*
 * The amount of total system buffer space for tapes will be
 * TAPE_MAX_PER_DEV * TAPE_MAX_CONF_UP bytes.
 */

```

```

#define MLS_INTEGRITY    NORMAL /* OFF integrity code in secure.c */
/*
 * Parameters used for machine generated passwords
 */
#define RANDOM_PASS_ON  0      /* 1 = machine generated passwords enabled
                                0 = disabled */
#define PASS_MINSIZE    7      /* machine generated passwords minimum size */
#define PASS_MAXSIZE    8      /* machine generated passwords maximum size */
/*
 * The following defines make NFS more useful while running
 * a secure system at the price of a less secure system.
 * Define NFS_SECURE_EXPORT_OK to be a non-zero value if it is ok
 * to export a secure file system via NFS.
 * The NFS code will then allow a remote system to access
 * files at security level 0 with no ACLs and no compartments.
 * Define NFS_REMOTE_RW_OK to be a non-zero value if it is ok
 * to NFS-mount a remote file system in read-write mode.
 * The NFS code will then allow users running at security level 0
 * with no compartments to write files to the remote
 * file system.
 */
#define NFS_SECURE_EXPORT_OK 1 /* non-zero if ok to export via NFS */
#define NFS_REMOTE_RW_OK    1 /* non-zero if ok to import via NFS */
#define STAT_RESTRICT    NORMAL /* ON restrict stat() and secstat() to */
/* active level or below */
#define MAXLOGS          5      /* max login attempts */
#define LOGDELAY         0      /* secs to delay between login tries */
#define DELAY_MULT       0      /* multiplier for failed logins */
#define DISABLE_TIME     600    /* duration in seconds for which the user
                                will be disabled when the MAXLOGS param
                                is exceeded. If -1 (or any negative
                                value), disable indefinitely */
#define CONSOLE_MSG      0      /* send a message to the console when */
/* MAXLOGS has been reached */
#define DISABLE_ACCT     1      /* disable an account when MAXLOGS exceeded */
#define MINSLEVEL        0      /* minimum security level */
#define MAXSLEVEL        16     /* maximum security level */
#define SYSVCOMPS        07777777777777777777 /* valid system compartments */
#define SECURE_SYSTEM_CONSOLE "" /* secure console: administration */
#define SYSTEM_ADMIN_CONSOLE "/dev/console" /* default console: admin. */
#define SECURE_OPERATOR_CONSOLE "/dev/console" /* secure console: operator */
#define COMPART_ACTIVE_DEFAULT 0 /* generic defaults */
#define COMPART_VALID_DEFAULT 0
#define DECLASSIFY_DISK  0      /* activate declsfy disk when not 0 */
#define OVERWRITE_COUNT  3      /* declassify disk overwrite count */
#define DECLASSIFY_PATTERN 0 /* declassify disk write pattern */
#define SANITIZE_PATTERN  0      /* scrub disk write pattern */
/*
 * TCP/UDP/IP network related stuff
 */
/*

```

```

*      TCP_NMBSPACE
*
*      This symbol controls the number of mbufs (TCP managed memory buffers).
*      This pool of buffers is allocated at boot time in
*      usr/src/uts/tcp/kern/uipc_mbuf.c and cannot be increased while the
*      system is running.  Factors which influence the number of mbufs
*      to allocate are:
*
*      - Number of routing entries (each requires 1 mbuf)
*      - Number of active sockets (each requires 3 mbufs for the
*        life of the connection).
*      - The above entries account for a small, but fixed pool of
*        mbufs.  Additional entries are required to handle packet
*        headers, plus the data queued for input or output to active
*        connections.  Allocating the maximum amount of data for these
*        connections is a waste, but deciding on a lower bound is
*        impossible.
*      - NFS.  NFS consumes about 250 mbufs for static and dynamic
*        memory allocators.  This is with one UIDMAP configured
*        for several machines.  More mbufs (up to 100 more) will
*        be used with an average load, and more will be needed
*        for each uidmap as well.
*
*      In the Mendota environment, our mbuf allocation is:
*
*          with NFS          without NFS
*      Static routes (40 routes)      40          40
*      65 users, fixed entries (3 ea) 195          195
*          3 buffers each (avg)      195          195
*      NFS w/2 UID maps              300
*
*          Minimum allocation      730          430
*
*      We've set our number of mbufs 50% above the minimums to handle
*      transient peaks, file transfer and remote shell connections, and the
*      like.
*
*      For a more complete treatment of how to set the TCP_NMBSPACE parameter,
*      review the section on mbufs in the administrator's guide.
*/
#ifdef TCP_NMBSPACE
#define TCP_NMBSPACE 1800
#endif
/*
*      SECURE_NET_OPTIONS
*
*      This symbol is used to select options for running tcp/ip on
*      a secure system.  The global kernel variable secure_net_options
*      is initialized with this value.  This variable is a bit mask
*      in which the following bits may be set:
*
*      NETW_EXEMPT_NAL          When this bit is set, hosts which do

```

```

*           not appear in the NAL are allowed access
*           to the system at the unclassified level
*           (level = 0, compartments = 0).  When not
*           set, such hosts are denied access to the
*           system.
*/
#define SECURE_NET_OPTIONS NETW_EXEMPT_NAL
/* end of TCP/UDP/IP defines */

```

Customizing root (/) and /usr file systems

2.7

After relinking the UNICOS kernel, you must customize the root (/) and /usr file systems before changing the system to multiuser mode. For further information about customizing the root (/) and /usr file systems, see *UNICOS System Administration*, publication SG-2113.

Changing to multiuser mode

2.8

After you have made the necessary changes to the UNICOS kernel parameter settings and to the root (/) and /usr file systems, enter the following to enter multiuser mode:

```
# /etc/init 2
```

The following messages are displayed on your screen:

```
INIT: New run level: 2
Day. Mon. X HH:MM:SS CST 1993
```

If you correctly set the time on your IOS, the time reported here should be correct, because the UNICOS clock is set from the IOS clock (see “Setting the IOS clock,” page 7).

To change the date and time here, enter them in the format *MMDDhhmmYYss* and press **RETURN**. You will receive several messages and a console login prompt.

You are now in multiuser mode. The root password is *initial*, which you must change using the `passwd` command. For details about the `passwd` command, see the *UNICOS User Commands Reference Manual*, publication SR-2011.

Adding tape devices

2.9

If you plan to add additional tape devices to your CRAY Y-MP EL system, you should configure them at this time. To use the tape daemon software provided as part of the UNICOS standard release, see *UNICOS System Administration*, publication SG-2113.

If you choose not to use the tape daemon software provided as part of the standard UNICOS release, you must configure additional tape devices that are added to your CRAY Y-MP EL system by using character special files. To configure additional tape devices by using character special files, see “Online Tape System,” page 75.

Boot procedures completed

2.10

The procedures for booting the preinstalled software are complete. You have booted the preinstalled IOS and UNICOS software, configured your disk space, customized the root (/) and /usr file systems, and are now ready to begin system operations.

Installing Fortran and C compilers

2.11

If you plan to use the Cray Research Fortran or C compilers and they have not been preinstalled on your CRAY Y-MP EL system, you must order and install them as a separate product. To install the compilers, refer to the installation guide shipped with the compilers.

Configuring MLS

2.12

If you want to configure multilevel security (MLS) on your system at this time, see “UNICOS Multilevel Security,” page 99.

Upgrade Installation [3]

This section describes the procedures for sites that are currently running CRAY Y-MP EL system software and are upgrading to a new release level. The currently mounted root (/), /usr, and /usr/src file systems will be modified by this upgrade procedure.

Preparing to upgrade

3.1

To begin, you must perform a full backup of your file systems by using the UNICOS `dump` utility because the upgrade procedure modifies the current mounted file systems. Also, if you do not have a current backup of your IOS disk, back that up now. See the *UNICOS Administrator Commands Reference Manual*, publication SR-2022, for information about the `dump` command. The following is an example of a root (/) file system dump to the quarter-inch cartridge (QIC) tape:

```
# dump -t 0 -f /dev/rpq01 /dev/dsk/root
```

To verify that the dumps completed successfully, use the `restore -i` command. This command lets you examine the contents of the tape without actually restoring anything. See the *UNICOS Administrator Commands Reference Manual*, publication SR-2022, for information about the `restore` command.

If you have back-up root (`bkroot`) and back-up /usr (`bkusr`) file systems available, use them as online backups for your current system. You can use the following commands to perform these disk-to-disk backups:

```
# /etc/dump -t 0 -f - /dev/dsk/root | /etc/restore -x -f -  
# /etc/dump -t 0 -f - /dev/dsk/usr | /etc/restore -x -f -
```

Saving your 6.1 configuration files

3.2

The kernel configuration file `conf.c` must be saved to `/usr/tmp`. (This is the same file that was used to build the current running kernel.) Log in as `root` on your currently running system in multiuser mode and enter the following commands to save the `conf.c` kernel configuration file. In the following examples, replace `SN` with your system's serial number.

```
# cp /usr/src/uts/c1/cf/conf.SN.c /usr/tmp/conf.c
```

Note

The `els_upgrade` script expects this kernel configuration file to exist in `/usr/tmp` and be named `conf.c`.

Also place a copy of your current `config.mh` file into `/usr/tmp` for reference purposes. You may want to refer to this later when setting up your UNICOS 7.0 `config.mh` file. Use the following command line to do this:

```
# cp /etc/config/config.mh /usr/tmp/config.mh
```

If your site is running NQS, you will need to follow these steps to prepare for the conversion of your NQS database:

1. To avoid losing jobs, ensure that all NQS queues are empty (no queued or running jobs). Jobs left in the queues will be lost when upgrading to UNICOS 7.0. To empty the queues, first disable the pipe queues. When the pipe queues are empty, disable the batch queues. Wait until the batch queues are empty and no jobs are running before stopping NQS.
2. Type `qmgr` to get the `qmgr` prompt (`Qmgr 61.x:`)
3. Save the current NQS configuration. Saving your configuration with the following command will allow NQS to automatically rebuild your NQS configuration under UNICOS 7.0. At the `qmgr` prompt, type the following:

```
snap file=/etc/config/nqs_config
```

4. Exit `qmgr` by typing `exit` at the `qmgr` prompt.
5. Stop NQS with the `qstop` command (see `qstop(8)`).

The disk space requirements for `/usr/src` on CRAY Y-MP EL systems have been dramatically reduced for the UNICOS 7.0 release. The recommended default value is 50,000 blocks. This is sufficient to hold all of the files necessary to relink UNICOS kernels and to build the CF77 and Cray Standard C (SCC) compilers. Table 3 shows the specific CRAY Y-MP EL systems' requirements for UNICOS and various other products.

Table 3. Disk space on `/usr/src` required for UNICOS and other products

Product	Size in 4 Kbyte blocks
UNICOS 7.0	25,000
CF77 5.0	11,500
SCC 3.0	9,500
Pascal 4.2	2,300
Data migration	15,400
DWB	9,100
OSI 2.0	44,700

Upgrading to a new release level

3.3

To upgrade to the UNICOS 7.0 release of the CRAY Y-MP EL system software, use the following procedures:

1. Determine your site's disk space needs from Table 3 and then modify the existing `config.uni` file to provide the required space. You will use this partition for the new `/dev/dsk/src` file system that is required for the CRAY Y-MP EL system software 7.0 release (see "Configuring the peripherals," page 70, for information about modifying the `config.uni` file).

You can reclaim for your own use up to 100,000 disk blocks from the `/dev/dsk/src` file system when reconfiguring for UNICOS 7.0 because of the new, more efficient packaging scheme for 7.0.

The following examples depict parts of the `config.uni` file before changes and after changes are made to allocate 50,000 blocks for the `/dev/dsk/src` file system. Specific disk space requirements at your site dictate the number of blocks to allocate for this file system.

Figure 2 shows the physical definition of an example `config.uni` file before changes are made. In this example, `tmp` file system space is used to create the 50,000-block disk partition using a DD-3 disk drive.

```
ESDI_010 := DD3 MIOP 1 UNIT=0 (
    0:usr_e010      /* 50000 blocks */
    150000:tmp_e010 /* 90000 blocks */
    240000:fs4_e010 /* 94200 blocks */
);
```

Figure 2. Physical definition of an example `config.uni` file before modifications

Figure 3 shows the physical definition of an example `config.uni` file after changes are made. In this example, `tmp` file system space from two DD-3 disk drives is used to create a 50,000-block partition for the `src` file system.

```
ESDI_010 := DD3 MIOP 1 UNIT=0 (
    0:usr_e010      /* 50000 blocks */
    150000:src_e010 /* 50000 blocks */
    200000:tmp_e010 /* 40000 blocks */
    240000:fs4_e010 /* 94200 blocks */
);
```

Figure 3. Physical definition of an example `config.uni` file after modifications

Figure 4 shows the logical definition of an example `config.uni` file before changes are made. In this example, `tmp` file system space is changed and `src` file system space is added.

```
tmp := (                               /* 990000 blocks */
    tmp_e001
    tmp_e010
    tmp_e011
    tmp_e012
    tmp_e003
    tmp_e013
);
```

Figure 4. Logical definition of an example `config.uni` file before modifications

Figure 5 shows the logical definition of an example `config.uni` file after changes are made. `tmp` file system space is changed and space is allocated for the `src` file system.

```
src := (                               /* 50000 blocks */
    src_e010
);
tmp := (                               /* 940000 blocks */
    tmp_e001
    tmp_e010
    tmp_e011
    tmp_e012
    tmp_e003
    tmp_e013
);
```

Figure 5. Logical definition of an example `config.uni` file after modifications

Enter the following command lines to create the new `config.uni` file:

```
# cd /
# exdf -i sys/config.uni > config.uni
# vi config.uni
(add the 50,000-block partition for /dev/dsk/src)
```

2. After you have added the 50,000-block partition for `/dev/dsk/src`, verify the new `config.uni` file by entering the following command line:

```
# bconfig config.uni
```

3. Using the `exdf` command, write the new `config.uni` file back to the IOS by entering the following:

```
# exdf -o sys/config.uni < config.uni
```

4. Switch your CRAY Y-MP EL system to single-user mode by entering the following command lines:

```
# shutdown 0 (wait for shutdown to complete)  
# sync; sync
```

6. Insert the CRAY Y-MP EL system 7.0 release tape into QIC tape device.

Caution

Gently insert the tape into the Anaconda QIC tape device until it “clicks” into the load position and the drive takes the tape and completes loading it.

7. Load the new IOS files from the QIC tape by entering the following command lines:

```
# CONTROL-a (toggles to the IOS console)  
IOS> cd /  
IOS> tar -xvf rpq01
```

The `tar` process takes about 10 minutes.

8. After the new IOS files are loaded, reload the IOS and reboot your CRAY Y-MP EL system to single-user mode by entering the following command lines:

```
IOS> reload  
IOS> boot
```

9. Check the `root (/)` file system by entering the following:

```
# fsck /dev/dsk/root
```

10. Create and check the new `/dev/dsk/src` partition by entering the following command lines:

```
# mkfs -q /dev/dsk/src
# fsck /dev/dsk/src
# labelit /dev/dsk/src src 0
```

Note

If you used `tmp` space to create your `src` partition (page 48), you must now enter the following additional command:

```
# mkfs -q /dev/dsk/tmp
```

11. Mount the `/usr` file system by entering the following command lines:

```
# fsck /dev/dsk/usr
# mount /dev/dsk/usr /usr
```

Note

If your site is running UNICOS multilevel security (MLS), mount the file system(s) in the path for the disk resident security logs (`/usr/adm/sl`, by default). Then start the security log daemon, `/etc/slogdaemon`, by entering the following command line:

```
# /etc/slogdaemon
```

12. Mount the `/usr/src` and `tmp` file systems by entering the following command lines:

```
# mount /dev/dsk/src /usr/src
# fsck /dev/dsk/tmp
# mount /dev/dsk/tmp /tmp
```

The `/usr/src` file system should be empty at this point. If you have previously split your `/usr` file system into multiple file systems such as `/usr/adm`, `/usr/spool`, and so on, mount these file systems also at this time.

13. Transfer the five new upgrade utilities (/scripts/els_upgrade, /scripts/els_update, /scripts/unlabel.sh, /scripts/udbcvt.sh and /usr/gen/bin/ldproto) and the new /etc/config/config.mh configuration file from the IOS to UNICOS by entering the following command lines:

```
# cd /
# exdf -i install/scripts.tar > /tmp/scripts.tar
# tar -xvf /tmp/scripts.tar
```

14. Using the file command, verify that you have the special devices necessary to read directly from the QIC tape device without using the tape daemon. The devices that must exist are /dev/rpq01 and /dev/nrpq01. (The command, "file /dev/*rpq*", should display the character special devices. The minor device numbers may be different from the following example, but must be unique.) If these devices do not exist, create them by entering the following:

```
# mknod /dev/rpq01 c 43 0 0 8 64 0 1 0 0 0
# mknod /dev/nrpq01 c 43 1 0 8 64 0 1 0 1 0
```

15. If you are not using MLS, skip to step 16; otherwise, run the unlabel.sh script to unlabel the MLS commands, as shown following:

```
# cd /scripts
# ./unlabel.sh
```

16. Edit the /etc/config/config.mh file as needed, using the following command, to set your system's serial number and to enable or disable features such as tapes or security. Comments in the file provide more information about this.

Caution

Change only the necessary items in section I and, if you want, the optional items in section II of /etc/config/config.mh. **Do not** change anything in section III of this file, or the resulting UNICOS kernel may not build or boot correctly.

```
# vi /etc/config/config.mh
```

17. Invoke the `els_upgrade` script, with the following two commands:

```
# cd /etc/install
# /scripts/els_upgrade
```

This script takes about one hour to complete. Answer the three questions as they appear, or press `RETURN` to use the defaults shown in brackets []. When you see the following message, the script is done:

```
=====
Successfully installed package. Upgrade process complete.
=====
```

The `els_upgrade` script completes the following functions:

- a. Positions the QIC tape device to the correct place on the installation tape, verifies the package type, and uses the `ldproto` command to load the UNICOS 7.0 package. Absolute binary files in the root (/) and /usr file systems are replaced and the /usr/src/uts directory is populated with the files needed to link a UNICOS kernel.
 - b. Reads the `conf.c` file from /usr/tmp (placed there in the step on page 46) and creates the /usr/src/uts/cf.SN directory, where SN is your system's serial number.
 - c. Runs the import and activate scripts from the UNICOS Installation and Configuration Menu System.
 - d. Relinks a new kernel in /usr/src/uts/cf.SN.
18. Make a back-up copy of your current kernel (SN is your system's serial number):

```
# cd /usr/src/uts/cf.SN
CONTROL-a (toggles to the IOS console)
IOS> mv /sys/unicos.ymp /sys/unicos.old
```

19. Transfer the new kernel to the IOS disk by entering the following commands:

```
CONTROL-a RETURN (toggles to UNICOS console)
# exdf -o sys/unicos.ymp < unicos
```

20. Unmount the file systems by entering the following:

```
# cd /
# /etc/umountem
# /etc/shutdown 0
# sync; sync; sync
CONTROL-a (toggles to the IOS console)
```

21. Use the following commands to remove the MEMORY= statement from the config.uni file before booting the UNICOS 7.0 kernel:

```
IOS> cd /
IOS> ed sys/config.uni
```

Caution

Failing to remove the MEMORY statement (required in earlier releases) from the config.uni file may cause a pscan error when booting.

22. Now boot the UNICOS 7.0 kernel, as follows:

```
IOS> boot (boots your UNICOS kernel to single-user mode)
```

23. If you are not using MLS, skip to step 24; otherwise, enter the following commands to label your trusted binaries and files:

```
# /etc/tsubcmd -i -1
# /etc/udbcmd -i -1
```

The -1 (label only) option properly labels security-sensitive files without removing them from their target locations.

See "Upgrading your MLS system to 7.0," page 111, for further information on MLS configurations.

24. Convert your udb to UNICOS 7.0 format, then start up multiuser mode by entering the following:

```
# fsck /dev/dsk/root
# mount /dev/dsk/usr /usr
# /scripts/udbcvt.sh
```

```
=====
Converting udb to new format
=====
```

Multi Level Security (Y/N) [N]:

```
UDB conversion.
215 entries converted to source
Input style: udb
added 215 records
Header tables updates 3 times
```

UDB conversion complete.

Network Queueing System (NQS)

3.3.1

Due to changes to the internal NQS structures between UNICOS 6.1 and UNICOS 7.0, the `/usr/spool/nqs` database from previous CRAY Y-MP EL UNICOS releases cannot be used with UNICOS 7.0 NQS. Consequently, no NQS jobs will be preserved across the upgrade to 7.0.

To convert your NQS database to UNICOS 7.0, enter the following command:

```
# /scripts/nqscvt.sh
```

For more information on NQS differences, refer to the “Compatibilities and Differences” section of the *UNICOS 7.0 Release Overview for CRAY Y-MP EL Systems*, publication RO-5200.

Network file system (NFS) UID maps

3.3.2

The binary format of NFS UID maps has changed between UNICOS release 6.1 and 7.0. To create a 7.0 UID map, you must recreate the UID maps by using the 7.0 utilities. This is typically done by a script called `/etc/uidmaps/Merge.domains`. You can create this script for

your site using the file `/etc/uidmaps/Merge.exmple` as a template. The standard UNICOS script, `/etc/nfsstart`, will call the `Merge.domains` script if the marker file `/etc/uidmaps/.merge` exists.

Upgrade procedures completed

3.4

The upgrade procedures are now complete. Your CRAY Y-MP EL system is ready to begin system operations. To boot the system to multiuser mode, type the following commands:

```
# cd /  
# /etc/umountem  
# init 2
```

Reinstalling UNICOS [4]

If you are doing an initial installation (not using preinstalled software), follow the procedures described in this section. Initial installations are performed at the factory prior to shipping.

Caution

The procedures in this section are for an initial installation and may destroy any preinstalled CRAY Y-MP EL system software.

Tape and tape device

4.1

Your CRAY Y-MP EL system software tape set consists of three quarter-inch cartridge (QIC) tapes.

- One CRAY Y-MP EL system software tape that contains both the UNICOS and IOS software. This tape is labeled “ELS UNICOS 7.0 RELEASE EL-7.0 UNICOS FOR Y-MP EL ONLY.”
- One IOS back-up tape that contains configuration files, flaw files, and a relinked kernel.
- One scan file tape.

Figure 6 illustrates the Anaconda QIC tape device found on your CRAY Y-MP EL system.

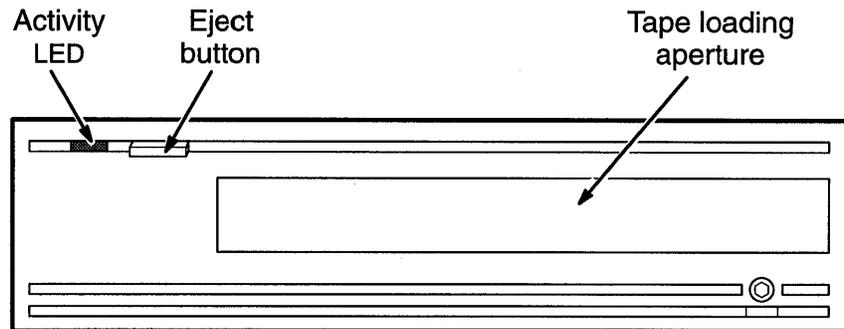


Figure 6. Anaconda QIC tape device

The QIC device is recognized as `rst0` when the system is operating from programmable read-only memory (PROM) and either `rst0` or `rpq01` when operating from the IOS. On UNICOS, the QIC device is recognized as `rpq01`.

Note

When a new tape is loaded into the tape drive, you must allow about 30 seconds for the drive to ready the tape. Understanding that there is a delay is important when trying to access the tape through a UNIX character special file interface (either from the IOS or from UNICOS) rather than through the UNICOS tape daemon.

When your system is powered up, you will see the `BOOT>` prompt on your system console. Before you can install the IOS software, you must create an IOS file system on the SCSI hard disk, as described in subsection 4.2. After the file system is created, transfer the IOS files from the QIC tape device to the IOS (SCSI) disk device by using the `tar` command, as described in subsection 4.3.

Create the IOS file system

4.2

The following subsections describe the procedures for creating and installing the IOS file system. To create the IOS file system on the SCSI disk, enter the following:

Warning

The `mkfs` command destroys all files previously written to the IOS disk.

```
BOOT> mkfs c:
```

This process takes about 7 minutes to complete. If the IOS file system has not been created yet, the system responds by displaying output similar to the following on your console screen:

```
Cluster size --- 4096 bytes
Max. Contiguous Free - 48373
  In Use          0
  Free           48373
  Bad             0
  Reserved       0
  Erroneous      0

Total           48373
```

The following message indicates that the file system has already been created and mounted on your CRAY Y-MP EL system:

```
disk buffer 'cachel:' mounted
'wd1a:' already mounted
```

If you encounter this message on your CRAY Y-MP EL system, you must unmount the existing file system by entering the following command:

```
BOOT> umount c:
```

After the unmount operation is complete, create the new IOS file system by entering the following:

```
BOOT> mkfs c:
```

Install the IOS software

4.3

CRAY Y-MP EL systems can support up to 16 I/O subsystems. On CRAY Y-MP EL systems that have more than one IOS, the master IOS has interfaces to the system console and its own Small Computer System Interconnect (SCSI) bus with a Winchester hard disk, a quarter-inch cartridge (QIC) device, and an optional 8-mm EXABYTE tape device.

To install the IOS files, insert the QIC tape labeled "ELS UNICOS 7.0 RELEASE EL-7.0 UNICOS FOR Y-MP EL ONLY" into your QIC tape device.

Caution

Gently insert the tape into the QIC tape device until it "clicks" into the load position. The Anaconda QIC tape device senses the tape and automatically completes the load operation.

After the tape is inserted, load the IOS files by entering the following:

```
BOOT> tar -xvf rst0
```

If the command fails and a `blocksize = 0` message is displayed, issue the `tar` command a second time to read the tape. This is a known problem with the QIC driver in PROM, which will be fixed in a future PROM revision.

The `BOOT>` prompt returns after about 10 minutes. During the loading of the IOS files, you will see output similar to the following on the your console screen:

```
BOOT> tar -xvf rst0
x ./ios/ios          100 tape blocks
x ./ios/ios.nm      8 tape blocks
.
.
.
```

Install the scan files

4.4

At this point, you should install your scan files from the QIC tape that contains these files. The tape contains instructions for loading these files onto your IOS SCSI disk. If you do not have current scan files on tape, contact your CRI customer service representative and install them before continuing.

Copy site-dependent IOS files

4.5

After you load the IOS files, you must copy one file and one script from the `/install` directory to their proper destinations. This step ensures that subsequent installations of IOS software releases do not overwrite the critical IOS and UNICOS files.

At the `BOOT>` prompt, enter the following command lines to copy the generic UNICOS kernel and boot script to their proper locations on the IOS disk:

```
BOOT> cp /install/unicos.ymp /sys/unicos.ymp
BOOT> cp /install/boot.ymp /bin/boot
```

IOS /config file

4.5.1

The procedures in this subsection automatically generate the `/config` file on the IOS. This file contains information about the mainframe architecture, the number of CPUs, and the strategies and driver software types loaded into the IOS during the initial system boot.

The following example `/config` file reflects a sample IOS in which the master IOS is configured with ESDI disk controllers, a standard SCSI bus, and console. The slave IOS is configured with an Ethernet connection and IPI disk controllers. If your site ever alters its hardware configuration (adds or removes CPUs, IOSs, controllers, peripherals, etc.), you must modify this file to reflect those changes.

```

#
# Define architecture:
#
ARCH=YMP-EL
#
# Define the slots (0, 1, 2, and 3) in the system cabinet that have CPUs.
# This CRAY Y-MP EL configuration has one CPU in slot 0.

NCPUS=0
#
# If this configuration had a multiple CPU system with CPUs in slots 0 and 2, the entry
# would be NCPUS=02.
#
# The master IOS is always IOS0. The 1274 value is the serial number from this
# sample IOP board. The serial number is stored in nonvolatile random-access memory
# (NVRAM) and is unique for every Heurikon board. Figure 7, page 64, shows IOS
# Heurikon boards. Refer to this illustration for the serial number location on IOS
# Heurikon boards.
#
# In a single IOS system, the IOS0: SN statement is optional unless the MME software
# is running on the Cray Research maintenance workstation (MWS). It is good practice
# to include this statement.
#
IOS0: 1274 1228800
#
# The following are strategy names, which are modules that manage packets,
# communicate with UNICOS, and handle memory management with the I/O buffer
# board (IOBB). Only the first (master) IOS has the /dev/console strategy; the
# master IOS also requires /dev/tape.
#
# Strategy name
# _____
# /dev/disk
# /dev/tape
# /dev/console

```

(continued)

```

# The possible device strategies are as follows:
#
# /dev/console (master IOS only)
# /dev/disk (disks)
# /dev/ethnet (Ethernet)
# /dev/hynet (HYPERchannel – strategy and driver)
# /dev/fdnet (FDDI network)
# /dev/tape (tapes – nontape daemon)
# /dev/taped (tapes – tape daemon)
#
# The following are device drivers. The device drivers you use in your configuration
# must be listed and loaded after the strategy names: they are device specific. The
# master IOS also requires /dev/stape.

# Device driver name
# _____
# /dev/stape
# /dev/esdi

# The possible device drivers are as follows:
#
# /dev/s3560 (SCSI controller driver – if included, this driver must precede
# /dev/stape)
# /dev/stape (EXABYTE and QIC tapes)
# /dev/esdi (ESDI disk driver)
# /dev/das (disk array)
# /dev/ether (Ethernet driver)
# /dev/ipi (IPI-2 disk driver)
# /dev/bm (buffer memory resident)
# /dev/pertec (STK or PERTEC 9-track tape)
# /dev/fddi (Fiber Distributed Data Interface driver)
#
IOS1: 1632 1228800
#
# Strategy names:
# _____
# /dev/disk
# /dev/ethnet
#
# Device drive names:
# _____
# /dev/ipi
# /dev/ether

```

Figure 7 shows an IOS Heurikon board that is found inside the right-side panel of your CRAY Y-MP EL system.

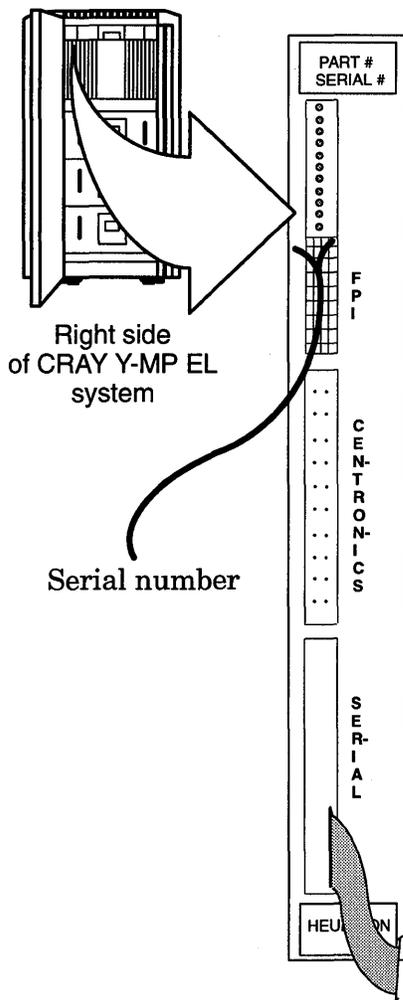


Figure 7. IOS Heurikon boards

**IOS /sys/unicos.ymp
file**
4.5.2

The IOS /sys/unicos.ymp file is a generic, bootable UNICOS kernel file. This kernel file is used when initially booting your system. Later, as part of the installation process, you will create a new custom kernel and write it to the /sys/unicos.ymp file.

IOS /bin/boot script

4.5.3

The /bin/boot start script boots the UNICOS system and contains the IOS commands that are used to load the UNICOS kernel and to begin communication between the IOS and UNICOS.

Automatic initial installation procedure

4.6

A new, automated initial installation procedure has been developed for the UNICOS 7.0 release for CRAY Y-MP EL systems. This procedure automates most of the initial install process. Three IOS-resident utilities (in the /install directory) are used in this process that perform various functions as described below:

<u>Utilities</u>	<u>Description</u>
ldios	This utility prompts the installer for the system architecture, number of CPUs, memory size, number of IOSs and their Heurikon serial numbers, and whether the tape daemon should be run. It then generates the /config file on the IOS without any strategies or drivers defined.
drivers	This utility probes all IOSs to determine the number and types of controllers that exist. It then updates the IOS /config file appropriately and executes a reload command to the IOS to activate strategies and drivers for all system controllers.
ldunicos	This utility optionally creates a default /sys/config.uni file, which UNICOS uses to allocate disk space to file systems. It then boots UNICOS, using only a DDRAM resident iroot file system (that is, no disk space required) and loads the default root (/), /usr, and /usr/src file systems from the release tape.

To perform an initial installation on a CRAY Y-MP EL system, follow the steps below:

1. At the `BOOT>` prompt, enter `/install/ldios`. Answer all of the questions. You will see output similar to the following:

```
BOOT> /install/ldios

System architecture[YMP-EL]:

Number of mainframe cpus[1]: 3

Memory size (mw) of mainframe[32]: 64

Enter number of ios(s) [1]: 2

Enter the serial number for IOS0: 1870

Enter the serial number for IOS1: 1990

Configure the tapedaemon[n]: y
```

2. When you have answered all the preceding questions, issue a load command at the `BOOT>` prompt:

```
BOOT> load

output from load

IOS>
```

3. At the `IOS>` prompt, enter `/install/drivers`:

```
IOS> /install/drivers

output from drivers

IOS>
```

4. At the IOS> prompt, type `/install/ldunicos` and answer the questions, as follows:
 - First question: If you want to automatically configure the disks, answer Y. This creates a `/sys/config.uni` file automatically. If you already have created a customized `/sys/config.uni` file, answer N.
 - Second question: To automatically load the root (`/`), `/usr`, and `/usr/src` file systems from the release tape, answer Y.

After you answer the questions, a DDRAM-resident `iroot` file system will be used to boot UNICOS, and the load of the UNICOS file systems will commence.

This procedure assumes that the root (`/`), `/usr`, and `/usr/src` file systems will reside on the logical file systems `/dev/dsk/root`, `/dev/dsk/usr`, and `/dev/dsk/src`, respectively.

Note

If you answer N to both questions, you may run the `els_install` utility in single-user mode to choose the different logical names for the file systems.

This step takes approximately 90 minutes.

```
IOS> /install/ldunicos

Do you want to auto configure the disks[Y]:

Autogen Normal Termination

Do you want to auto install the software[Y]:

more output from ldunicos

Installing of system software has successfully completed.

#
```

Edit the boot script

4.7

With your bootable UNICOS utilities installed, you must use the `ed` command to edit the `/bin/boot` script on the IOS. It must be edited to match the number of CPUs and memory size on your CRAY Y-MP EL system before booting UNICOS.

```
# CONTROL-a (toggles to the IOS console)
IOS> ed /bin/boot
```

A sample boot script file follows:

```
mc
mc
#
# Uncomment the following jstar command for each additional CPU present
# on your CRAY Y-MP EL system. For example, if you have a system with two CPUs,
# uncomment the entry ds /sys/jstar 1.
#
# ds /sys/jstar 3
# ds /sys/jstar 2
# ds /sys/jstar 1
#
echo Zeroing memory. Please wait...
#
# Uncomment the zero memory command that matches your machine's memory size.
#
# ds /sys/zero128 0
# ds /sys/zero64 0
# ds /sys/zero32 0
#
wait 5
echo Clear memory complete. Loading UNICOS...
lu /sys/unicos.ymp /sys/config.uni
echo Starting unicos
iostart
ds
conswitch
```

Set the IOS clock

4.8

To verify that the IOS clock is set for your local date and time and to adjust this clock, use the `time` command:

```
IOS> time
```

The time reported and set by this command is the time that UNICOS will boot with; therefore, be sure it is correct, so you do not have to change it later for UNICOS and then have your UNICOS clock be out of sync with your IOS clock.

For more information on the `time` command, see the *CRAY Y-MP EL System-specific Administrator Commands Reference Manual*, publication SR-2408.

Note

Use the `rcmd` command to execute IOS commands on slave IOSs (slave IOSs are those controlled by the IOS designated as the master IOS. The master IOS can perform actions that the slaves cannot). For information about the `rcmd` command, see the *CRAY Y-MP EL System-specific Administrator Commands Reference Manual*, publication SR-2408.

Reboot the UNICOS system

4.9

Now use the following command lines to shut down UNICOS and reboot the system from disk:

```
IOS> CONTROL-a RETURN (toggles to the UNICOS console)
# shutdown 0
# sync; sync; sync
CONTROL-a (toggles to the IOS console)
IOS> boot
```

To complete the booting of your UNICOS system, go to “Modify the parameter file `config.uni`,” page 18, and follow the steps there to the end of section 2.

Relink the UNICOS kernel

4.10

Go to “Relink the UNICOS kernel,” pages 19 through 42, and follow the steps shown there, then return here and continue with the following subsection.

Configure the peripherals

4.11

A generic `/sys/config.uni` file (on the IOS disk) was generated automatically by the automated initial installation process described in “Automatic initial installation procedure,” page 65.

This subsection describes how to reconfigure your peripherals by editing the automatically generated `/sys/config.uni` file and changing the configuration specification language (CSL) variables.

Using CSL, you will define run-time system configurable variables, your disk peripherals, and file systems. See *UNICOS System Administration*, publication SG-2113, for more information about CSL.

To familiarize you with CSL, an overview is provided. The overview includes a description of CSL, provides parameter setting recommendations, and a sample configuration file.

The first step involves the use of the UNICOS CSL to define run-time system configurable variables, your disk peripherals, and file systems. The second step, used to customize your UNICOS kernel, is presented in “Relink the UNICOS kernel,” page 19. You must complete both steps as part of the installation of the UNICOS software.

Note

System performance is significantly impacted by disk configuration. See *UNICOS System Administration*, publication SG-2113, for details about file system placement.

Configuration specification language (CSL) overview

4.11.1

CSL is used to describe configuration and run-time parameter settings that are interpreted when you boot UNICOS. CSL statements permit you to extend or override the default configuration by letting you do the following:

- Define the physical attributes of your disk devices
- Define the logical grouping of your physical disk slices
- Define system devices
- Configure logical device cache

- Define memory size (required)
- Define the number of IOSs, CPUs, and channels

Your `/sys/config.uni` file contains the definitions of your physical, logical, and system disk devices. Optionally, `/sys/config.uni` can contain CSL statements to modify or override specific UNICOS kernel parameters and the additional statements required to configure UNICOS disk striping.

Each time the UNICOS operating system is booted, UNICOS interprets and sequentially processes the CSL statements; therefore, the order of your CSL statements is very important. After processing the CSL file, UNICOS automatically creates the following parameter files that reflect the current system configuration. These files are written to the following UNICOS file names:

<u>File name</u>	<u>Definition</u>
<code>/IOS-param</code>	This file is an exact copy of your CSL file.
<code>/CONFIGURATION</code>	This file contains CSL-defined definitions and compiled UNICOS kernel definitions.
<code>/unicos</code>	This file is a copy of the running UNICOS kernel.

CSL recommendations 4.11.2

When configuring UNICOS during the installation process, you must use CSL to define the size and location of the disk region used for the root (`/`), `/usr`, and `/usr/src` file systems. You must also use CSL to define the size and location of the UNICOS swap and dump devices, `/bkroot` (back-up root), `/bkusr` (back-up `/usr`), `/tmp`, and `/core`. The remaining space is typically allocated as `/home` or site-defined file systems. The recommended size of the dump device is 65,536 blocks.

Note

Disk and buffer memory resident (BMR) partitions are defined in units of blocks (1 block = 4096 bytes) on CRAY Y-MP EL systems.

Balancing I/O
4.11.2.1

To balance your I/O, place the root (/) and /usr file systems on different disk drives. If possible, also isolate the root (/) and /usr file systems on different disk controllers. The recommended disk drive for the root (/) and /usr file systems is the DD-4 disk drive; use the DD-3 and DDES DI disk types as alternatives.

**Defining the root (/),
/usr, and /usr/src file
systems**
4.11.2.2

The root (/), /usr, and /usr/src file systems that you define in this step will become the location of your root (/), /usr, and /usr/src file systems when you run `els_install`. You must define the root (/) and back-up root (/bkroot) file systems as the same size; they require a minimum of 90,000 blocks. You must also define the /usr and back-up /usr (/bkusr) file systems as the same size; they require a minimum of 150,000 blocks. The /usr/src file system requires a minimum of 50,000 blocks.

Configure the swap device
4.11.2.3

Generally, you should configure your swap device to be three to eight times larger than your UNICOS main memory. If your system includes a disk array subsystem (DAS), use it for your swap device. By using the DAS for your swap device, your CRAY Y-MP EL system performance will be optimized because the DAS is best suited for large-block sequential I/O. The DAS is also an excellent location for seldom-used file systems or application disk space.

If you have no DAS, configure your swap device on available DD-4 or DD-3 disk drives. Locate your swap device on different disk drives than those used for your user's /home directories and applications.

Configure the stripe groups
4.11.2.4

Stripe group members must be placed on the same disk type (for example, DD-4), be the same length, and start at the same block. When possible, place stripe group members on separate controllers.

A striped swap configuration example of `stripe_20` and `stripe_30` on IPI disk drives follows:

```
STRIPED := DDSTR (
    stripe_20 + stripe_30 /* 150000 blocks */
);
swap := (
    stripe_20
    /* 150000 blocks */
);
```

*Using a buffer memory
resident (BMR) device*
4.11.2.5

When using a BMR device, you should adhere to the following rules:

- Striped file systems cannot contain BMR partitions.
- Banded file systems (physically adjacent file systems on a disk) may be composed of BMR partitions from more than one IOS.
- BMR and disk partitions can be mixed in file systems that are not striped.
- The swap device can be defined to include partitions of BMR from more than one IOS.
- A BMR device defined as `ldcache` can be used only to cache file systems that are wholly composed of disk partitions that reside within the same IOS as that BMR device.
- The CSL `BMRDEV` statement cannot be defined to include partitions of a BMR device from more than one IOS.

Disk types

4.11.3

Table 4 shows the disk types that can be configured on your CRAY Y-MP EL system:

Table 4. CRAY Y-MP EL system disk types

Disk type	Description	Number of user blocks per device	Sectors/track
DDESDI	Old ESDI drive	170,100	9
DD-3	New ESDI drive	334,200	10
DR-1	Removable ESDI	334,200	10
DD-4	IPI-Sabre 7	653,000	28
DDL DAS	Old DAS	1,269,114	65
DDAS2	New DAS	2,502,000	75
DDBMR	VME RAM device	Multiple of 32,768	—
DDRAM	Central memory resident	Configuration dependent	—

Sample CSL settings

4.11.4

The sample `/sys/config.uni` file on page 13 contains CSL for a CRAY Y-MP EL system with two CPUs and two IOSs. This file exemplifies a configuration for the basic file systems and ten sample file systems (`fs1...fs10`). Each file system name has a suffix of the form `_dxyz`, where the letters have the following meanings:

<u>Component</u>	<u>Meaning</u>
<i>d</i>	Device type
<i>x</i>	IOS number
<i>y</i>	Controller number
<i>z</i>	Unit number

Online Tape System [A]

CRAY Y-MP EL systems are designed with a tape subsystem that provides you with two methods of accessing the tape devices. One method uses the feature-rich tape daemon software provided as part of a standard UNICOS software release. This method offers features such as multivolume support, labeled tapes, tape silo access, and data buffering. The tape daemon also offers users a lot of tape device control, flexibility, and security.

Use of the standard UNICOS tape daemon requires you to specify a tape daemon configuration file. All user tape device requests are made through the tape daemon, which actually manages the devices. See *UNICOS System Administration*, publication SG-2113, for information about configuring tape devices under tape daemon control.

The second method is similar to the traditional UNIX approach of accessing tape devices by using character special files. The character special file tape interface permits standard UNIX control over tape devices. To create these character special files, use the standard `mknod` UNICOS utility. Information about how to create character-specific files is provided in this appendix.

Note

Multivolume support is not provided with the character special file tape interface.

For information about cleaning and maintaining your tape subsystem, see the *CRAY Y-MP EL Basic Administration Guide*, publication SG-2416.

Configuring your online tape system

A.1

Your CRAY Y-MP EL system is initially supplied with sufficient special files for most basic device configuration. You must configure additional tape devices that will not be controlled by the tape daemon by creating additional character special files. These additional character special files establish a connection between an arbitrary path name and the actual specified device. Each special file resides under the UNICOS /dev directory. The following subsections describe the components of special files that support additional devices on your CRAY Y-MP EL system.

Components of character special files

A.2

Each character special file consists of a special file name and mknod information. The following command line creates a new character special file under the /dev directory. The special file name is defined first, followed by the mknod information:

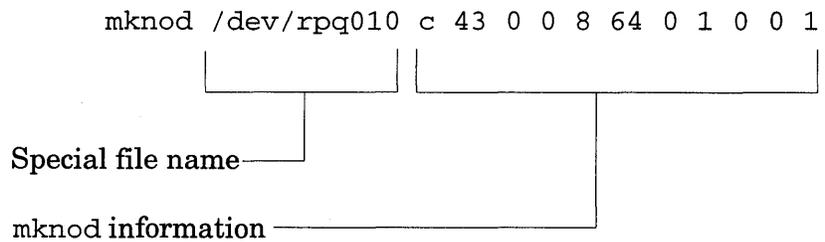


Figure 8 shows the components of special file names and mknod information.

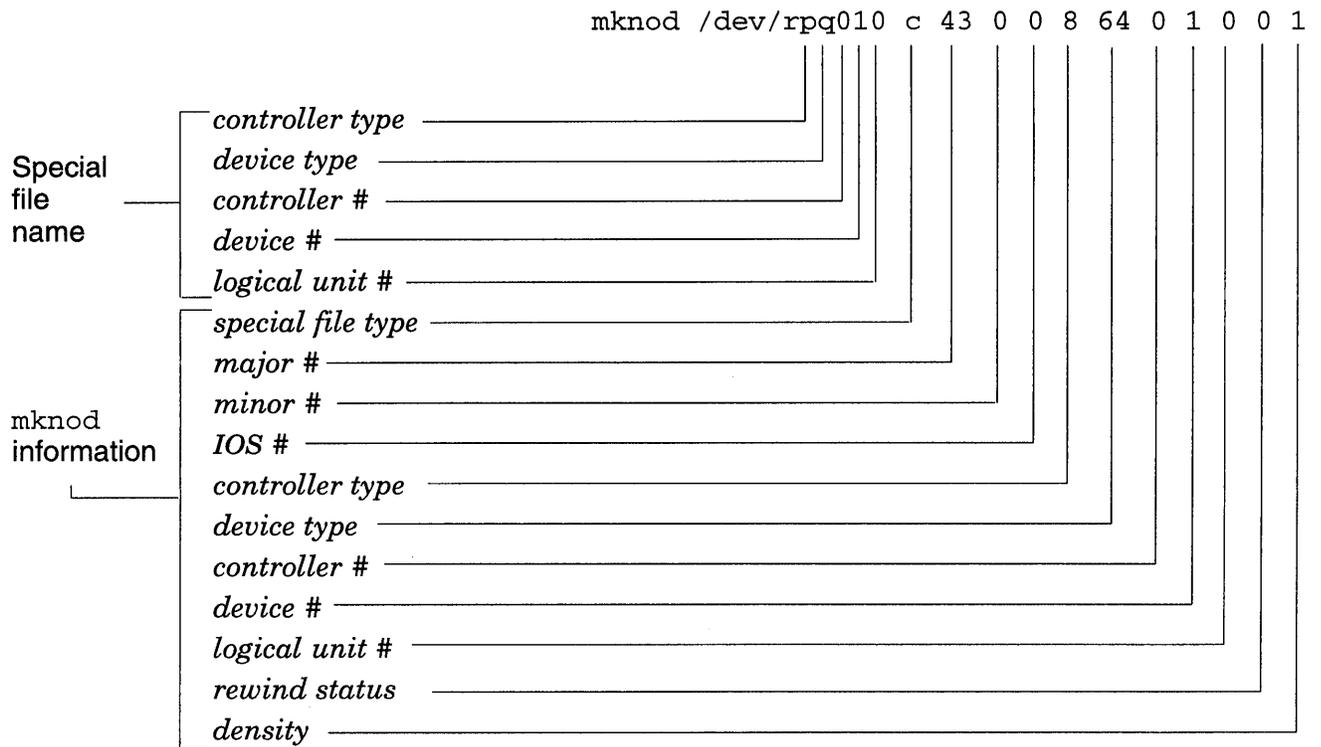


Figure 8. Special file name and mknod components

Tape special file naming convention
A.2.1

The special file name variables are as follows:

<u>Variable</u>	<u>Description</u>										
<i>controller type</i>	Governs the functions of the attached devices. Possible values are as follows:										
	<table border="0"> <thead> <tr> <th style="text-align: left;"><u>Type</u></th> <th style="text-align: left;"><u>Description</u></th> </tr> </thead> <tbody> <tr> <td>p</td> <td>I/O processor</td> </tr> <tr> <td>s</td> <td>Small Computer Systems Interface (SCSI) controller</td> </tr> <tr> <td>m</td> <td>PERTEC controller</td> </tr> </tbody> </table>	<u>Type</u>	<u>Description</u>	p	I/O processor	s	Small Computer Systems Interface (SCSI) controller	m	PERTEC controller		
<u>Type</u>	<u>Description</u>										
p	I/O processor										
s	Small Computer Systems Interface (SCSI) controller										
m	PERTEC controller										
<i>device type</i>	Defines device being used. Possible values are as follows:										
	<table border="0"> <thead> <tr> <th style="text-align: left;"><u>Type</u></th> <th style="text-align: left;"><u>Description</u></th> </tr> </thead> <tbody> <tr> <td>q</td> <td>Quarter-inch cartridge (QIC) tape device</td> </tr> <tr> <td>e</td> <td>EXABYTE tape device</td> </tr> <tr> <td>s</td> <td>STK square tape</td> </tr> <tr> <td>t</td> <td>9-track tape (PERTEC)</td> </tr> </tbody> </table>	<u>Type</u>	<u>Description</u>	q	Quarter-inch cartridge (QIC) tape device	e	EXABYTE tape device	s	STK square tape	t	9-track tape (PERTEC)
<u>Type</u>	<u>Description</u>										
q	Quarter-inch cartridge (QIC) tape device										
e	EXABYTE tape device										
s	STK square tape										
t	9-track tape (PERTEC)										
<i>controller #</i>	Defines controller being used. Possible values are as follows:										
	<table border="0"> <thead> <tr> <th style="text-align: left;"><u>Type</u></th> <th style="text-align: left;"><u>Description</u></th> </tr> </thead> <tbody> <tr> <td>0</td> <td>I/O processor</td> </tr> <tr> <td>0-7</td> <td>SCSI and PERTEC</td> </tr> </tbody> </table>	<u>Type</u>	<u>Description</u>	0	I/O processor	0-7	SCSI and PERTEC				
<u>Type</u>	<u>Description</u>										
0	I/O processor										
0-7	SCSI and PERTEC										
<i>device #</i>	Defines device ID number. Possible values are as follows:										
	<table border="0"> <thead> <tr> <th style="text-align: left;"><u>Type</u></th> <th style="text-align: left;"><u>Description</u></th> </tr> </thead> <tbody> <tr> <td>0-6</td> <td>SCSI</td> </tr> <tr> <td>1-6</td> <td>I/O processor</td> </tr> <tr> <td>0-3</td> <td>PERTEC</td> </tr> </tbody> </table>	<u>Type</u>	<u>Description</u>	0-6	SCSI	1-6	I/O processor	0-3	PERTEC		
<u>Type</u>	<u>Description</u>										
0-6	SCSI										
1-6	I/O processor										
0-3	PERTEC										
<i>logical unit #</i>	Defines the SCSI logical unit being used; optional value. This field does not apply to PERTEC tapes. Possible values are as follows:										
	<table border="0"> <thead> <tr> <th style="text-align: left;"><u>Type</u></th> <th style="text-align: left;"><u>Description</u></th> </tr> </thead> <tbody> <tr> <td>0-7</td> <td>SCSI</td> </tr> </tbody> </table>	<u>Type</u>	<u>Description</u>	0-7	SCSI						
<u>Type</u>	<u>Description</u>										
0-7	SCSI										

Tape special file mknod fields

A.2.2

The variables of the special file mknod fields are as follows:

<u>Variable</u>	<u>Description</u>										
<i>special file type</i>	Specifies the special file type entry.										
	<table border="0"> <thead> <tr> <th style="text-align: left;"><u>Type</u></th> <th style="text-align: left;"><u>Description</u></th> </tr> </thead> <tbody> <tr> <td>c</td> <td>Creates character special file.</td> </tr> </tbody> </table>	<u>Type</u>	<u>Description</u>	c	Creates character special file.						
<u>Type</u>	<u>Description</u>										
c	Creates character special file.										
<i>major #</i>	Specifies major device number.										
	<table border="0"> <thead> <tr> <th style="text-align: left;"><u>Type</u></th> <th style="text-align: left;"><u>Description</u></th> </tr> </thead> <tbody> <tr> <td>43</td> <td>Major number is always 43.</td> </tr> </tbody> </table>	<u>Type</u>	<u>Description</u>	43	Major number is always 43.						
<u>Type</u>	<u>Description</u>										
43	Major number is always 43.										
<i>minor #</i>	Specifies minor device. Possible values are as follows:										
	<table border="0"> <thead> <tr> <th style="text-align: left;"><u>Type</u></th> <th style="text-align: left;"><u>Description</u></th> </tr> </thead> <tbody> <tr> <td>0-31</td> <td>This number can be any unique number between 0 and 31.</td> </tr> </tbody> </table>	<u>Type</u>	<u>Description</u>	0-31	This number can be any unique number between 0 and 31.						
<u>Type</u>	<u>Description</u>										
0-31	This number can be any unique number between 0 and 31.										
<i>IOS #</i>	Specifies the IOS on which the drive resides. Possible values are 0 through 15.										
<i>controller type</i>	Specifies the controller on which the drive resides. Possible values are as follows:										
	<table border="0"> <thead> <tr> <th style="text-align: left;"><u>Type</u></th> <th style="text-align: left;"><u>Description</u></th> </tr> </thead> <tbody> <tr> <td>8</td> <td>I/O processor</td> </tr> <tr> <td>9</td> <td>PERTEC controller</td> </tr> <tr> <td>10</td> <td>SCSI controller</td> </tr> </tbody> </table>	<u>Type</u>	<u>Description</u>	8	I/O processor	9	PERTEC controller	10	SCSI controller		
<u>Type</u>	<u>Description</u>										
8	I/O processor										
9	PERTEC controller										
10	SCSI controller										
<i>device type</i>	Specifies device being used. Possible values are as follows:										
	<table border="0"> <thead> <tr> <th style="text-align: left;"><u>Type</u></th> <th style="text-align: left;"><u>Description</u></th> </tr> </thead> <tbody> <tr> <td>6</td> <td>STK square tape 4220 or STK 18-track square tape (SCSI)</td> </tr> <tr> <td>64</td> <td>QIC tape device (SCSI)</td> </tr> <tr> <td>65</td> <td>EXABYTE tape device (SCSI)</td> </tr> <tr> <td>66</td> <td>9-track round tape (PERTEC)</td> </tr> </tbody> </table>	<u>Type</u>	<u>Description</u>	6	STK square tape 4220 or STK 18-track square tape (SCSI)	64	QIC tape device (SCSI)	65	EXABYTE tape device (SCSI)	66	9-track round tape (PERTEC)
<u>Type</u>	<u>Description</u>										
6	STK square tape 4220 or STK 18-track square tape (SCSI)										
64	QIC tape device (SCSI)										
65	EXABYTE tape device (SCSI)										
66	9-track round tape (PERTEC)										

<u>Variable</u>	<u>Description</u>								
<i>controller #</i>	<p>Defines controller being used. Possible values are as follows:</p> <table border="0"> <thead> <tr> <th style="text-align: left;"><u>Type</u></th> <th style="text-align: left;"><u>Description</u></th> </tr> </thead> <tbody> <tr> <td>0</td> <td>I/O processor</td> </tr> <tr> <td>0-7</td> <td>SCSI and PERTEC</td> </tr> </tbody> </table>	<u>Type</u>	<u>Description</u>	0	I/O processor	0-7	SCSI and PERTEC		
<u>Type</u>	<u>Description</u>								
0	I/O processor								
0-7	SCSI and PERTEC								
<i>device #</i>	<p>Device ID number. Possible values are as follows:</p> <table border="0"> <thead> <tr> <th style="text-align: left;"><u>Type</u></th> <th style="text-align: left;"><u>Description</u></th> </tr> </thead> <tbody> <tr> <td>0-3</td> <td>PERTEC</td> </tr> <tr> <td>0-6</td> <td>SCSI</td> </tr> <tr> <td>1-6</td> <td>I/O processor</td> </tr> </tbody> </table>	<u>Type</u>	<u>Description</u>	0-3	PERTEC	0-6	SCSI	1-6	I/O processor
<u>Type</u>	<u>Description</u>								
0-3	PERTEC								
0-6	SCSI								
1-6	I/O processor								
<i>logical unit #</i>	<p>Specifies the device logical unit number for SCSI devices. Possible values are 0 through 7.</p>								
<i>rewind status</i>	<p>Specifies whether the device is a rewind or no-rewind device. Possible values are as follows:</p> <table border="0"> <thead> <tr> <th style="text-align: left;"><u>Type</u></th> <th style="text-align: left;"><u>Description</u></th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Rewind device</td> </tr> <tr> <td>1</td> <td>No-rewind device</td> </tr> </tbody> </table>	<u>Type</u>	<u>Description</u>	0	Rewind device	1	No-rewind device		
<u>Type</u>	<u>Description</u>								
0	Rewind device								
1	No-rewind device								
<i>density</i>	<p>Specifies whether the device (9-track only) will write tapes in high density or low density. Possible values are as follows:</p> <table border="0"> <thead> <tr> <th style="text-align: left;"><u>Type</u></th> <th style="text-align: left;"><u>Description</u></th> </tr> </thead> <tbody> <tr> <td>0</td> <td>High density (6250 b/i)</td> </tr> <tr> <td>1</td> <td>Low density (1600 b/i)</td> </tr> </tbody> </table>	<u>Type</u>	<u>Description</u>	0	High density (6250 b/i)	1	Low density (1600 b/i)		
<u>Type</u>	<u>Description</u>								
0	High density (6250 b/i)								
1	Low density (1600 b/i)								

Sample character special files

A.3

Table 5 shows example command lines that create special character files for various devices and configurations.

Table 5. Example special character file command lines

Device	Configuration	Command line
QIC	—	mknod /dev/rpq01 c 43 0 0 8 64 0 1 0 0 0
		mknod /dev/nrpq01 c 43 1 0 8 64 0 1 0 1 0
EXABYTE	Attached to the IOP SCSI bus	mknod /dev/rpe02 c 43 2 0 8 65 0 2 0 0 0
		mknod /dev/nrpe02 c 43 3 0 8 65 0 2 0 1 0
EXABYTE	Attached to the VME SCSI controller	mknod /dev/rse02 c 43 4 0 10 65 0 2 0 0 0
		mknod /dev/nrse02 c 43 5 0 10 65 0 2 0 1 0
Two SCSI square tapes	One SCSI ID number and attached to the VME SCSI controller	mknod /dev/rss000 c 43 6 0 10 6 0 0 0 0 0
		mknod /dev/rss001 c 43 7 0 10 6 0 0 1 0 0
		mknod /dev/nrss001 c 43 8 0 10 6 0 0 1 1 0
		mknod /dev/nrss000 c 43 9 0 10 6 0 0 0 1 0
Two SCSI square tapes	Different SCSI ID numbers and attached to the VME SCSI controller	mknod /dev/rss000 c 43 10 0 10 6 0 0 0 0 0
		mknod /dev/rss010 c 43 11 0 10 6 0 1 0 0 0
		mknod /dev/nrss000 c 43 12 0 10 6 0 0 0 1 0
		mknod /dev/nrss010 c 43 13 0 10 6 0 1 0 1 0

SCSI
105
I

Panics and File System Crashes [B]

If your CRAY Y-MP EL system experiences an IOS assertion panic or a UNICOS system panic, you should perform IOS and UNICOS dumps to save the data in your system's memory and determine the cause of the panic.

The information retrieved from a system dump provides data that you can use to determine the cause of the panic. Therefore, if a system panic occurs, you should follow specific procedures to ensure a good system dump for complete data retrieval.

This appendix describes the procedures for conducting IOS and UNICOS dumps, and it provides procedures for recovering from a root (/) file system crash.

Types of panics

4.12

There are three types of panics:

- IOS assertion panic
- IOS processor fault panic
- UNICOS panic

An IOS assertion panic indicates that the IOS software has encountered a catastrophic problem. When this occurs, the console displays a message similar to the following indicating the cause of the problem (*n* is the IOS number):

IOS*n* ASSERTION PANIC: *error message text*

An IOS processor fault panic indicates some type of unexpected hardware condition. The IOS error message will be followed by a register dump:

IOS*n*: *error message text*
register dump

A UNICOS panic occurs when the UNICOS operating system encounters a catastrophic problem. In such a case, a message similar to the following, indicating the cause of the problem, will be displayed on the console:

```
IOSn NOTICE: date time UNICOS PANIC: error message text
```

IOS assertion panic

B.1

When either an IOS assertion panic or an IOS processor fault panic occurs, the IOS PROM code starts executing in the affected IOS. The `BOOT>` prompt is displayed, if the affected IOS is the master IOS. For both of these types of errors, an IOS dump of the IOS issuing the panic is automatically performed by the CRAY Y-MP EL system. However, if it is necessary to perform an IOS dump manually, follow these steps to ensure a good system dump:

1. Issue the `iosdump` command for each IOS in your CRAY Y-MP EL system. See “IOS dumps,” following, for more information.
2. Reload each IOS by executing the `reload` command.
3. Dump the mainframe memory and registers by following the instructions in “Using the `mfdump` command on CRAY Y-MP EL systems,” page 87.

IOS dumps

B.2

The `iosdump` command saves both the memory of the IOPs and the I/O buffer board (IOBB) memory to the small computer system interface (SCSI) hard disk. Invoke the `iosdump` command at the `BOOT>` prompt.

The format of the `iosdump` command follows:

```
iosdump [-n filename] [-s size]
```

The arguments are optional. The IOS generates a default file name and the `iosdump` command halts an operating IOS.

The command lines that follow use the `iosdump` command:

```
BOOT> iosdump
```

```
Saving IOS dump to /ADM/DUMP0/DO31992.0...done
```

The following example uses the `-n` option, which lets you specify a file name for the contents of the dump:

```
BOOT> iosdump -n /adm/iosdump1
```

```
Saving IOS dump to /ADM/IOSDUMP1...done
```

You must use the `rcmd` command to dump IOSs other than IOS0. This command cannot be run from the `BOOT>` prompt.

```
IOS> rcmd ios4 iosdump
```

```
Saving IOS dump to /ADM/DUMP4/DO31992.0...done
```

If an IOS panic has occurred in the master IOS (IOS0), you will not be able to take a dump of slave IOSs without reestablishing communications between the master IOS and the slave IOS. You must be careful not to destroy information contained on the slave IOS. Reloading the IOSs after the panic using the default configuration file will destroy information necessary to analyze the cause of the panic. Use the following steps to dump slave IOSs after an IOS0 panic. The IOS prompt should be `BOOT>`.

1. Save the old configuration file (`/config`) in `config.bak`.

```
# cp /config /config.bak
```

2. Change the configuration file to boot a single IOS. Edit `/config` and remove all information about slave IOSs. Only IOS0 information should remain in the `/config` file.

If you have saved a copy of this file from a previous instance of taking IOS dumps (see step 7), you can copy this saved file (`/config.0`) to `/config`.

3. Load IOS0. Issue the IOS command `load`.
4. Initialize the inter-IOS channel to the slave IOS. This step is necessary to establish communications between the master IOS and the slave. Use the IOS command `iosinit` to establish this link. The format of this command follows:

```
iosinit iosN SN
```

where N is the is slave IOS number (1, 2, 3, ...) and SN is the serial number of the Heurikon IOP board in the slave IOS. This serial number can be obtained from the `/config.bak` file created in step 1. For example, to dump IOS2, find the entry in the `/config.bak` for IOS2. It will look like:

```
IOS2: 1594
```

The number 1594 is the serial number. In this example, the `iosinit` command would be as follows:

```
iosinit ios2 1594
```

5. Dump the slave IOS. Use the IOS command `rcmd` to dump the slave IOS (N is the number of the slave IOS):

```
rcmd iosN iosdump
```

6. To dump other slave IOSs, repeat steps 4 and 5 using the appropriate IOS number.
7. Restore the original configuration file. When all IOS dumps have been taken, save the new configuration for future use.

```
# cp /config /config.0
```

Next, move the original IOS configuration file back into the default location:

```
# cp /config.bak /config
```

8. Reload the IOS. Use the IOS command `reload` to reload the IOS with the standard configuration.

If a mainframe dump is also desired, you may now proceed. See subsections B.3 and B.4 for information on taking a mainframe dump. If a mainframe dump is not needed, you may continue with booting the system as usual with the IOS command `boot`.

UNICOS dumps

B.3

You should perform a system dump with each UNICOS or IOS system panic. To perform a system dump, you must have allocated disk space to receive the dump. That partition is usually called `/dev/dsk/dump`. The size of `/dev/dsk/dump` may vary, but it must be large enough to contain the dump. You must also initialize `/dev/dsk/dump` before each use. Usually, the initialization of the dump partition and copying of the dump are accomplished in the `/etc/coredd` start-up script.

When performing a system dump, you will use the following commands:

<u>Command</u>	<u>Function</u>
<code>/etc/coredd</code>	Start-up script
<code>/etc/idmp</code>	Initializes the dump partition
<code>/etc/cpdmp</code>	Copies the dump from <code>/dev/dsk/dump</code> to a file for subsequent processing
<code>/etc/crash</code>	Analyzes the dump file

See the *UNICOS Administrator Commands Reference Manual*, publication SR-2022, for more information the on `coredd`, `idmp`, `cpdmp`, and `crash` commands.

Using the `mfdump` command on CRAY Y-MP EL systems

B.4

The memory dump utility, `mfdump`, does not write to the dump disk partition if that partition is not formatted properly. To initialize the dump partition, issue the UNICOS `idmp` command. You must initialize the dump partition before the system is down because it cannot be done afterward; the `/etc/coredd` script usually does this during UNICOS startup.

Before taking a mainframe dump on the CRAY Y-MP EL system, you should save the scan chains from the CPUs. This is a temporary workaround to retrieve the hardware registers because `mfdump` will not do that yet. To save the scan chains on a four-CPU system, enter the following commands at the `IOS>` prompt:

```
IOS> clk off
IOS> bscan o 0 /tmp/cpu0
IOS> bscan o 1 /tmp/cpu1
IOS> bscan o 2 /tmp/cpu2
IOS> bscan o 3 /tmp/cpu3
IOS> clk on
```

If your system has fewer than four CPUs, enter the `bscan` commands that relate to the CPUs you have installed. The `bscan` command saves the scan chain information on the IOS SCSI disk in file `/tmp/cpux` in the preceding example. You must analyze this file along with the dump that `mfdump` creates. After saving the scan chains, reload the IOS before executing the `mfdump` command. As of the UNICOS 7.0 release, you no longer need to reload the IOS before trying to boot UNICOS.

An example of the execution of `mfdump` follows. Comments are preceded by a `#` and do not actually appear when running `mfdump`.

Note

The `mfdump` command will prevent you from accidentally overwriting unprocessed dumps. If your dump partition contains an unprocessed dump, indicated by the `mfdump` error message, `Unable to force dump`, you can force a dump by using `mfdump -F` command. As of the UNICOS 7.0 release, you no longer need to reload the IOS between executions of the `mfdump` command.

See the *CRAY Y-MP EL System-specific Administrator Commands Reference Manual*, publication SR-2408, for more information on the `mfdump` command.

```
# The -q option forces these questions. The information you enter is stored in
# /sys/mfdump.arg. If that file does not exist, mfdump will enter query mode
# automatically. After mfdump.arg is created, you may use mfdump with no
# arguments and bypass these questions.
#
IOS> mfdump -q

Enter hardware configuration information and dump memory size: (in
megawords: legal values are 32, 64, 128, 256)
64
Number of cpus: (legal values: 1, 2, 3, 4, 5, 6, 7, 8)
4
Do you want a register dump? (y/n)[y]
y
Do you want a system register dump? (y/n)[n]
y
Do you want to dump the ublocks? (y/n)[y]
y
#
# mfdump lets you dump up to four noncontiguous sections of memory.
#
Enter the memory ranges to dump (in words).

Enter memory range 0.
Low:
0
High:
16000000

Enter memory range 1.
Low:
0
High:
0

Enter memory range 2.
Low:
0
High:
0
```

(continued)

Enter memory range 3.

Low:

0

High:

0

Enter dump device information.

This information is in the /CONFIGURATION file on the UNICOS system.

#

This is the mainframe channel number (in decimal) that connects to the IOS where
the disk dump partition resides. CRAY Y-MP EL systems that have only one IOS or
a CRAY Y-MP EL system in which the dump partition is on IOS0, the channel
number is 16 (020). On CRAY Y-MP EL systems that have multiple IOSs, the
channel number is in octal in the configuration file and must be converted to
decimal.

#

Enter the channel number:

16

#

This is the MIOP number of the controller in which the dump partition resides.

#

Enter the IOS number (0-15):

0

Enter the controller number:

0

Enter the dump disk type (18 = DAS, 19 = DAS-2, 16 = ESDI, 17 = DD3,
20 = DD4):

17

Enter the unit number:

0

#

Enter the starting block and length of the disk partition.

#

Enter the starting block:

0

Enter the length in blocks:

131072

(continued)

```
#  
#mfdump redisplay your answers and asks whether you want to save your answers  
#permanently before performing the dump.
```

Recovery from a root (/) file system crash

B.5

If your root (/) file system becomes corrupt, you must determine the cause. If the corruption is because of a bad disk or other hardware problem, you must remedy this before reloading the software.

Use the following steps to recover from a root (/) file system crash. This procedure assumes that a back-up copy of the root (/) file system (bkroot) is available.

1. Load the IOS by entering the following:

```
BOOT> load
```

2. Edit the configuration file so that it boots from bkroot, as follows:

```
IOS> ed /sys/config.uni
```

3. Modify the values for ROOTDEV and PIPEDEV to point to bkroot rather than root (/), as follows:

```
ROOTDEV = bkroot;  
PIPEDEV = bkroot;
```

4. Boot UNICOS by using /dev/dsk/bkroot as the root (/) device, as follows:

```
IOS> boot
```

5. Replace the corrupt root (/) file system with a clean copy (that is, bkroot) by entering the following:

```
# dd if=/dev/dsk/bkroot of=/dev/dsk/root bs=128k
```

Note

You must configure bkroot and root with the same number of disk blocks that are in the IOS config.uni file.

6. Edit the configuration file on the IOS to point back to root (/), and reboot the system by using /dev/dsk/root as the root (/) device.

Alternative root (/) file system recovery procedure

B.6

Except for the value of ROOTDEV, this procedure is similar to the first. In this procedure, you must change ROOTDEV to `iroot`, rather than `bkroot`. When you are operating in `iroot`, you can use the `restore` command to restore root (/) from your most recent back-up tape.

If there is no backup root or `iroot` file system on disk, place the release tape into the QIC tape device and enter the `iboot` command. This boots an `iroot` file system in DDRAM (not disk). A `/sys/config.uni` file must be present to run this utility. After booting to single-user mode, run the `els_install` utility to re-install the 7.0 package. Because the root (/) file system on the release tape is generic, any customization, user validations, and so on are lost and must be re-created. It is recommended, therefore, that you keep `/bkroot` up-to-date as you make changes to the root (/) file system.

Console Terminal Settings [C]

This appendix describes the settings used to define your WYSE terminal settings.

Defining your console settings

C.1

To define your WYSE console settings, use the following terminal settings:

<u>Feature menu</u>	<u>Settings</u>
Display	Columns = 80 Lines = 25
Status line	Off
General	Terminal type = VT100
Communication	Baud = 19200 Data/stop = 8/1 Parity = NONE Handshakes = XON/XOFF1
Tabs	Default tabs = every 8 columns (tabs must be reset after every power-up)

DAS Fault-resilient Features [D]

Beginning with the release 8.9 of the CRAY Y-MP EL I/O subsystem software, two additional fault-resilient features for disk array subsystem controllers (DAS) are available. These features apply to DAS-1 and DAS-2 controllers. As part of the installation process, you must initialize DAS controllers on your CRAY Y-MP EL system.

The first feature automatically remaps bad sectors during system operation. If a drive exceeds the threshold for uncorrected read or write errors during system operation, the second feature allows your CRAY Y-MP EL system to switch automatically to an available spare DAS (hot spare).

You must initialize the threshold values and error counts when you install the CRAY Y-MP EL I/O subsystem software. After it is initialized, these threshold values and error counts can be viewed and, if necessary, changed as described in the subsections that follow.

DAS initialization

D.1

When installing the CRAY Y-MP EL I/O subsystem software, you must initialize the threshold values and error counts. You do not have to perform this initialization again unless you completely reinitialize the DAS controller. To initialize your DAS controller, use the command line that is appropriate for your configuration.

If your DAS controller is connected to IOS 0, enter the following:

```
dwconfig D00 /install/das_init.cfg
```

If your DAS controller is connected to IOS n , enter the following:

```
rcmd ios $n$  dwconfig D00 /install/das_init.cfg
```

Viewing threshold and error counts

D.2

To view or review the DAS threshold and error counts, enter the command line that is appropriate for your configuration.

If your DAS controller is connected to IOS 0, enter the following:

```
dstat d00
```

If your DAS controller is connected to IOS *n*, enter the following:

```
rcmd iosn dstat d00
```

```
IOS> dstat d00
```

```
Driver Strategy Numbers:
```

```
Number of driver calls made : 8933
Number of chained calls made : 277
Times requests were suspended: 4323
```

```
DAS  ctrl 0 bank 0 [D00], type = DDLAS
```

```
DAS Driver Numbers:
```

```
Total number of DAS requests      : 1442
Total number of cntrlr requests    : 1283
Current number of DAS requests     : 0
Outstanding controller requests    : 0
Average controller command depth   : 0
Number of requests waiting         : 0
Total requests which have waited  : 24
No. of times req's were stacked    : 217
No. of times max req's stacked     : 25
No. of times dma suspended         : 0
List of all tsb's                  : 0x32fe104
List of current tsb's              : 0x3307ac4
```

```
Bank 0 status 83ff, Read err threshold = 10, Write err threshold = 1
Substitute = none, Disable = none
```

	Drive status	Read errs	Write errs
drive 1	30	0	0
drive 2	30	0	0
drive 3	30	0	0
drive 4	30	0	0
drive 5	30	0	0
drive 6	30	0	0
drive 7	30	0	0
drive 8	30	0	0
drive 9	30	0	0
drive 10	30	0	0

```
IOS>
```

For further details about the `dwconfig` and `dstat` commands, see the *CRAY Y-MP EL System-specific Administrator Commands Reference Manual*, publication SR-2408.

Changing threshold values

D.3

If your site does not use hot-spare switching, set the threshold counts to 0. You may choose to use a different value, which is limited only by a maximum byte size of 255. If you intend to change the values, copy the `/install/das_init.cfg` text file to a new file name and edit it by using the `ed` command. After editing the new file, execute the appropriate `dwconfig` command to load the values from the file into the DAS controller.

Bad sector remapping

D.4

The DAS tracks each bad sector that it encounters. The bad sectors are transparently logged with address information in the DAS Growth Error table (GET) and an error packet is sent to UNICOS for system logging.

You can slip or remap the bad sector by issuing the `dformat` command. This merges GET with the DAS Raw Flaw table (RFT) information and clears GET. The DAS controller refers to this process as Format Function Level 5. For more information about `dformat`, see the *CRAY Y-MP EL System-specific Administrator Commands Reference Manual*, publication SR-2408.

DAS hot-spare switching

D.5

If a drive exceeds the threshold for uncorrected read or write errors during system operation, the system will switch automatically to a DAS hot spare. Parity reconstruction is used to reconstruct the hot spare with the data from the failing disk. This procedure takes approximately 45 minutes.

Threshold values and individual running error counts for each drive in the array are stored in an unused portion of the DAS controller memory. To provide an accurate long-term record of disk errors, this data is retained if your system is rebooted or powered down.

Note

The release software defaults are set for ten unrecovered reads per disk and one unrecovered write per disk; however, these thresholds can be changed for individual site preferences (see “Changing threshold values,” page 97).

Both UNICOS and IOS make error log entries when reconstruction begins and again when reconstruction is complete. If the reconstruction operation encounters uncorrectable errors on individual sectors, they are reported to the IOS console and the UNICOS error log. At this time, the system administrator must restore the lost files from the back-up files.

After a drive is switched out and the hot spare is in use, the system administrator must contact the CRAY Y-MP EL system local field engineer who will fix the failing drive. When appropriate, the repaired or replaced drive should be restored on your CRAY Y-MP EL system.

UNICOS Multilevel Security [E]

The UNICOS multilevel security (MLS) feature provides mechanisms to protect both system integrity and sensitive information. Design specifications for the UNICOS MLS feature were derived from the *Department of Defense Trusted Computer System Evaluation Criteria* (DoD 5200.28-STD). For more information on MLS, see *UNICOS System Administration*, publication SG-2113.

This appendix provides procedures for configuring MLS on your preinstalled UNICOS 7.0 system and for upgrading your MLS system to 7.0.

Configuring MLS on your pre-installed UNICOS 7.0 system

E.1

The following instructions describe how to configure the UNICOS MLS feature on your preinstalled UNICOS 7.0 system. These procedures may be executed at any time in single-user or multiuser mode. If the system is in single-user mode, make sure `/usr` and `/usr/src` are mounted.

For additional information on new MLS features for 7.0 and how to set up secure NFS, see the “UNICOS Multilevel Security” section of the *UNICOS 7.0 Release Overview for CRAY Y-MP EL Systems*, publication RO-5200 7.0.

1. Enter the UNICOS Installation and Configuration Menu System.

Enter the menu system by executing the following commands:

```
# cd /etc/install
# ./install
```

2. Enable MLS.

To enable MLS, set the following menu selection to ON, then **[TAB]** down to the Activate the configuration action and press **[RETURN]**:

<u>Menu</u>	<u>Option to select</u>
Configure System ==>	
Major software configuration ==>	
Multilevel Security (MLS)	ON
.	
.	
.	
Activate the Configuration...	

If you also want to administer your system security by using the menu system, enable menu system configuration by setting the following selection to YES:

<u>Menu</u>	<u>Option to select</u>
Configure System ==>	
Configurator automation options =>	
Multilevel security (MLS) configuration	YES
=>	

3. Import the MLS configuration files.

To verify that the installation menu files are up-to-date with the configuration files on your system, you must import your MLS configuration files.

Before importing the MLS configuration files for the first time, you must modify the import table to refer to the template `seclabs.c` file, as follows:

```
Utilities ==>
  Import utility ==>
    Import table ==>
```

Go to the `seclabs.c` entry in the Import Table and change the first argument from:

```
/usr/src/uts/cf/seclabs.c
```

to:

```
/usr/src/uts/cf.SN/seclabs.c
```

Then set the Import? selection to Yes.

Now go to the `hosts.nal` entry in the Import Table and set the Import? selection to YES.

Then proceed with the import as follows:

```
Configure system ==>
  Multilevel Security (MLS) Configuration ==>
    Import the security configuration ==>
```

After the import is complete, reset the value in the Import table back to `/usr/src/uts/cf.SN/seclabs.c`. This can be done by executing the following menu action:

```
Utilities ==>
  Import utility ==>
    Import option ==>
      Reload default import table ...
```

4. Set up the network access list (NAL).

If you plan to permit TCP/IP network access to your UNICOS system (for example, telnet or ftp), entries for the following Internet address(es) must appear in the network access list (NAL), as follows:

```
Local host (127.0.0.1)
Local network interfaces for all directly
connected networks
```

To set up entries in your `hosts.nal` file, go to the following submenus. The security option should be set to either NONE, BASIC, or CIPSO, depending on your security policy.

Menu

```
Configure system ==>
  Multilevel Security (MLS) Configuration ==>
    Network Security Options
      Network Access List (NAL) ==>
```

Options to select

Add a new entry by pressing n. Modify this new entry by pressing `RETURN` while the cursor is on the new entry just added. Fill in the following submenu:

<u>Menu</u>	<u>Options to select</u>
Network type	TCP
Host name / address or USCP station I.D.	xxxx
Minimum security level	0
Maximum security level	16
Minimum security compartments	0
Maximum security compartments	07777777777777777777777777777777
Receive?	YES
Send?	YES
Network trap?	NO
Security class	B1
Security option	NONE
Authority-in	
Authority-out	
Domain of interpretation	

5. Configure other MLS options.

Other MLS options can be configured at this time. Select and modify the submenus under the following menu:

```
Configure system ==>
  Multilevel Security (MLS) configuration ==>
```

The functions of the submenus are as follows:

- Modify the system console options:

```
System Options ==>
```

- Modify the Login / Password options:

```
Login / Password options ==>
```

- Modify your disk scrubbing policy:

```
Disk options ==>
```

- Modify the location of your security log, the name of your security log, or what is audited:

```
Security log file configuration ==>
```

- Define site-specific compartments or levels:

```
Site labels configuration ==>
```

6. Activate the security configuration.

After you have made your selections, you must activate the new configuration with the following menu:

```
Configure system ==>
  Multilevel Security (MLS) Configuration ==>
    Activate the security configuration ...
```

7. Label network interfaces and routes.

You must configure the security levels and compartments on the networking interfaces and TCP/IP static routes.

If you do not label these interfaces and routes, the authorized security levels and compartments of users who connect to the UNICOS system will be set to 0 and None. The steps are as follows:

- a. Import the current network configuration, as follows:

```
Configure system ==>
  Network configuration ==>
    General network configuration ==>
      Import general network configuration ...
```

- b. Label the network interfaces in the following submenu:

```
Configure system ==>
  Network configuration ==>
    General network configuration ==>
      Network interface configuration ==>
```

Under this submenu, you must define each of your network interfaces; for each interface, set the minimum and maximum level and compartments.

Your routes will be labeled automatically with the range of labels on your interfaces.

When you have completed your general network configuration, activate the new configuration:

```
Configure system ==>
  Network configuration ==>
    General network configuration ==>
      Activate general network configuration ...
```

8. Remove old TFMgmt compartments.

If your site has used the old TFMgmt style, you must upgrade to the new TFMgmt style (if you have not used the old TFMgmt style, proceed to step 9).

To remove reserved compartments, use the following menu:

```
Configure system ==>
  Multilevel security (MLS) configuration ==>
    Site labels configuration ==>
      Compartments
```

Typically, you will want to remove from this menu the six reserved compartments used by the `tsubcmd` and `udbcmd` commands: `secadm`, `sysadm`, `sysops`, `netadm`, `crayri`, and `unicos`. If, however, your site used any of these reserved compartments to protect sensitive data other than the files labeled by the `tsubcmd` and `udbcmd` commands, you should not remove the definitions for those compartments.

Any reserved compartments that you remove can be used for future compartment expansion. After a compartment definition has been removed, you can reference that compartment only by its octal representation.

Note

Before reusing any removed compartment bits, use the `spset` command to remove and relabel all file labels that reference the removed compartments (see the `spset(1)` man page for more information).

9. Build an MLS kernel and transfer it to the expander disk.

To link a new MLS kernel, enter the `Build/install` selection of the install tool and set the appropriate options, as follows.

After all options have been selected, select "Do the Build..." to build the new kernel, as follows:

<u>Menu</u>	<u>Option to select</u>
Build/install system ==>	
/usr/src reconfiguration files ==>	
Build action to take	install
Build object	all objects
Components to build	specific component
Major components selection ==>	
Specific component to build	uts
Do the build ...	
Restart the build ==>	
Review last build error summary ...	
Escape to a chroot shell ...	

Move the new kernel to the expander disk.

If you have not made a back-up copy of your current kernel, either make one now or transfer the kernel with a unique suffix.

If you want to make a back-up copy now, escape to the shell and execute the following commands:

```
# CONTROL-a (toggles to the IOS console)
IOS> mv sys/unicos.ymp sys/unicos.old
IOS> CONTROL-a (toggles to the UNICOS console)
```

Return to the install menus and press M to go to the main menu. Enter the Utilities menu and select file transfer utilities. Define where to transfer the new kernel and any other specific information needed as shown in the following example. When all options have been specified, select the menu selection Do the transfer to the expander.

<u>Menu</u>	<u>Option to select</u>
Utilities ==>	
File transfer utilities ==>	
Expander file transfers ==>	
Transfer UNICOS Kernel to the expander?	YES
Transfer UNICOS load map to the expander?	
Transfer CSL param file to the expander?	
Transfer IOS Kernel to the expander?	
Transfer diagnostics to the expander?	

MenuOption to select

Expander directory name	sys
Expander filename suffix	.ymp

Do the transfer to the expander ...
 Create an IOS deadstart tape ...
 Create an offline diag tape ...

You can now exit the install tool and continue with MLS configuration.

10. Set up the security log device file.

Before booting the new kernel, you need to make sure that the `/dev/slog` security log device file exists. If this device file does not exist, it must be created, as follows:

```
# /etc/mknod /dev/slog c 20 0
```

The `/dev/slog` file is a pseudo device in which all security events are recorded. If this device is not available before the system is booted, an error message from the security log daemon is displayed on the console.

11. Label necessary file systems.

Label all file systems that are to be mounted under the MLS kernel with the appropriate security labels for your site. You will want to label the root (`/`), `/usr`, `/usr/src`, and `/usr/home` file systems at a minimum.

If your site uses nonzero levels and compartments, use the following guidelines for labeling file systems:

- a. Label the root file system with all the security levels and compartments that are assigned to users on the MLS system. This is necessary because pipes are usually allocated through the root device. Also, tty devices change labels to match that of the user through the root device.
- b. Label the file systems in which NQS and mail files reside with all the security levels and compartments assigned to the users on the MLS system. The `/usr/tmp` and `/tmp` file systems must also be labeled with all the security levels and compartments.

Your site may need to specify a maximum security level and authorized compartment values that are nonzero. For example, if your site plans on using security levels 0 through 16 and compartment 04700, your root and other file systems must be labeled as follows:

```
# /etc/labelit -s -l 0 -u 16 -c 04700 /dev/dsk/root
# /etc/labelit -s -l 0 -u 16 -c 04700 /dev/dsk/usr_tmp
# /etc/labelit -s -l 0 -u 16 -c 04700 /dev/dsk/tmp
```

12. Shut down the system and boot the new MLS kernel.

Shut the system down to single-user mode, and reboot your new kernel by entering the following:

```
# cd /
# /etc/shutdown 0
# sync; sync; sync
# CONTROL-a (toggles to the IOS console)
```

Note

If you have transferred your kernel with a unique suffix, make a back-up copy of your old kernel, and rename the new kernel `unicos.ymp`.

```
IOS> mv sys/unicos.ymp sys/unicos.old
IOS> mv sys/unicos.mls sys/unicos.ymp
```

```
IOS> boot (Boot the new kernel)
```

13. Label trusted binaries and files.

The `tsubcmd` and `udbcmd` commands must be run to label the trusted binaries and files. When running these commands, you must use the `-l` option (label only). This will properly label the necessary MLS sensitive files without removing the files from their target locations. First mount the `/usr` file system, then run the `tsubcmd` and `udbcmd` commands as follows:

```
# /etc/mount /dev/dsk/usr /usr
# /etc/tsubcmd -l
# /etc/udbcmd -l
```

14. Create a security administrator account.

If you are performing an initial installation of UNICOS 7.0, or upgrading to UNICOS 7.0 MLS from a non-secure UNICOS 6.1 system, you must create an account in the UDB for system's security administrator, using either the `udbgcn(8)` or the `nu(8)` command.

A security administrator should be given the following UDB values, at a minimum:

- `valcat:secadm,sysfil:`
- `permits:install,reclsfy,suidgid:`
- `maxcls:16:`

In addition, if you want the security administrator to be able to reclassify user files, you should give the security administrator account all security levels and defined compartments for your system.

15. Label console devices.

For UNICOS 7.0, the system console device entries under the `/dev` directory must be labeled as follows:

Use the `-L` and `-C` options to the `/etc/getty` command to specify the level and compartment ranges for the system console device `/dev/console`.

To label your console device, edit the `/etc/inittab` file as follows:

```
# vi /etc/inittab
```

```
co::respawn:/etc/getty -L 0-16 -C 0-077 console console
```

16. Configure workstation access list.

The `spwal(8)` command creates workstation access list (WAL) entries for the UNICOS MLS system. Because the WAL is not required, you may configure it at a later time. If, however, you want to configure your WAL entries at this time, see the `spwal(8)` man page for more information.

17. Create CIPSO Security Label Translation table.

If the local networks at your site support the Commercial IP Security Option (CIPSO), you must create the security label translation tables, using the `spsomap(8)` command. The translation tables that `spsomap(8)` creates are read at boot time; therefore, you must reboot the system after creating (or modifying) the tables for the entries to take effect. See the `spsomap(8)` man page for more information.

18. Arrange for NAL initialization at system startup.

When the UNICOS 7.0 system enters multiuser mode, you must initialize the kernel's network access list (NAL) table by using the `spnal(8)` command. The default `/etc/rc` script supplied with UNICOS 7.0 does this for you. If you are not using the UNICOS `/etc/rc` script for your multiuser startup, you must arrange for the NAL initialization at system startup by including a line such as the following in your system start-up script:

```
/etc/spnal -k /etc/hosts.nal
```

See the `spnal(8)` man page for more information.

19. Enter multiuser mode.

You can now boot the system to multiuser mode by executing the following command:

```
# init 2
```

20. Define each user in the UDB.

At some point, the security administrator must define each user in the UDB. Proper definition of users in the UDB is critical to maintaining a trusted system. Each UDB entry defines the following:

- Active integrity class
- Authorized integrity class
- Active integrity category
- Authorized integrity categories
- Active compartments
- Authorized compartments
- Active security level
- Minimum security level
- Maximum security level
- Permissions

The security administrator can use the `nu(8)` or `udbgen(8)` command to create UDB entries or to modify security components of existing UDB entries. The `nu(8)` command creates and initializes a home directory for newly created accounts; the `udbgen(8)` command does not.

You must also set the proper security labels on the home directories of user accounts to have a nonzero default level or a nonnull default compartment set (or both). Although the `nu(8)` command creates a home directory and related files for a new account, it cannot set the directory to a nonnull label. You can set labels on these created directories and files by using the `spset(1)` command.

This work should be done in single-user mode. To do so, you must manually mount the file systems on which the `nu(8)` command will create the user login directories. Alternatively, you may bring the system up to multiuser mode without configuring the network, to initialize the security classifications of the users without permitting access to the system before proper security levels are established.

Upgrading your MLS system to 7.0

E.2

The following instructions describe how to complete the MLS portion of upgrading your system to 7.0.

For additional information on new MLS features for 7.0 and how to set up secure NFS see the “Multilevel Security” section of the *UNICOS 7.0 Release Overview for CRAY Y-MP EL Systems*, publication RO-5200 7.0.

At this point, you should have already booted a new 7.0 MLS kernel and be in single-user mode.

1. Label UNICOS 7.0 file systems correctly.

You must correctly label the following UNICOS 7.0 file systems for the UNICOS 7.0 MLS system to function properly:

- root (/)
- The file system(s) in which the following directories reside:

```
/usr/mail  
/usr/spool/NQS  
/usr/spool/USCP
```

- /tmp and /usr/tmp

You must label these file systems with all of the security levels and compartments that are assigned to the system's users. This is necessary for the root file system because pipes are usually allocated through the root device, and because tty devices, which reside on the root device, change labels to match those of a user connecting to the UNICOS 7.0 MLS system. Labeling is necessary for the other file systems because users create files in them.

2. Convert your UNICOS 6.1.6 format `/etc/hosts.nal` file to the new UNICOS 7.0 format. Enter the following commands to do this:

```
# cd /etc  
# /etc/spnal -c hosts.nal hosts.nal
```

3. Enter the menu system.

Use the following commands to enter the UNICOS Installation and Configuration Menu System:

```
# cd /etc/install  
# ./install
```

4. Import the MLS configuration files.

To verify that the installation menu files are up-to-date with the configuration files on your system, you must import your MLS configuration files.

Before importing the MLS configuration files for the first time, you must modify the import table to refer to the template `seclabs.c` file, as follows:

```
Utilities ==>
  Import utility ==>
    Import table ==>
```

Go to the `seclabs.c` entry and change its first argument from:

```
/usr/src/uts/cf.SN/seclabs.c
```

to:

```
/usr/src/uts/cf/seclabs.c
```

Then set the `Import?` selection to `Yes`.

Now go to the `hosts.nal` entry and set the `Import?` selection to `Yes`.

Then proceed with the import as follows:

```
Configure system ==>
  Multilevel Security (MLS) Configuration ==>
    Import the security configuration ==>
```

After the import is complete, reset the value in the `Import table` back to `/usr/src/uts/cf.SN/seclabs.c`. This can be done by executing the following menu action:

```
Utilities ==>
  Import utility ==>
    Import option ==>
      Reload default import table ...
```

5. Set up the network access list (NAL).

If you plan to permit TCP/IP network access to your UNICOS system (for example, `telnet` or `ftp`), entries for the following Internet address(es) must appear in the network access list (NAL):

- Local host (127.0.0.1)
- Local network interfaces for all directly connected networks

To set up entries in your `hosts.nal` file, go to the following submenus. The security option should be set to either none, basic, or cipso depending on your security policy.

Menu

Option to select

Configure system ==>

 Multilevel Security (MLS) Configuration ==>

 Network Security Options ==>

 Network Access List (NAL) ==>

Network type	TCP
Host name / address or USCP station I.D.	snxxxx
Minimum security level	0
Maximum security level	16
Minimum security compartments	0
Maximum security compartments	07777777777777777777777777777777
Receive?	YES
Send?	YES
Network trap?	NO
Security class	B1
Security option	NONE
Authority-in	
Authority-out	
Domain of interpretation	

6. Configuring other MLS options.

Other MLS options can be configured at this time. Select and modify the submenus under the following menu:

Configure system ==>

 Multilevel Security (MLS) configuration ==>

The functions of the submenus are as follows:

- Modify the system console options:

 System Options ==>

- Modify the Login / Password options:

 Login / Password options ==>

- Modify your disk scrubbing policy:

```
Disk options ==>
```

- Modify the location of your security log, the name of your security log, or what is audited:

```
Security log file configuration ==>
```

- Define site-specific compartments or levels:

```
Site labels configuration ==>
```

7. Activate the security configuration.

When you have made your selections, you must activate the new configuration, as follows:

```
Configure system ==>  
Multilevel Security (MLS) Configuration ==>  
Activate the security configuration ...
```

8. Label network interfaces and routes.

You must configure the security levels and compartments on the networking interfaces and TCP/IP static routes. If you do not label these interfaces and routes, the authorized security levels and compartments of users who connect to the UNICOS system will be set to 0 and none.

First, import the current network configuration, as follows:

```
Configure system ==>  
Network configuration ==>  
General network configuration ==>  
Import general network configuration ...
```

Next, label the network interfaces in the following submenu:

```
Configure system ==>  
Network configuration ==>  
General network configuration ==>  
Network interface configuration ==>
```

Under this submenu, you must define each of your network interfaces, and for each interface set the minimum and maximum level and compartments.

Your routes will be labeled automatically with the range of labels on your interfaces.

When you have completed your general network configuration, activate the new configuration, as follows:

```
Configure system ==>
  Network configuration ==>
    General network configuration ==>
      Activate general network configuration ...
```

9. Remove old TFMgmt compartments

If your site has used the old TFMgmt style, you must upgrade to the new TFMgmt style (if you have not used the old TFMgmt style, proceed to step 10).

To remove reserved compartments, use the following menu:

```
Configure system ==>
  Multilevel security (MLS) configuration ==>
    Site labels configuration ==>
      Compartments
```

Typically, you will want to remove from this menu the six reserved compartments used by the `tsubcmd` and `udbcmd` commands, which are `secadm`, `sysadm`, `sysops`, `netadm`, `crayri`, and `unicos`. If, however, your site used any of these reserved compartments to protect sensitive data other than the files labeled by the `tsubcmd` and `udbcmd` commands, you should not remove the definitions for those compartments.

Any reserved compartments that you remove can be used for future compartment expansion. After a compartment definition has been removed, you can reference that compartment only by its octal representation.

Note

Before reusing any removed compartment bits, use the `spset` command to remove and relabel all file labels that reference the removed compartments (see the `spset(1)` man page for more information).

10. You may now exit the UNICOS installation and configuration menu system.

11. Label console devices.

For UNICOS 7.0, the system console device entries under the `/dev` directory must be labeled as follows:

Use the `-L` and `-C` options to the `/etc/getty` command to specify the level and compartment ranges for the system console device `/dev/console`. You can also use the `-m` option to `/etc/getty` to indicate that the specified console device is a multilevel device.

To label your console device, edit the `/etc/inittab` file as follows:

```
# vi /etc/inittab

co::respawn:/etc/getty -L 0-16 -C 0-077 console console
```

12. Configure the WAL

The `spwal(8)` command creates workstation access list (WAL) entries for the UNICOS MLS system. Because the WAL is not required, you can configure it at a later time. If you want to configure your WAL entries at this time, see the `spwal(8)` man page for more information.

13. Create CIPSO Security Label Translation table.

If the local networks at your site support the Commercial IP Security Option (CIPSO), you must create the security label translation tables, using the `spsomap(8)` command. The translation tables that `spsomap(8)` creates are read at boot time; therefore, you must reboot the system after creating (or modifying) the tables for the entries to take effect. See the `spsomap(8)` man page for more information.

14. Arrange for NAL initialization at system startup

When the UNICOS 7.0 system enters multiuser mode, you must initialize the kernel's network access list (NAL) table by using the `spnal(8)` command. The default `/etc/rc` script supplied with UNICOS 7.0 does this for you. If you are not using the UNICOS `/etc/rc` script for your multiuser startup, you must arrange for the NAL initialization at system startup by including a line such as the following in your system start-up script:

```
/etc/spnal -k /etc/hosts.nal
```

See the `spnal(8)` man page for more information.

15. Enter multiuser mode.

You can now boot the system to multiuser mode by executing the following command:

```
# init 2
```


Configuring Network Interfaces [F]

UNICOS kernel configuration

F.1

This appendix describes the steps for configuring a network device so that it is accessible to the UNICOS kernel. This portion of the configuration is the same regardless of what protocol family will be using the device.

For CRAY Y-MP EL systems, this part of the configuration is usually done at the factory, so you should not need to alter the kernel configuration under most circumstances. However, you may have to change the kernel configuration if you add a new network interface or move a network interface from one IOS to another.

Note

The following instructions do not address configuring the HIPPI interface. Although the following sections provide information on parameters related to the HIPPI interface, they do not provide sufficient information to configure a HIPPI interface into UNICOS. Separate instruction will be provided when the HIPPI hardware is installed on your system.

The comm_info structure

F.1.1

Network interfaces are configured into the kernel when the kernel is linked. The `comm_info` structure in the `/usr/src/uts/c1/cf/conf.XXX.c` file (where `XXX` is the serial number of your system) is where the network interfaces are defined.

Figure 9, page 121, shows the fields and default values for the `comm_info` structure. The following paragraphs explain the meaning of each field.

<u>Field</u>	<u>Default value</u>
Driver	The name of the driver for the interface. Possible values are <code>COMM_EN_DRIVER</code> for Ethernet, <code>COMM_FDDI_DRIVER</code> for FDDI, <code>COMM_HY_DRIVER</code> for HYPERchannel, and <code>COMM_HSX_DRIVER</code> for HIPPI.
IOS	The IOS number in which the controller is physical located.
IOP	Not used on CRAY Y-MP EL systems. Set this field to zero for all entries.
Channel	<p>The channel determines which controller to use. This number is derived from a base number and an ordinal representing the controller's VMEbus address jumper setting.</p> <p>The base numbers are: HYPERchannel 0x00, HIPPI 0x00, Ethernet 0x10, and FDDI 0x20.</p> <p>To obtain the ordinal channel value for a network controller, look at the message generated during the IOS load. For example, the Ethernet driver will report "Ethernet: Controller 0 - detected," where 0 is the controller's ordinal channel value.</p> <p>Combine the base and ordinal values together to create the channel number. For this example, the channel would be 0x10. Refer to Figure 9 for more examples.</p>
Device type	Type of interface: <code>COMM_EN</code> for Ethernet, <code>COMM_FDDI</code> for FDDI, <code>COMM_N400</code> for NSC 400-series HYPERchannel adapters, and <code>COMM_HIPPI</code> for HIPPI.
Logical paths	Used only for HYPERchannel and HIPPI entries, in which case it is the number of logical paths available (set between 1 and 16). For all other types, set to 0.
Device-dependent	Only used for HIPPI entries. For HIPPI, set this field to <code>HXCF_HIPPI</code> . For all other types, set to zero.

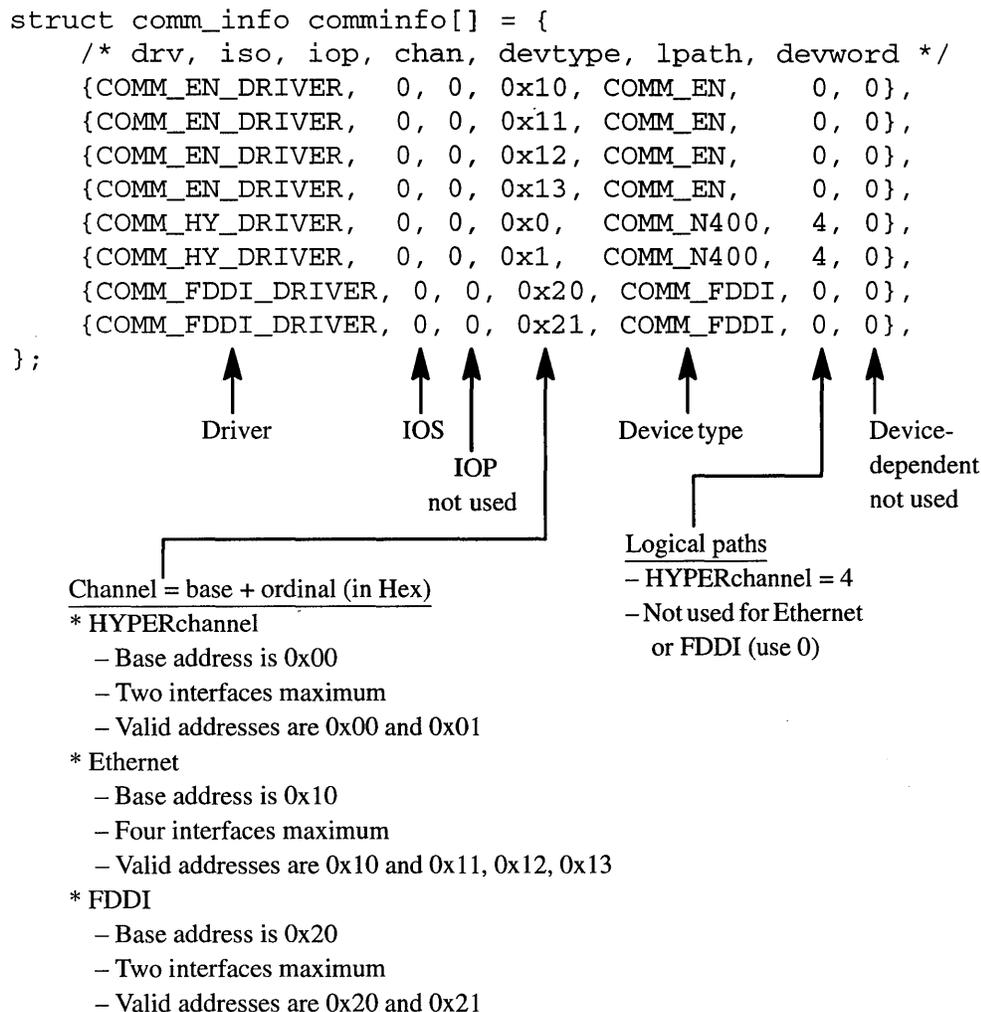


Figure 9. The comm_info structure

Determining the interface name

F.2

It is important that you know how the interface name is formed, because this name is referenced in the configuration files and commands that are used to assign the Internet address, map the hardware address, and enable the interface.

The interface name consists of two parts. The first part, the prefix of the interface name, is used to indicate the type of hardware (specifically, the type of driver). The second part, the number of the interface name, identifies the entry in the comm_info structure.

On CRAY Y-MP EL systems the following prefixes are used:

<u>Prefix</u>	<u>Type of hardware</u>
en	Ethernet interface
fddi	FDDI interface
hy	HYPERchannel interface
hsx	HIPPI interface
lo	Loopback interface

The interface number is determined by the ordinal position of the entry (relative to the other entries for the same type of driver) in the `comm_info` structure. For example, the first Ethernet interface entry in the `comm_info` is `en0`, the second Ethernet interface entry is `en1`. See the examples in Figure 10.

```
struct comm_info comminfo[] = {
    /* drv, iso, iop, chan, devtype, lpath, devword */
    {COMM_EN_DRIVER, 0, 0, 0x10, COMM_EN, 0, 0},      en0
    {COMM_EN_DRIVER, 0, 0, 0x11, COMM_EN, 0, 0},      en1
    {COMM_EN_DRIVER, 0, 0, 0x12, COMM_EN, 0, 0},      en2
    {COMM_EN_DRIVER, 0, 0, 0x13, COMM_EN, 0, 0},      en3
    {COMM_HY_DRIVER, 0, 0, 0x0,  COMM_N400, 4, 0},      hy0
    {COMM_HY_DRIVER, 0, 0, 0x1,  COMM_N400, 4, 0},      hy1
    {COMM_FDDI_DRIVER, 0, 0, 0x20, COMM_FDDI, 0, 0},    fddi0
    {COMM_FDDI_DRIVER, 0, 0, 0x21, COMM_FDDI, 0, 0},    fddi1
};
```

Examples 

Figure 10. Examples of interface entry numbers

UNICOS Configuration Menu System [G]

The UNICOS configuration menu system (*the menu system*) provides a menu-driven interface for configuring and maintaining UNICOS on CRAY Y-MP EL systems.

Note

You cannot use the menu system for installing UNICOS on a CRAY Y-MP EL system, but once UNICOS is successfully installed, you can use the menu system for configuration and system maintenance.

The menu system includes the following functions:

- Controls media access
- Converts old structures to new ones
- Imports existing UNICOS configuration information into the menu system configuration database

The menu system also provides ease-of-use features such as context-sensitive input checking and online help, among others.

Menu screens

G.1

The menu system divides your screen into several areas, displaying a different type of information in each area. Figure 10 shows the way a screen is divided. After the figure, each of the screen information areas is described.

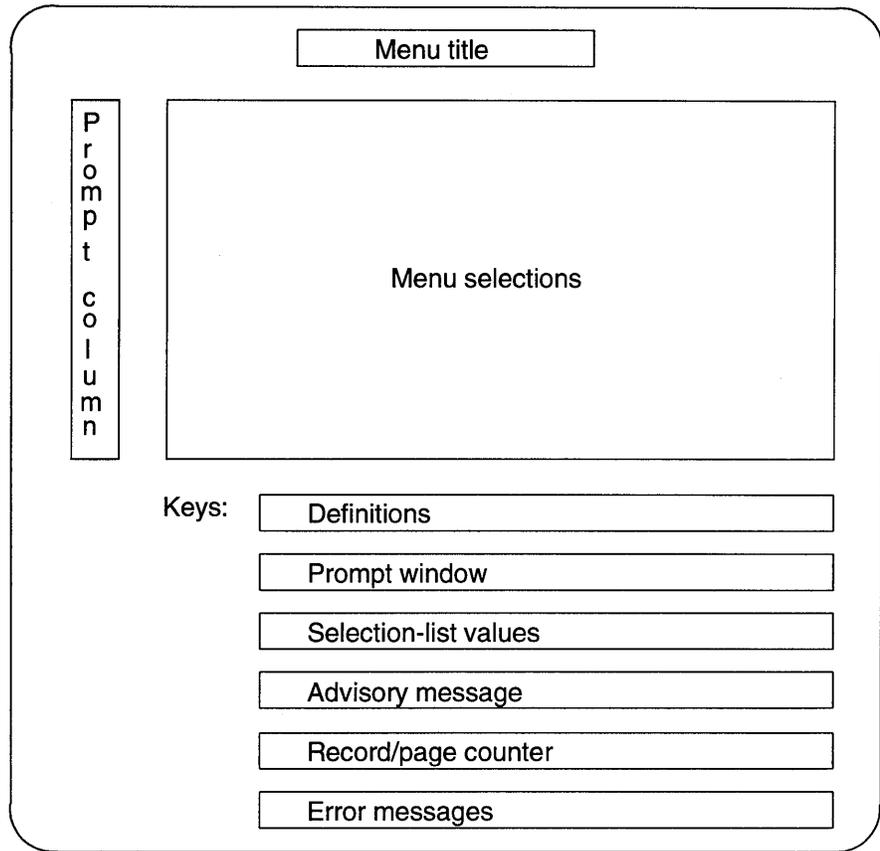


Figure 10. Menu system screen information areas

The screen information areas shown in Figure 10 contain the following types of information:

<u>Area</u>	<u>Contents</u>
Menu title	Contains the title of this menu.
Prompt column	Contains the prompt, showing you the menu item ready to be selected and its type.
Menu selections	Displays the items in this menu, one per line.
Key definitions	Shows available keys for the current menu or selection. (For more information about these keys see the subsections describing menu system keys on pages 128 through 132.)

<u>Area</u>	<u>Contents</u>
Prompt window	Displays prompts for information you must enter.
Selection-list values	Displays one of the following when you are in input mode on a selection line: <ul style="list-style-type: none"> • A list of the permissible values if the number is limited • The message <code>Any non-white-space input value allowed if only strings without spaces are allowed</code> (usually for UNICOS path names, file names, and so on) • The message <code>Any input value allowed if the number of permissible values is very large</code>
Advisory message	Displays informational messages that the menu system issues.
Record/page counter	Shows the record to which the prompt currently points and the total number of records in a menu. It is displayed only when you are in a menu of record lines. This line also shows the page number and total number of pages, when you are in a multipage menu.
Error messages	Displays error messages that the menu system issues.

Terminology

G.2

You should become familiar with the following terms when reading this section:

<u>Term</u>	<u>Definition</u>
<i>action</i>	A menu item that executes a program. The A-> prompt indicates actions.
<i>form list</i>	A screen representation of the record lines that make up a configuration file.

<u>Term</u>	<u>Definition</u>
<i>menu</i>	A menu item that leads to another menu, one level down in the menu tree structure. (See Figure 11, page 135, for a diagram of the menu tree.) The M-> prompt indicates menus.
<i>record line</i>	One line of a configuration file, consisting of one or more fields of information.
<i>selection</i>	A menu item that is a configuration parameter that has an associated value. The S-> prompt indicates selections.
<i>selection list</i>	A menu that contains only selection items.

Menu selections

G.3

The main body of the screen (Menu selections in Figure 10, page 124) displays one menu at a time from the menu hierarchy. It is identified by the menu title, which is centered at the top of the screen.

A menu is divided into multiple pages (screens) if all of the entries cannot fit on one screen. The + and - keys, defined in "General menu keys," page 128, let users scroll through multipage menus.

The menus in the menu hierarchy are organized by product in a top-down fashion. Items contained in most menus are listed according to the order in which you will likely use them: first submenu, then selections, and then actions at the bottom of the menu.

Two distinct menu types exist. One permits rapid navigation of the menu system, the other permits modification of configuration files. The first type of menu organizes its items into three separate groups: menus, selections, and actions. This type of menu lets users navigate the menu hierarchy quickly by having all of the submenu definitions near the top of each menu. After a user locates a specific menu, the user can modify the selection values displayed or execute an action to perform an installation task. The prompts M->, S->, or A-> identify this type of menu.

The second type of menu contains only form-list items. This type of menu lets users manipulate individual configuration-file entries (called *record lines*). The E-> prompt identifies this type of menu.

Menu prompts

G.4

The prompts of the menu system indicate the type of a menu item and what you can do with it. The following are the available prompts and their meanings:

<u>Prompt</u>	<u>Meaning</u>
A->	Indicates an item of type <i>action</i> . The line specifies the action taken when you press <code>RETURN</code> on this line. Items that end with ... are of type <i>action</i> .
E->	Indicates a record line of a configuration file (one line of a form-list screen). Pressing <code>RETURN</code> on this line places you in a selection list for the record line, where you can change the values of the fields.
M->	Indicates an item of type <i>menu</i> . Pressing <code>RETURN</code> on this line places you in the indicated menu one level lower in the menu hierarchy. (Figure 11, page 135, shows the top three levels of menus.) Menu selections that end with ==> are of type <i>menu</i> .
N/A	Indicates a menu item that is either not applicable for the current step or not enabled.
S->	Indicates an item of type <i>selection</i> . The line is a configuration parameter that requires a value. Pressing <code>RETURN</code> at this prompt places the cursor in the value field, in input mode (see "Input keys," page 130).

The following is a common sequence of actions (based on these prompts as identifiers):

1. Go to a menu (press `RETURN` at an M-> prompt)
2. Change the value of a selection (press `RETURN` at an S-> prompt)
3. Perform an action (press `RETURN` at an A-> prompt)

General menu keys

G.5

The menu system supports some general-use keys that are available (active) in most screens. These keys (either uppercase or lowercase) perform the following actions:

<u>Key</u>	<u>Action</u>
<code>RETURN</code>	<p>Depending on where you are located, performs the following functions:</p> <ul style="list-style-type: none"> • At an A-> prompt, it initiates the action. (Usually this means invoking a program.) • At an M-> prompt, it moves you to the menu listed. • At an S-> prompt, it moves the cursor to the value field and puts you in input mode to enter a value. • In input mode, it saves the value entered and returns the cursor to the prompt column at the left of the screen (if the value is valid; otherwise, it beeps and issues a message).
<code>H</code>	<p>Displays online help for the menu screen in which you are currently working.</p> <p>This key is displayed only when online help information is available for the current screen.</p>
<code>V</code>	<p>Accesses either the Docview online help system or online man pages (man(1) command).</p>
<code>?</code>	<p>Shows a list of the valid inputs you can enter for menu items of type selection.</p>
<code>W</code>	<p>Displays the path through the menu hierarchy that you have traversed. This helps you to remember where a particular menu is located in the menu hierarchy. You can use this key to determine the context of a menu accessed from two or more paths.</p>
<code>SPACE</code>	<p>Toggles through a list of predefined values for a selection if such a list exists (when the cursor is in the prompt column at the left of a menu). The <code>⊞</code> key also does this.</p>
<code>Q</code>	<p>Quits the menu system after asking you whether this is really what you want to do.</p>
<code>CONTROL-L</code>	<p>Redraws the screen.</p>

<u>Key</u>	<u>Action</u>
[!]	Escapes to a shell. The type of shell is specified in the Preferences menu.
[CONTROL-?]	Displays menu commands that are currently available and valid selections if the current menu item is a selection. AMPEX terminals use [CONTROL-SHIFT] (Control-shift-underscore).
[<]	Selects previous selection-list value. (opposite of [>])

Movement keys

G.6

The menu system provides the following keys for you to use in moving around the menu hierarchy and within individual menus. (See Figure 11, page 135, which shows the first three levels of menus.)

<u>Key</u>	<u>Movement</u>
[M]	Returns you to the initial main menu screen.
[E]	Returns you to the previous menu, one level up the menu hierarchy.
[RETURN]	At the M-> prompt, moves you one level down the menu hierarchy.
[+]	Moves you to the next screen of a multiscreen menu. This is displayed only when the menu is more than one screen.
[-]	Moves you to the previous screen of a multiscreen menu. This is displayed only when the menu is more than one screen.
[TAB]	Moves you down to the next line of the menu. If you are on the last item in the menu, this wraps to the top. You can also use the [j] key (as in the vi editor) or [CONTROL-I] to perform this function.
[BACKSPACE]	Moves you up to the previous line of the menu. If you are on the first item in the menu, this wraps to the bottom. You can also use the lowercase [k] key (as in the vi editor) or [CONTROL-H] to perform this function.

Input keys

G.7

To change the value of a selection item in a menu, use one of the input keys. These keys are set to emulate the functions of one of the text editors `vi` or `emacs`, which you choose in the Preferences menu. Table 6 describes the available keys.

Table 6. Input keys

vi key	emacs key	Action
<code>^</code> or <code>[O]</code>	<code>CONTROL-A</code>	Moves cursor to beginning of input buffer.
<code>h</code> or <code>CONTROL-H</code>	<code>CONTROL-B</code>	Moves cursor backward one character (nondestructive).
<code>x</code>	<code>CONTROL-D</code>	Deletes the character on which the cursor is located.
<code>\$</code>	<code>CONTROL-E</code>	Moves cursor to end of input buffer.
<code>l</code> or <code>SPACE</code>	<code>CONTROL-F</code>	Moves cursor forward one character (nondestructive).
<code>X</code>	<code>CONTROL-H</code> or <code>DELETE</code>	Deletes previous character (to left of cursor).
<code>D</code>	<code>CONTROL-K</code>	Deletes from current cursor location to end-of-line.
<code>CONTROL-U</code>	<code>CONTROL-U</code>	Deletes entire input line.
<code>TAB</code> or <code>></code>	<code>TAB</code>	Gets next selection value (toggle). <code>CONTROL-I</code> also performs this action.
<code><</code>		Gets previous selection value (toggle).
<code>i</code>		Inserts new text to left of cursor.
<code>a</code>		Appends new text to right of cursor.
<code>A</code>		Appends new text to end of input buffer.
<code>I</code>		Inserts new text at beginning of input buffer.
<code>s</code>		Replaces the character under the cursor with input text until <code>ESCAPE</code> .
<code>S</code> or <code>CC</code>		Replaces the entire line with input text until <code>ESCAPE</code> .
<code>C</code> or <code>c\$</code>		Replaces from cursor position to end-of-line with input text until <code>ESCAPE</code> .
<code>r</code>		Replaces character on which cursor is located.
<code>R</code>		Replaces existing characters by overstriking them; ends with <code>ESCAPE</code> .
<code>RETURN</code>	<code>RETURN</code>	Verifies current input and accepts it, if valid.
<code>U</code>		Undoes all changes, restoring original value.
<code>u</code>		Undoes last editing change.

Table 6. Input keys (continued)

vi key	emacs key	Action
<code>CONTROL-L</code>	<code>CONTROL-L</code>	Redraws screen.
<code>CONTROL-?</code>	<code>CONTROL-?</code>	Displays active keys.
<code>DELETE</code>		Moves cursor backward one character (nondestructive). Used only when in insert, append, or replace mode.
<code>ESCAPE</code>		Terminates input, append, or replace modes, returning you to command mode.

Accelerator keys

G.8

The keys described in this subsection provide time-saving shortcuts for moving around in the menu hierarchy. As a handy reminder, the `CONTROL-?` key displays the accelerator keys you have already defined.

<u>Key</u>	<u>Description</u>
------------	--------------------

<code>A</code>	Assigns the current location in the menu hierarchy to one of the keys <code>1</code> through <code>9</code> .
----------------	---

<code>ESCAPE [n]</code>	Traverses the menu system to get to the menu defined by assign key <i>n</i> . If you specify 0 for <i>n</i> , the previous menu is displayed, allowing you to toggle between two menu locations anywhere in the hierarchy of menus.
-------------------------	---

Form-list keys

G.9

When you are in a form list (indicated by the `E->` prompt), you can search for, add, delete, copy, and move lines in the list. The form-list editing keys are as follows:

<u>Key</u>	<u>Action</u>
------------	---------------

<code>D</code>	Deletes the current line.
----------------	---------------------------

<code>n</code>	Creates a new line, with default values in the fields, after the current line.
----------------	--

<code>N</code>	Creates a new line, with default values in the fields, before the current line.
----------------	---

<u>Key</u>	<u>Action</u>
<input type="checkbox"/> p	Puts a yanked or deleted line after the current line.
<input type="checkbox"/> P	Puts a yanked or deleted line before the current line.
<input type="checkbox"/> s	Requests a string or regular expression and searches the form list for it. You can also use <input type="checkbox"/> / for this (as in the vi editor).
<input type="checkbox"/> S	Repeats the last search.
<input type="checkbox"/> U	Edits the current form list by using the editor specified in the preference menu.
	Use this feature with caution; do not alter record fields that should not be modified.
<input type="checkbox"/> Y	Copies the current line into a buffer for later use (like the vi editor "yank" (y) command).
<input type="checkbox"/> ?	Searches backward for specified expression (opposite of <input type="checkbox"/> /).
<input type="checkbox"/> ^	Displays the first column of the current form list. You can also use <input type="checkbox"/> 0.
<input type="checkbox"/> \$	Displays the last column of the current form list.
<input type="checkbox"/> <	Displays the previous column of the current form list.
<input type="checkbox"/> >	Displays the next column of the current form list.

Record-line keys

G.10

Each line of a form-list screen is called a *record line*. A record line is a line from a file (usually a configuration file) that is being parsed by the menu system for display (but that cannot be edited). With the cursor on a record line, pressing RETURN moves you to a selection list, which is an editable list of all fields that make up a record line.

In a selection list, you can use the following keys to change a fields' value:

<u>Key</u>	<u>Action</u>
Input keys	You can use all input keys described in "Input keys," page 130.
F	Moves you forward one record line so that you do not have to return to the form-list screen, one menu level up. The F key also performs this function.
B	Moves you backward one record line so that you do not have to return to the form-list screen, one menu level up. The B key also performs this function.

Log file

G.11

The menu system keeps a log file of configuration actions, including any errors or problems. Use the following menu sequence to examine this file:

```
UNICOS 7.0 Installation / Configuration Menu System
Utilities ==>
Inspect the installation log...
```

To examine the log file outside of the menu system, look at the `install.log` file in the installation directory.

Entering numbers

G.12

The menu system recognizes several different types of numbers. Table 7 shows the correct format you must use when entering numbers.

Table 7. Numeric types in the menu system

Number type	Description	Examples
Floating point	Starts with an optional minus sign, followed by one or more digits [0–9] appearing on either side or both sides of the decimal.	-.3 6. 3.45
Hexadecimal	Starts with 0X or 0x, followed by one or more digits [0–9] or letters [a–f A–F].	0X7A 0x35 0xffff
Octal	Starts with a 0, followed by one or more digits [0–7]. Octal 0 is represented by two consecutive 0's.	045 07777 00
Integer	Starts with an optional minus sign, followed by one or more digits [0–9].	-456 0 6999

Signal processing

G.13

The menu system traps the UNICOS SIGQUIT signal, gracefully exiting. It ignores UNICOS signals SIGINT and SIGTERM.

The SIGINT, SIGQUIT, and SIGTERM signals are reset to SIG_DFL when a program is executed from inmenu, letting that program decide whether certain signals should be ignored.

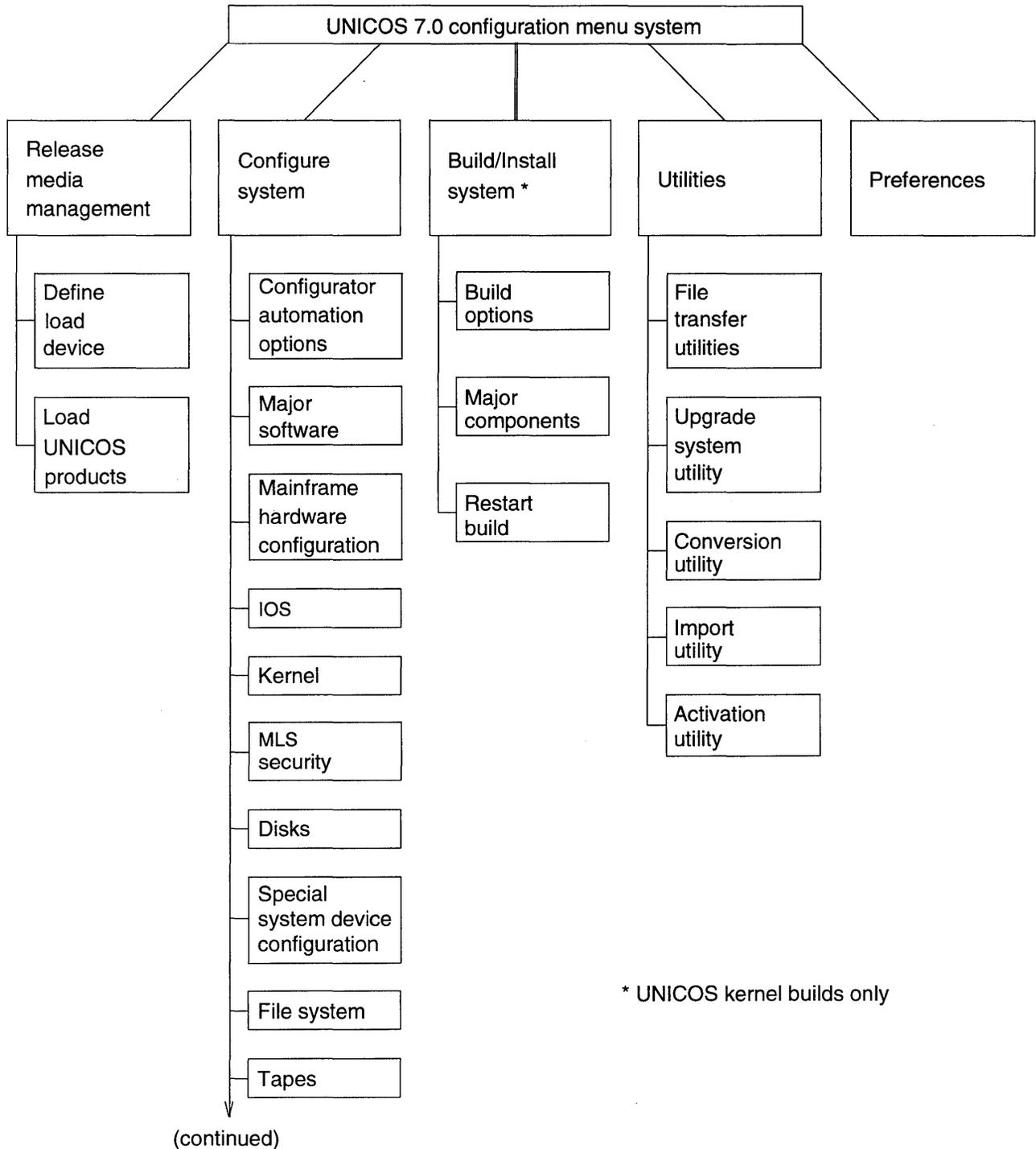
Note

The menu system traps the SIGQUIT signal, commencing an orderly shutdown and immediate exit, with an error code of 9 and the message Program terminated by the user entering a SIGQUIT signal.

Top three menu levels

G.14

To help you visualize your position when using the menu system, Figure 11 shows the top three levels of the UNICOS installation and configuration menu system.



* UNICOS kernel builds only

Figure 11. Top three levels of the menu system

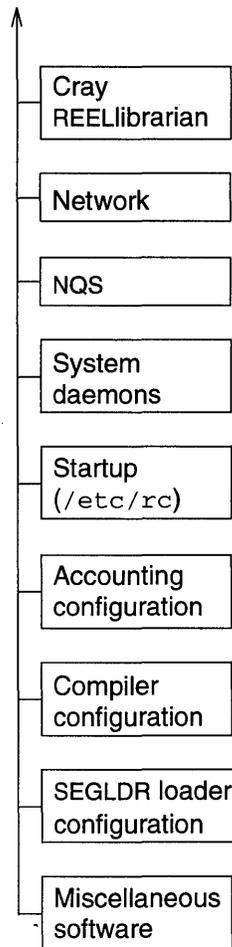


Figure 11. Top three levels of the menu system (continued)

Appendix for the
*UNICOS Installation Guide for
CRAY Y-MP EL Computer Systems*
SG-5201 7.0

The attached appendix H, "Disk Flaw Handling," was new with the UNICOS 7.0.4.2 update package. If you did not receive the UNICOS 7.0.4.2 update package, please insert this appendix into your copy of the *UNICOS Installation Guide for CRAY Y-MP EL Computer Systems*, SG-5201 7.0.

Disk Flaw Handling [H]

This appendix describes how to prepare ESDI and IPI drives for using the new disk flaw handling utilities and shows examples of using the `dsurf` and `dslip` utilities for disk flaw handling.

The new disk flaw handling feature makes flaw management much simpler than was possible using previous releases. After you have prepared the drives for disk flaw handling, the software will automatically save all flaw information. This flaw information can then be used on subsequent formats (if desired), so that no flawing information is lost.

The following changes were made to support the new disk flaw handling capability:

- A `dsurf` utility was added to provide a more flexible method of performing disk surface analysis. It can perform read-only tests or pattern write and read tests.
- The `dformat` utility was updated to use the saved flaw information. New options are `-l level`, `-s serial number`, and `-f file`.
- The `dformat` utility no longer performs a `dverify` for IPI drives.
- The `dslip` utility was updated; the new `dslip C:` syntax allows blocks to be slipped on a IOS SCSI disk.
- The `dflawr` utility has a new `-r` option that will read the sector IDs on IPI and ESDI drives to report the sectors and tracks that have actually been slipped and mapped on the drive. Additional new options are `-f file`, `-l`, and `-s`.
- The `dflaww` utility has the new options `dflaww E` (for ESDI disks), `dflaww I` (for IPI disks), and `-f file`.

The new `dsurf` and revised `dformat`, `dslip`, `dflawr`, and `dflaww` man pages are provided online only; a printed version of these man pages is not provided.

The following terms are used in this appendix:

<u>Term</u>	<u>Description</u>
Growth Error Table (GET)	This table shows all known flaws on the drive. For the IPI and ESDI disks, this table includes the OEM defect list, along with all flaws slipped and mapped by the user.
OEM defect list	This list shows all flaws found by the drive manufacturer; this list is permanently stored on the disk.

Preparing disks for flaw handling

H.1

ESDI disks (DD-2 and DD-3)

H.1.1

This subsection describes how to prepare ESDI and IPI drives for using the new disk flaw handling utilities.

To prepare ESDI drives for using the new disk flaw handling utilities, you must do one of the following:

- Format the drive by using `dformat` and the `-l 2` option, as follows:

```
dformat E00 -l 2 -s <serial number of drive>
```

- Run `dflawr` by using the `-r` option and write the GET read from the drive to a file, as follows:

```
dflawr E00 -r -s <serial number>
```

Then run `dflaww` to write the GET to the disk, as follows:

```
dflaww E00
```

Note

When running the `dflawr` utility with the `-r` option on ESDI drives, the flaw block numbers found in the GET list may not always match the OEM list.

IPI disks (DD-4)**H.1.2**

To prepare IPI drives for using the new disk flaw handling utilities, you must do one of the following:

- Format the drive by using `dformat` and the `-l 2` option, as follows:

```
dformat I00 -l 2 -s <serial number of drive>
```

- Run `dflawr` by using the `-r` option, and write the GET read from the drive to a file, as follows:

```
dflawr I00 -r -s <serial number>
```

Then run `dflaww` to write the GET to the disk, as follows:

```
dflaww I00
```

Examples**H.2**

This subsection shows examples of using the `dsurf` and `dslip` utilities for disk flaw handling.

dsurf utility**H.2.1**

The `dsurf` utility has the following syntax:

```
dsurf C: [-adfirv] [-n blocks] [-p passes] [-s start]
```

```
dsurf Ecd [-adfirvw] [-l level] [-n blocks] [-p passes]
[-s start] [-t count]
```

```
dsurf Icd [-adfirvw] [-l level] [-n blocks] [-p passes]
[-s start] [-t count]
```

<u>Option</u>	<u>Description</u>
-a	Ask before flawing; default is to flaw silently.
-d	Debug on; errors will not be flawed.
-f	Run test until one pass completes without an error.
-i	Inhibit recheck on flawed errors.

<u>Option</u>	<u>Description</u>
-l	Test level: 0 Read Test (Default) 1 Eight-pattern write/read 2 Four-pattern random write/read
-n	Number of blocks to test; default is entire disk.
-p	Number of passes to run; default is 1.
-r	Do not flaw errors; default is to flaw.
-s	Start block; default is 0.
-t	Read/write I/O size in sectors; default is one track.
-v	Verbose.
-w	Allow writing without prompting; default is to prompt before writing.

The following example performs a read of the entire ESDI disk, slipping/mapping any bad sectors found:

```
IOS>dsurf E00
```

The following example performs an eight-pattern write/read on the entire IPI disk and runs until no errors are found in a pass:

```
IOS>dsurf I00 -vf1 1
```

The following example performs a read-only test starting at block 32500 and reading until the end of the disk. The test does five passes and prompts the user to verify before slipping/mapping:

```
IOS>dsurf E00 -avp 5 -s 32500
```

The following example performs a read-only test of the entire IOS SCSI disk, slipping/mapping any defects found:

```
IOS>dsurf C:
```

dslip utility H.2.2

The dslip utility has the following syntax:

```
dslip C: sector
```

```
dslip Ecd sector
```

```
dslip Icd sector
```

The following example slips block 32456 on the IPI drive:

```
IOS>dslip I01 32456
```

The following example slips block 3456 on the IOS SCSI disk:

```
IOS>dslip C: 3456
```

Note

The block number given by the `dflawr` utility for the IOS SCSI disk is a physical block number, which may not be the same as the logical block number given to the `dslip` utility.

Reader's Comment Form

UNICOS Installation Guide for CRAY Y-MP EL Computer Systems

SG-5201 7.0

Your reactions to this manual will help us provide you with better documentation. Please take a moment to complete the following items, and use the blank space for additional comments.

List the operating systems and programming languages you have used and the years of experience with each.

Your experience with Cray Research computer systems: ____0-1 year ____1-5 year ____5+years

How did you use this manual: ____in a class ____as a tutorial or introduction ____as a procedural guide
____as a reference ____for troubleshooting ____other

Please rate this manual on the following criteria:

	Excellent			Poor
Accuracy	4	3	2	1
Appropriateness (correct technical level)	4	3	2	1
Accessibility (ease of finding information)	4	3	2	1
Physical qualities (binding, printing, illustrations)	4	3	2	1
Terminology (correct, consistent, and clear)	4	3	2	1
Number of examples	4	3	2	1
Quality of examples	4	3	2	1
Index	4	3	2	1

Please use the space below for your comments about this manual. Please include general comments about the usefulness of this manual. If you have discovered inaccuracies or omissions, please specify the number of the page on which the problem occurred.

Name _____
Title _____
Company _____
Telephone _____
Today's date _____

Address _____
City _____
State/Country _____
Zip code _____
Electronic mail address _____

Cut along this line

Fold



NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

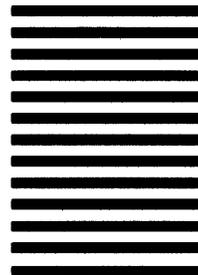
BUSINESS REPLY MAIL

FIRST CLASS MAIL PERMIT NO. 6184 ST. PAUL, MN

POSTAGE WILL BE PAID BY ADDRESSEE



**ATTN: Software Information Services
655 LONE OAK DR BLDG F
EAGAN MN 55121-9957**



Fold