ULTRIX and UWS Version 4.3

digital

Release Notes

Part Number: AA–ME85H–TE

ULTRIX and UWS Version 4.3

Release Notes

Order Number: AA-ME85H-TE

August 1992

Product Version: ULTRIX and ULTRIX Worksystem Software (UWS) Version 4.3

This manual lists new features and changes to the ULTRIX and UWS products. It also discusses product software and documentation problems.

digital equipment corporation
Maynard, Massachusetts

Contents

About This Manual

1 Installation Notes

3 ULTRIX Software Notes

4 ULTRIX Worksystem Software Notes

5 Layered Products Notes

6 Documentation Notes

7 New Features and Changes

A Problems Resolved Since Last Release

B FDDI Support in ULTRIX

Figures

Tables

About This Manual

This manual lists the major new features and changes to the software and documentation for the ULTRIX and ULTRIX Worksystem Software Version 4.3 products. It also describes problems in the software and documentation that were discovered too late to document elsewhere.

Read these release notes before you install the ULTRIX and ULTRIX Worksystem Software (UWS) products.

If you discover errors, omissions, or inaccuracies as you use the software and documentation, submit a Software Performance Report (SPR).

Audience

This document is written for people responsible for installing, managing, and maintaining the ULTRIX and UWS system and its documentation. Programmers and other users of the ULTRIX and UWS facilities will find information in these release notes that affects their work as well.

Organization

This document contains seven chapters and one appendix:

Chapter 1	Discusses the general installation of the software and provides workarounds to software and hardware problems, if workarounds exist.
Chapter 2	Discusses workarounds for software and hardware problems for specific processors.
Chapter 3	Discusses problems with the ULTRIX components and provides workarounds to these problems, if workarounds exist.
Chapter 4	Discusses problems with the UWS components and provides workarounds to these problems, if workarounds exist.
Chapter 5	Discusses problems with layered products and provides workarounds to these problems, if workarounds exist.
Chapter 6	Discusses problems with the ULTRIX and UWS documentation and provides corrections to these problems.
Chapter 7	Discusses the new and changed software and documentation features of this release.
Appendix A	Lists the software problems that were in previous versions of the products but have been fixed in ULTRIX and UWS Version 4.3.
Appendix B	Discusses how the FDDI interface to ULTRIX is supported.

Related Documentation

You should have the ULTRIX and UWS documentation kit and your hardware documentation.

Aside from this document, the four documents most likely to help you get started are:

- *ULTRIX and UWS Version 4.3 Software Product Description*
- *Guide to Installing ULTRIX*
- *Guide to System and Network Setup*
- *Guide to Sharing Software on a Local Area Network*

Conventions

`%` `$`	A percent sign represents the C shell system prompt. A dollar sign represents the system prompt for the Bourne and Korn shells.
`#`	A number sign represents the superuser prompt.
`% cat`	Boldface type in interactive examples indicates typed user input.
file	Italic (slanted) type indicates variable values, placeholders, and function argument names.
. . .	In syntax definitions, a horizontal ellipsis indicates that the preceding item can be repeated one or more times.
`cat`(1)	A cross-reference to a reference page includes the appropriate section number in parentheses. For example, `cat`(1) indicates that you can find information on the `cat` command in Section 1 of the reference pages.
Ctrl/*x*	This symbol indicates that you hold down the first named key while pressing the key or mouse button that follows the slash. In examples, this key combination is enclosed in a box (for example, Ctrl/C).

Installation Notes 1

This chapter discusses issues and known problems with the installation procedure and, when possible, provides solutions or workarounds to the problems. Read this chapter before you install the ULTRIX and UWS software.

For additional installation notes specific to individual processors, see Chapter 2.

The notes in this chapter cover the following topics:

- Hardware
- Booting
- General Installation
- Configuration

1.1 Hardware

The following sections discuss hardware and peripheral devices.

1.1.1 Procedure to Change ID Strings for TSZ07 Units

A vendor identity string change in the TSZ07 may cause some early units not to be recognized by the ULTRIX operating system. This section describes how to reset the Vendor ID and the Device ID strings associated with the earlier versions of the TSZ07.

To reset the Vendor ID, use the following steps. Note that you must take no longer than three seconds to perform each step from 3 through 7. If you do not perform these steps in the allotted time, you must begin the procedure from step 1.

1. Turn on both the AC Power Switch (rear panel) and the DC Power Switch (front panel).
2. If the ONLINE indicator is on, press the ONLINE switch (3) to turn the indicator off.
3. Press the WRT EN / TEST switch (4).
4. Press the DENSITY SELECT switch (5) and release, and then press the DENSITY SELECT switch (5) again.
5. Press the WRT EN / TEST switch (4).
6. Press the UNLOAD switch (2).
7. Press the DENSITY SELECT switch (5). The service aid now starts with the current vendor ID being displayed on the alphanumeric display and a cursor is placed on the left character space. The front panel switches function as follows:

- 1 – Increments the selected character up the ASCII character set.
- 2 – Decrements the selected character down the ASCII character set.
- 3 – Moves the cursor one position to the right to select a character.
- 4 – Exits service aid without saving changed characters.
- 4 and 5 – Pressing both switches simultaneously saves new characters and exits the service aid; the message SAVING appears.
- 5 – No function.

8. If the Vendor ID is DEC, exit the service aid. If the Vendor ID is CIPHER, change it to DEC, and save the change before you exit the service aid.

To reset the Device ID, use the following steps. Note that you must take no longer than three seconds to perform each step from 3 through 7. If you do not perform these steps in the allotted time, you must begin the procedure from step 1.

1. Turn on both the AC Power Switch (rear of panel) and the DC Power Switch (front panel).
2. If the ONLINE indicator is on, press the ONLINE switch (3) to turn the indicator off.
3. Press the WRT EN / TEST switch (4).
4. Press the DENSITY SELECT switch (5) and release, and then press the DENSITY SELECT switch (5) again.
5. Press the WRT EN / TEST switch (4).
6. Press the ONLINE switch (3).
7. Press the DENSITY SELECT switch (5). The service aid now starts with the current device ID being displayed on the alphanumeric display and a cursor is placed on the left character space. The front panel switches function as follows:
   - 1 – Increments the selected character up the ASCII character set.
   - 2 – Decrements the selected character down the ASCII character set.
   - 3 – Moves the cursor one position to the right to select a character.
   - 4 – Exits service aid without saving changed characters.
   - 4 and 5 – Pressing both switches simultaneously saves new characters and exits the service aid; the message SAVING appears.
   - 5 – No function.
8. If the Device ID is TSZ07, exit the service aid. If the Device ID is M995, change it to DEC, and save the change before you exit the service aid.

1.1.2 Graphics Drivers

If you set the ROM to use the serial line as the console, and there is a graphics board in the TURBOchannel option, the X server will not start up because the driver has not been initiated. This is because the graphics driver assumes it has been set up for the graphics console by the time it attempts to attach.

To avoid this, do not use the serial line console when you have a graphics device. Use the graphics console.

1.1.3 RQDX Q-bus Controller Jumper Settings

If there are multiple RQDX controllers and the RQDX2 is the last controller on the bus, ignore the jumper configuration stated in the hardware manual. The jumper setting should be 1, not 4 as stated in the hardware manual.

1.1.4 TK50 and TK70 Tape Usage

When a blank TK50 or TK70 tape is inserted into the drive, calibration marks are written on the tape. These calibration marks determine the tape format. The tape format can only be changed by a bulk tape eraser.

When using TK50 and TK70 tapes, the following rules apply:

- A TK50 formatted tape can be written and read on a TK50 drive.
- A TK50 formatted tape can be read but not written on a TK70 drive. The TK70 drive considers TK50 tapes to be hardware write-protected.
- A TK70 formatted tape can be written or read on a TK70 drive.
- A TK70 tape is completely unusable on a TK50 drive.

Violation of these rules may result in command failure and drive error log messages.

1.1.5 Data Corruption from Programs Accessing Tape Units

The TK70, TS11, and TU81 tape units require the data buffers to be aligned on a machine word boundary. Data corruption can occur if the data buffer boundaries are not aligned properly. When the buffer is declared as a local variable, the buffer will be on the user's program stack; alignment is therefore not ensured and may cause data corruption.

To ensure that the data buffer is correctly aligned, all programs that access tapes must declare the data buffer as a type static or as a global variable. The compilers then assure proper alignment of the data buffer and data.

1.1.6 Required Switch Settings for TSV05 Tape Drive

The switch settings for the TSV05 controller (M7196 and M7206) as described in the manuals *TSV05 Tape Transport – Pocket Service Guide* (EK-TSV05-PS-005) and *TSV05 Tape Transport Subsystem – Technical Manual* (EK-TSV05-TM-004) are incorrect.

The factory switch setting information for the M7206-PA module in the manuals lists switch E61-9 OFF. This factory switch setting does not work on ULTRIX systems running Version 3.0 and higher. The tape device always appears off line to the operating system if the switch is set to OFF.

1.1.7 Installing from a TE16 Tape Drive

During installation, you are asked to identify the software distribution device. If you are using a TE16 tape drive, choose the TU77 tape drive option. When your system is booted, the TE16 will be identified correctly.

1.1.8 Powering Down a TU81 Tape Drive

On the ULTRIX operating system, you cannot power down a controller while it is on line. If you attempt to power down or pull a controller while it is on line, problems may occur. Problems have been encountered because TU81 and TU81+ tape drives contain the TMSCP server logic.

If you must power down a TU81 drive, use the following steps:

1. Disconnect the TU81+ white cable, which connects the drive unit to the back plane of the host.
2. Power down the tape drive.
3. Attend to the tape unit as needed.
4. Power up the tape device.
5. Reconnect the TU81+ white cable.

1.1.9 TU81 Tape Drive Displaying Fault Light

A problem in the TU81 tape unit can cause data transfers to fail. All users who have TU81 tape units should contact a customer service representative to ensure that FCO number TU81 R-005 is applied and that the revision level is up to at least D1. Units that do not have this FCO applied will experience hard errors logged and the unit's controller fault light will light.

1.1.10 MSCP Disks Remain Off Line If Switched Off Line While in Use

If a disk unit served by the MSCP driver (any RA disk) is switched off line while operations are in progress, the disk cannot be brought back on line. If this happens, in-progress and subsequent data transfer operations to the unit will fail. The system call that failed will return an EIO error.

This condition can be cleared only by setting the unit back on line and rebooting the system.

1.1.11 Eight-Bit Terminal Driver Support

You must set up your hardware and software properly if you intend to use a terminal in full eight-bit mode. Refer to the `gettytab`(5) reference page for instructions on how to enable logins on terminal lines that require eight-bit characters. The `p8` and `pd` flags have been added to `gettytab` to facilitate the use of eight-bit characters.

Digital's VT100 series terminals are capable of displaying only the lower half of the DEC Multinational Character Set. Standard seven-bit ASCII characters are included in the lower half of the Multinational Character Set and ISO-8859/2.

Digital's VT200 and VT300 series terminals are capable of displaying the full DEC Multinational Character Set and ISO-8859/2. However, they do not display eight-bit characters when they are in VT100 mode. To determine the current terminal mode, call up the terminal's Set-Up Directory menu and select the `General` setup option.

For example, to change your VT220 terminal set up into eight-bit mode, follow these steps:

1. Call up the terminal's setup menu by pressing the Set-Up (F3) key. Select the `General` menu option.

2. Move to the field that allows you to select the terminal mode. Select the option `VT200 Mode, 7 Bit Controls.`
3. Select the `To Directory` option to return you to top level.
4. Call up the `Comm` menu. Select the `8 Bits, No Parity` option.
5. Exit from setup mode by pressing the Set-Up key again.

Note that when you change a VT200 or VT300 series terminal from VT100 mode, the F11 key no longer represents the escape key. Refer to your terminal's installation guide for a complete description of terminal setup.

The DECwindows terminal emulator, `dxterm` can also be set up for use with eight-bit characters. In this case, the terminal mode must be set to `VT300 Mode, 7-bit Control`. Select the `Customize` menu, then the `General` menu.

1.2 Booting

This section discusses boot issues.

1.2.1 Personal DECstation Model Does Not Recognize Mouse or Keyboard

Either the keyboard or the mouse is not recognized when the Personal DECstation is booted. The keyboard is not recognized until a character is typed; the mouse is not recognized until it is unplugged then plugged in again.

The problem only occurs during system boot and is not a problem during normal system operation.

1.2.2 Installation Incorrectly Identifies DECsystem/DECstation 5000 Model 133

The kernels that are part of the installation procedure and `genvmunix` incorrectly identify the DECsystem/DECstation 5000 Model 133 processors as DECsystem/DECstation 5000 Model 120 processors. Once the target kernel is running, the processors are correctly identified.

1.2.3 Booting MSCP-type Disk Drives

For MSCP-type disk drives, the wait for critical system devices such as `root`, `swap`, and `dump` to become available may take up to 2 minutes. After waiting for 1 minute, a status message appears on the console that indicates that you must wait for a specified device.

1.3 General Installation

The following sections discuss the installation of the ULTRIX operating system and ULTRIX Worksystem Software. For additional installation notes specific to individual processors, see the appropriate section in Chapter 2, Processor–Specific Notes.

1.3.1 New Customer Upgrade Procedure

There is an upgrade procedure for Version 4.2 (or higher) customers. By performing the upgrade, these customers can avoid a full installation, provided they have no new processors or new graphic hardware to support. See Chapter 5 of the *Guide to Installing ULTRIX* for more information.

1.3.2 Upgrade Procedure Mentions Obsolete Manual

During the upgrade procedure, the server installation may refer to the obsolete manual, *Mandatory Upgrade Installation Instructions.* The script should refer to the *Guide to Installing ULTRIX,* where the upgrade procedure is described in Chapter 5.

1.3.3 Upgrade Procedure and the 3D Compatibility Kit

The X11R3 3D Compatibility Kit, which was supported in ULTRIX and UWS Version 4.2A, is no longer supported.

As a result, the 3D Compatibility Kit subset, U3DCOMPSER425, if installed, must be manually deleted before you perform the upgrade procedure. You also must remove some device definitions related to the 3D Compatibility Kit from the system configuration file.

1.3.3.1 Before Performing an Upgrade

Before you perform an upgrade, perform the following tasks:

1. Remove the U3DCOMPSER425 subset from the system subset inventory by entering the following command:

   ```
   # /etc/setld -d U3DCOMPSER425
   ```

2. Edit your system configuration file, `/sys/conf/mips/`*HOSTNAME*, where *HOSTNAME* is the name of your system, to remove the old graphics drivers by deleting the following lines in the file:

   ```
   device      ga0      at ibut?      vector gaintr
   device      gq0      at ibus?      vector gqintr
   ```

3. Edit the system configuration file to ensure that the following device definition is in the file:

   ```
   device      px0      at ibus?      vector pxintr
   ```

4. Shut down and reboot the system, but without using the `old_2da` switch in the `boot` command. If necessary, modify the console environment `boot` variable to remove the `old_2da` switch before you reboot the system. An example of the boot command is:

   ```
   >> boot 3/rz0/vmunix -a
   ```

5. When the system comes back up, log in and perform the upgrade.

1.3.3.2 During the Upgrade

If you performed the steps described in Section 1.3.3.1, you should observe no messages about the 3D Compatibility Kit during the upgrade. If you do observe such messages, something has gone wrong. You must abort the upgrade and perform the tasks in Section 1.3.3.1 again.

1.3.4 Existing SCSI/CAM and OSF/Motif Components Must Be Reinstalled

There are two software components in the kit, SCSI/CAM and OSF/Motif, which are on separate distribution media and must be installed after you install ULTRIX and UWS Version 4.3.

When you install Version 4.3, any earlier versions of SCSI/CAM and OSF/Motif on your system are destroyed. You must install these new components after installing Version 4.3.

1.3.5 VAX Console Media

Some of the VAX console media have not changed since Version 4.2: those media retain their Version 4.2 labels. Other VAX console media have changed and have new labels for Version 4.3. The instructions in the *Guide to Installing ULTRIX* take this fact into account.

1.3.6 Media Labels

Table 1-1 lists the media labels for Version 4.3.

Table 1-1: Media Labels

Media Type	Media Label
TK50 (RISC)	ULTRIX/UWS V4.3 (RISC) BOOT/UPGRADE
	ULTRIX/UWS V4.3 (RISC) SUPPORTED VOL 1
	ULTRIX/UWS V4.3 (RISC) SUPPORTED VOL 2
	ULTRIX/UWS V4.3 (RISC) UNSUPPORTED
TK50 (VAX)	ULTRIX/UWS V4.3 (VAX) BOOT/UPGRADE
	ULTRIX/UWS V4.3 (VAX) SUPPORTED
	ULTRIX/UWS V4.3 (VAX) UNSUPPORTED

Table 1-1: (continued)

Media Type	Media Label
MT9 (RISC)	ULTRIX/UWS V4.3 (RISC) BOOT/UPGRADE
	ULTRIX/UWS V4.3 (RISC) SUPPORTED VOL 1
	ULTRIX/UWS V4.3 (RISC) SUPPORTED VOL 2
	ULTRIX/UWS V4.3 (RISC) SUPPORTED VOL 3
	ULTRIX/UWS V4.3 (RISC) SUPPORTED VOL 4
	ULTRIX/UWS V4.3 (RISC) UNSUPPORTED VOL 1
	ULTRIX/UWS V4.3 (RISC) UNSUPPORTED VOL 2
MT9 (VAX)	ULTRIX/UWS V4.3 (VAX) BOOT/UPGRADE
	ULTRIX/UWS V4.3 (VAX) SUPPORTED VOL 1
	ULTRIX/UWS V4.3 (VAX) SUPPORTED VOL 2
	ULTRIX/UWS V4.3 (VAX) UNSUPPORTED VOL 1
	ULTRIX/UWS V4.3 (VAX) UNSUPPORTED VOL 2
CDROM (RISC)	ULTRIX/UWS V4.3 (RISC) SUPPORTED
CDROM (VAX)	ULTRIX/UWS V4.3 (VAX) SUPPORTED

1.3.7 Space Needed to Install ULTRIX and UWS in RIS and DMS Areas

Table 1-2 lists the space (in kilobytes) needed to install the VAX and RISC subsets for the ULTRIX operating system, the ULTRIX Worksystem Software, the Boot/Upgrade, and the unsupported subsets in a RIS area.

Table 1-2: Space Needed in RIS Areas

Component	Space (RISC)	Space (VAX)
ULTRIX	59752	32988
UWS	36537	21210
Boot/Upgrade	15109	7813
Unsupported	34419	36577

Instructions for determining the size of a DMS area are contained in *Guide to Sharing Software on a Local Area Network.* The approximate sizes of the software subsets for Version 4.3 are:

- RISC: all ULTRIX/UWS, 225 Mbytes
- RISC: all ULTRIX/UWS unsupported, 52 Mbytes
- VAX: all ULTRIX/UWS, 120 Mbytes
- VAX: all ULTRIX/UWS unsupported, 48 Mbytes

1.3.8 ULTRIX and UWS Subset Sizes

The following sections list the subset sizes for the supported, the Boot/Upgrade, and unsupported subsets that make up ULTRIX and UWS Version 4.3.

For a description of each subset, see the *Guide to Installing ULTRIX.*

1.3.8.1 ULTRIX RISC Supported Subset Sizes

Table 1-3 lists the supported ULTRIX RISC subsets in kilobytes for the `root`, `/usr`, and `/var` directories.

Table 1-3: ULTRIX RISC Supported Subset Sizes

Subset	root Size (Kbytes)	/usr Size (Kbytes)	/var Size (Kbytes)	Total (Kbytes)
UDTACCT430	0.043	249.856	—	249.899
UDTAFM430	—	901.258	—	901.258
UDTBASE430	3133.740	20270.800	35.126	23439.700
UDTBIN430	12.880	41212.800	—	41225.700
UDTCDABASE430	—	4007.940	—	4007.940
UDTCDAPGMR430	—	1125.190	—	1125.190
UDTCOMM430	13.731	1253.380	—	1267.110
UDTDCMT430	0.017	274.375	—	274.392
UDTDCMTEXT430	—	1073.620	—	1073.620
UDTDL430	87.004	2902.880	—	2989.880
UDTEXAMPLES430	—	1015.930	—	1015.930
UDTEXER430	—	974.954	—	974.954
UDTINET430	442.298	5674.740	58.354	6175.390
UDTINTLPGMR430	—	110.727	—	110.727
UDTINTLRT430	—	587.279	—	587.279
UDTKERB430	—	1277.270	773.269	2050.540
UDTMAN430	—	3194.520	—	3194.520
UDTMANPGMR430	—	1455.600	—	1455.600
UDTMH430	0.512	7877.700	1.024	7879.240
UDTMOP430	30.489	430.729	78.336	539.554

Table 1-3: (continued)

Subset	root Size (Kbytes)	/usr Size (Kbytes)	/var Size (Kbytes)	Total (Kbytes)
UDTNFS430	201.744	1435.240	553.463	2190.450
UDTPGMR430	0.040	8368.280	—	8368.320
UDTPRESTO430	—	163.840	0.539	164.379
UDTPRINT430	32.960	2607.150	0.512	2640.620
UDTRPCDEV430	—	662.293	94.208	756.501
UDTRPCRT430	0.014	275.750	1338.790	1614.550
UDTSCCS430	—	1212.870	—	1212.870
UDTSEC430	385.975	959.922	—	1345.900
UDTSMSCAMP430	—	99.872	—	99.872
UDTUMAIL430	28.054	799.947	—	828.001
UDTUUCP430	0.020	620.727	1014.800	1635.550
TOTALS	4369.520	113077.000	3948.420	121395.000

1.3.8.2 UWS RISC Supported and Unsupported Subset Sizes

Table 1-4 lists the supported and unsupported RISC UWS subsets in kilobytes for the `root`, `/usr`, and `/var` directories.

Table 1-4: UWS RISC Supported and Unsupported Subset Sizes

Subset	root Size (Kbytes)	/usr Size (Kbytes)	/var Size (Kbytes)	Total (Kbytes)
UDWDECW430	—	10804.700	—	10804.700
UDWFONT15430	—	3773.710	—	3773.710
UDWFONT430	—	2801.140	—	2801.140
UDWFONTSTR430	—	164.352	—	164.352
UDWMAIL430	—	2533.610	—	2533.610
UDWMAN430	—	317.632	—	317.632
UDWMANPGMR430	—	1562.800	—	1562.800
UDWSER430	—	4641.280	100.396	4741.680
UDWSYSMGNT430	—	2909.520	—	2909.520
UDWTPV024425	—	28.672	16437.300	16466.000
UDWTPVMAN425	—	1.865	—	1.865
UDWWSTSER430	—	2759.960	—	2759.960
UDWX11430	—	6304.620	—	6304.620
UDWXDEV430	—	13501.900	—	13501.900
UDWXTXSER430	—	2170.880	—	2170.880

Table 1-4: (continued)

Subset	root Size (Kbytes)	/usr Size (Kbytes)	/var Size (Kbytes)	Total (Kbytes)
UDXUNEXAMP430	—	1053.970	—	1053.970
UDXUNFONTS430	—	1629.610	—	1629.610
UDXUNMAN430	—	452.563	—	452.563
UDXUNMIT430	—	26596.400	—	26596.400
TOTALS	—	84009.200	16537.700	100547.000

1.3.8.3 ULTRIX VAX Supported Subset Sizes

Table 1-5 lists the supported subset sizes for ULTRIX on VAX processors. The subset sizes are described in kilobytes for the `root`, `/usr`, and `/var` directories.

Table 1-5: ULTRIX VAX Supported Subset Sizes

Subset (VAX)	root Size (Kbytes)	/usr Size (Kbytes)	/var Size (Kbytes)	Total (Kbytes)
ULTACCT430	0.043	142.336	—	142.379
ULTAFM430	—	901.258	—	901.258
ULTBASE430	2024.290	10665.900	35.638	12725.800
ULTBIN430	12.879	7184.650	—	7197.530
ULTBSC430	0.036	209.920	—	209.956
ULTCDABASE430	—	2792.450	—	2792.450
ULTCDAPGMR430	—	586.124	—	586.124
ULTCOMM430	13.731	751.616	—	765.347
ULTDCMT430	0.017	171.805	—	171.822
ULTDCMTEXT430	—	749.882	—	749.882
ULTDL430	47.068	1637.060	—	1684.130
ULTEXAMPLES430	—	716.940	—	716.940
ULTEXER430	—	588.198	—	588.198
ULTINET430	277.434	3418.490	106.482	3802.410
ULTINTLPGMR430	—	110.727	—	110.727
ULTINTLRT430	—	326.735	—	326.735
ULTKERB430	—	931.660	538.773	1470.430
ULTMAN430	—	3465.290	—	3465.290
ULTMANPGMR430	—	1277.940	—	1277.940
ULTMH430	0.512	4856.900	1.024	4858.440
ULTMOP430	30.489	226.953	49.716	307.158
ULTNFS430	112.656	800.358	257.527	1170.540

Table 1-5: (continued)

Subset (VAX)	root Size (Kbytes)	/usr Size (Kbytes)	/var Size (Kbytes)	Total (Kbytes)
ULTPASCAL430	—	720.784	—	720.784
ULTPGMR430	0.040	3648.780	—	3648.820
ULTPRESTO430	—	96.256	0.539	96.795
ULTPRINT430	15.552	2006.180	0.512	2022.240
ULTRPCDEV430	—	456.517	53.248	509.765
ULTRPCRT430	0.014	275.750	870.819	1146.580
ULTSCCS430	—	719.170	—	719.170
ULTSEC430	256.951	574.898	—	831.849
ULTSMSCAMP430	—	99.872	—	99.872
ULTUMAIL430	28.054	511.179	—	539.233
ULTUUCP430	0.020	365.751	629.780	995.551
ULTVAXC430	—	878.598	—	878.598
TOTALS	2819.790	52866.900	2544.060	58230.800

1.3.8.4 UWS VAX Supported and Unsupported Subset Sizes

Table 1-6 lists the supported and unsupported subset sizes for UWS on VAX processors. The subset sizes are described in kilobytes for the `root`, `/usr`, and `/var` directories.

Table 1-6: UWS VAX Supported and Unsupported Subset Sizes

Subset (VAX)	root Size (Kbytes)	/usr Size (Kbytes)	/var Size (Kbytes)	Total (Kbytes)
ULXUNEXAMP430	—	1053.970	—	1053.970
ULXUNFONTS430	—	1629.610	—	1629.610
ULXUNMAN430	—	452.563	—	452.563
ULXUNMIT430	—	13282.000	—	13282.000
UWS3DFONT430	—	3922.590	—	3922.590
UWSDECW430	—	8014.850	—	8014.850
UWSFONT15430	—	3773.710	—	3773.710
UWSFONT430	—	2801.140	—	2801.140
UWSMAIL430	—	1673.450	—	1673.450
UWSMAN430	—	321.428	—	321.428
UWSMANPGMR430	—	1562.800	—	1562.800
UWSSER430	—	6096.730	81.964	6178.690
UWSSYSMGNT430	—	2176.340	—	2176.340
UWSX11430	—	4568.420	—	4568.420

Table 1-6: (continued)

Subset (VAX)	root Size (Kbytes)	/usr Size (Kbytes)	/var Size (Kbytes)	Total (Kbytes)
UWSXDEV430	—	4502.950	—	4502.950
TOTALS	—	55832.600	81.964	55914.500

1.3.8.5 Boot Upgrade Subset Sizes

The following tables, Table 1-7 and Table 1-8, list the RISC and VAX subsets that make up the Boot/Upgrade subset sizes in kilobytes for the `root`, `/usr`, and the `/var` directories.

Table 1-7: RISC Boot/Upgrade Subset Sizes

Subset	root Size (Kbytes)	/usr Size (Kbytes)	/var Size (Kbytes)	Total (Kbytes)
UDTGENVMUNIX431	—	2811.540	—	2811.540
TOTALS	—	2811.540	—	2811.540

Table 1-8: VAX Boot/Upgrade Subset Sizes

Subset	root Size (Kbytes)	/usr Size (Kbytes)	/var Size (Kbytes)	Total (Kbytes)
ULTGENVMUNIX431	—	1564.670	—	1564.670
TOTALS	—	1564.670	—	1564.670

1.3.8.6 ULTRIX DECnet Subset Sizes

The following tables list the RISC and VAX subsets sizes for ULTRIX DECnet.

Table 1-9 lists the ULTRIX DECnet subset sizes for RISC processors in kilobytes for the `root`, `/usr`, and `/var` directories.

Table 1-9: DECnet RISC Subset Sizes

Subset RISC	root size (kbytes)	/usr size (kbytes)	/var size (kbytes)	Total (kbytes)
DNPBASE430	1.794	2742.11	0	2743.9
DNPINETGW430	0	458.752	0	458.752

Table 1-9: (continued)

Subset RISC	root size (kbytes)	/usr size (kbytes)	/var size (kbytes)	Total (kbytes)
DNPMAN430	0	99.661	0	99.661
DNPUNS430	0	501.555	0	501.555
TOTALS	1.794	3902.08	0	3803.87

Table 1-10 lists the ULTRIX DECnet subset sizes for VAX processors in kilobytes for the `root`, `/usr`, and `/var` directories.

Table 1-10: DECnet VAX Subset Sizes

Subset VAX	root size (kbytes)	/usr size (kbytes)	/var size (kbytes)	Total (kbytes)
DNUBASE430	1.794	1790.22	0	1792.01
DNUINETGW430	0	310.272	0	310.272
DNUMAN430	0	99.661	0	99.661
DNUUNS430	0	317.766	0	317.766
TOTALS	1.794	2517.92	0	2519.71

1.3.8.7 ULTRIX and UWS Encryption Subset Sizes

The following tables list the ULTRIX and UWS encryption subset sizes for both RISC and VAX processors.

Table 1-11 lists the ULTRIX encryption subset sizes on RISC processors in kilobytes for the `root`, `/usr`, and `/var` directories.

Table 1-11: ULTRIX RISC Encryption Subset Sizes

Subset	root Size (Kbytes)	/usr Size (Kbytes)	/var Size (Kbytes)	Total (Kbytes)
UDCCRYPT430	—	610.304	0.538	610.842
UDCPGMR430	—	1581.310	—	1581.310
TOTALS	—	2191.610	0.538	2192.150

Table 1-12 lists the encryption subset sizes for VAX processors in kilobytes for the `root`, `/usr`, and `/var` directories.

Table 1-12: ULTRIX VAX Encryption Subset Sizes

Subset	root Size (Kbytes)	/usr Size (Kbytes)	/var Size (Kbytes)	Total (Kbytes)
ULCCRYPT430	—	376.832	0.538	377.370
ULCPGMR430	—	1259.920	—	1259.920
TOTALS	—	1636.750	0.538	1637.290

1.3.8.8 ULTRIX Unsupported Subset Sizes

The following tables list the ULTRIX unsupported subset sizes on both VAX and RISC processors.

Table 1-13 lists the ULTRIX unsupported subset sizes on RISC processors in kilobytes for `root`, `/usr`, and `/var` directories.

Table 1-13: ULTRIX RISC Unsupported Subset Sizes

Subset	root Size (Kbytes)	/usr Size (Kbytes)	/var Size (Kbytes)	Total (Kbytes)
UDXBASE430	0.192	3371.660	0.512	3372.360
UDXBIB430	—	291.746	—	291.746
UDXCOURIER430	—	156.246	—	156.246
UDXDCMT430	—	394.138	—	394.138
UDXDOC430	—	3447.050	—	3447.050
UDXEDIT430	—	6295.620	—	6295.620
UDXGAMES430	—	2488.560	—	2488.560
UDXINET430	90.164	848.291	—	938.455
UDXLEARN430	—	737.854	—	737.854
UDXMAN430	—	175.436	—	175.436
UDXNEWS430	—	1321.060	—	1321.060
UDXNOTES430	—	1856.250	—	1856.250
UDXRCS430	—	212.617	—	212.617
UDXSHELLS430	—	90.064	—	90.064
UDXTERM430	—	324.464	—	324.464
UDXTOOLS430	—	115.650	—	115.650
TOTALS	90.356	22126.700	0.512	22217.600

Table 1-14 lists the ULTRIX unsupported subset sizes for VAX processors in kilobytes for the `root`, `/usr`, and `/var` directories.

Table 1-14: ULTRIX VAX Unsupported Subset Sizes

Subset (VAX)	root Size (Kbytes)	/usr Size (Kbytes)	/var Size (Kbytes)	Total (Kbytes)
ULXAPL430	—	269.805	—	269.805
ULXBASE430	1.511	1882.070	0.512	1884.090
ULXBIB430	—	194.058	—	194.058
ULXCOURIER430	—	102.594	—	102.594
ULXCPM430	—	28.934	—	28.934
ULXDCMT430	—	370.936	—	370.936
ULXDOC430	—	3447.050	—	3447.050
ULXEDIT430	—	6295.620	—	6295.620
ULXF77430	—	732.647	—	732.647
ULXGAMES430	—	2223.270	—	2223.270
ULXHYPER430	—	80.771	—	80.771
ULXICON430	—	346.396	—	346.396
ULXINET430	52.276	449.181	—	501.457
ULXINGRES430	—	2609.510	—	2609.510
ULXLEARN430	—	652.285	—	652.285
ULXLISP430	—	3216.470	—	3216.470
ULXMAN430	—	310.709	—	310.709
ULXMOD2430	—	1035.550	—	1035.550
ULXNEWS430	—	1321.060	—	1321.060
ULXNOTES430	—	1174.270	—	1174.270
ULXRCS430	—	226.733	—	226.733
ULXSHELLS430	—	54.224	—	54.224
ULXSPMS430	—	1211.480	—	1211.480
ULXTERM430	—	324.464	—	324.464
ULXTOOLS430	—	54.210	—	54.210
ULXVARIAN430	—	2736.110	—	2736.110
TOTALS	53.787	31350.400	0.512	31404.700

1.3.9 Unsupported Subsets

To install the unsupported subsets from tape, load the unsupported tape. To install subsets from your CDROM distribution, mount the `c` partition of the CDROM on `/mnt` and load subsets from `/mnt/RISC/UNSUPPORTED` or `/mnt/VAX/UNSUPPORTED`, as is appropriate.

1.3.10 The setld Command with the -v Function

The `setld` command with the `-v` function does not invoke `fverify`. This function checks to see if the subset is installed. If the subset is installed and it contains an installation verification procedure (IVP), the IVP is executed.

1.3.11 Layered Products and the setld Command

Some layered products will not install because of an incompatibility with the `setld` command. When you try to install them, these products will issue an error message and exit before the `setld` menu is presented. To install these products, set the `STL_NOACTM` environment variable to 1. On VAX and RISC systems, `csh` users should enter the following command as `root`:

```
# setenv STL_NOACTM 1
```

Users of all other shells should enter this command:

```
# STL_NOACTM=1; export STL_NOACTM
```

Once you have set this variable, the product will be installed correctly.

Before installing any more products, unset the variable. On VAX and RISC systems, `csh` users should enter the following command as `root`:

```
# unsetenv STL_NOACTM
```

Users of all other shells should log out of the system and log back in before installing more software.

1.3.12 Optional Removal of the Kernel Object Subset

The ULTRIX operating system contains a kernel object subset, which allows full debug capabilities using the `dbx` debugger.

In order to save space, you can remove your kernel object subset from your installed system. On RISC processors, enter the following command:

```
# /etc/setld -d UDTBIN430
```

On VAX processors, enter the following command:

```
# /etc/setld -d ULTBIN430
```

It is important that you understand some of the trade-offs you make by removing this subset.

If you need to apply a kernel patch to your environment, you will need to reinstall the kernel object subset. This might require that you remove other system or user environments to make room for its installation. (This assumes that the space made available after removing the subset was used for another purpose.)

In general, any time you need to build a kernel for your system, the subset will need to be reinstalled. This could be for a simple need to modify some hardware or software configuration parameter in the system's configuration file.

We recommend that this procedure not be followed routinely. Rather, it is intended to solve space problems on smaller systems with smaller system disks. If you choose to remove the kernel object subset, wait until the system is properly configured and proven for some reasonable period of time.

1.3.13 Installing the Internationalization Subsets

Internationalization software has been divided into two subsets:

- UDTINTLPGMR430 and UDTINTLRT430 (RISC)
- ULTINTLPGMR430 and ULTINTLRT430 (VAX)

If you plan to use the Internationalization software, you must install both subsets.

1.3.14 Initial System Load Using RIS

During a base system installation using RIS, if a problem occurs when contacting the RIS server or with network traffic, an error in the `setld` utility causes the installation procedure to state that all mandatory subsets have been installed successfully and the following message is displayed:

```
The installation software successfully installed the mandatory software
subsets. However, it failed to install other parts of the supported
software subsets. You can continue with the installation and install
the other supported software subsets at a later time using the setld
utility or, you can request that the installation software again try
to install these supported software subsets.

Select one of the following options:

        1) Continue with the installation
        2) Try to re-install supported software subsets

Enter your choice [1]:
```

In order to complete the installation, check the network/RIS server status with the system administrator. If the problems are corrected, type 2 in response to the query and the installation process will continue.

1.3.15 Display Server Installation Script

The server installation script loads all servers but then removes all servers not relevant to the hardware you are installing on.

1.3.16 Miskitted Fonts

With this release, some 100 dpi fonts are installed from a 75 dpi font subset. By default, 75 dpi fonts are always installed during an installation. Therefore, miskitted 100 dpi fonts are also automatically installed whether or not they are needed. There are two workarounds for this problem:

- If you want only 75 dpi fonts installed, the installation process installs them automatically. To remove the miskitted 100 dpi fonts, delete the `/usr/lib/X11/fonts/decwin/100dpi` directory.
- If you want only 100 dpi fonts installed, the installation process installs 75 dpi fonts automatically, and you must specify that you want 100 dpi fonts installed. Then, to remove the 75 dpi fonts, delete the `/usr/lib/X11/fonts/decwin/75dpi` directory.

If you remove either of these directories, subsequent invocations of `fverify` will indicate an error.

1.3.17 ULTRIX Worksystem Fonts Optional in DMS Installations

ULTRIX Worksystem Software fonts are not automatically installed on a diskless RISC or VAX workstation when the Diskless Management Services (DMS) utility is used for installation. Because all fonts are contained in optional subsets, when you install only the mandatory subsets, the X server will not come up because no fonts are available to it. Therefore, when doing a DMS installation, you must install a font for the server to use.

1.3.18 Rebuilding /genvmunix After an Upgrade Installation

The generic kernel image created by the installation process as `/genvmunix` can be useful at a later time. For example, it can be useful when you add new hardware such as a backup kernel image, and also for Prestoserve crash recovery when the system disk is lost.

To save the generic kernel image for later use, rebuild the generic kernel image after you perform the upgrade installation, but before you install any layered product. See the *Guide to Installing ULTRIX* for information on how to perform this upgrade.

To rebuild the generic kernel on a RISC system:

```
# /etc/doconfig -c GENERIC
# cp /sys/MIPS/GENERIC/vmunix /genvmunix
# rm -rf /sys/MIPS/GENERIC
```

To rebuild the generic kernel on a VAX system:

```
# /etc/doconfig -c GENERIC
# cp /sys/VAX/GENERIC/vmunix /genvmunix
# rm -rf /sys/VAX/GENERIC
```

1.3.19 Upgrading a RIS Client

When doing a RIS upgrade, be sure the client is modified to include both the Supported Products and the Boot Upgrade Products from the menu. Refer to Chapter 5 of the *Guide to Installing ULTRIX* for further information.

1.3.20 Building DMS Areas Including The Personal Visualizer Software

While building a DMS area on a server that includes The Personal Visualizer software subsets, you may see the following question:

```
Would you like to run the IVP after the installation (y/n) [y]?
```

Answer `n` to the question, rather than taking the default `y` answer. There is no IVP for The Personal Visualizer software.

1.3.21 Potential Problem: DMS Clients with 8 Mbyte Memory

If you have a DMS client system with a minimum memory of 8 Mbytes, you could receive `not enough core` messages in the Session Manager window when logging into the system. If you receive the message, the system is inoperable. To work around the problem, ask your system administrator to increase the swap space for this DMS client to 24 Mbytes.

1.4 Configuration

The following sections discuss system configuration.

1.4.1 BINARY Configuration File and slip

The BINARY configuration file in `/sys/conf/mips` contains the pseudodevice definition for `slip`. However, the files necessary to build a kernel with `slip` are in the UDXINET430 subset, Unsupported TCP/IP Networking Util Extension.

If you want the `slip` program, you must install the UDXINET430 subset from the Unsupported Kit. If you do not want the program, you must delete its pseudodevice definition from the configuration file. The line to delete is:

```
pseudo-device sl
```

1.4.2 Vectors

To use vector processors on a system that supports vector processors, 6000-4xx, 6000-5xx, or 9000, you must add the following lines to the configuration file:

```
options           VECTORS
pseudo-device     vectors
```

Use the following syntax format for the options line:

```
options VECTORS =<number>
```

In the previous example, <number> is the maximum number of vector processes allowed. If a number is not specified, the maximum defaults to the MAXUPROC divided by 10.

1.4.3 Use of the hsc? Token in the System Configuration File

The `config` utility incorrectly handles the hsc? token in the system configuration file. Use the mscp token in place of the hsc? token to indicate a wildcard connection or, completely specify the hsc controller. For example:

```
ra0    at       hsc?  drive 0
```

The preceding line can be replaced with the following to match any hsc:

```
ra0    at       mscp  drive 0
```

Or, if connected to hsc0, the line can be replaced with the following:

```
ra0    at       hsc0  drive 0
```

1.4.4 Q-bus SCSI Adapter (KZQ)

To install the Q-bus SCSI (KZQ) adapter on an RISC ULTRIX machine, you must edit the system configuration file after the initial software installation. Because the KZQ is not a valid boot device, it is not supported by the generic kernel; hence, `doconfig` does not create a valid configuration file for a system supporting the KZQ device.

Note

Do not install the KZQSA hardware before you have built a kernel that includes support for the hardware. If you install the hardware before building the kernel for it, data corruptions will occur on the hardware.

The following information describes the entries you must make in the controller, disk, and tape sections of the configuration file. For more information on configuring SCSI devices, see the *Guide to Configuration File Maintenance.*

When you use multiple SCSI controllers, the controllers must be numbered sequentially. For example, on a DECsystem 5500 with the onboard SCSI controller, `asc0`, the KZQSA will be the second SCSI controller, `kzq1`, and the third SCSI controller, `kzq2`. There can be no `ksq0` because the onboard asc controller is SCSI controller zero.

To enable KZQ controller support, add the following line to the controller section of the configuration file:

```
controller     kzq1      at uba0       csr  XXXXXXX vector kzq_intr
```

In the previous example, XXXXXXX represents the next available floating CSR. Because the KZQ is a Q-bus device, the CSRs at which the device appears is determined by the other devices present on the Q-bus. For example, in the following entry, the device appears at CSR 0761300. This CSR may not be valid for your configuration.

```
controller     kzq1      at uba0      csr 0761300 vector kzq_intr
```

Along with the other disk entries in the configuration file, add an entry for each disk device you want to configure. For example:

```
disk           rz8       at kzq1      drive 0
```

In the tape section of the configuration file, add an entry for each SCSI tape device you want recognized. For example:

```
tape           tz15       at kzq1     drive 7
```

To configure the KZQ to allow SCSI disks or tapes at any SCSI IDs, add the following to the configuration file:

```
controller      kzq1       at uba0            csr 0761300 vector kzq_intr

disk            rz8       at kzq1                 drive 0
disk            rz9        at kzq1          drive 1
disk            rz10       at kzq1          drive 2
disk            rz11       at kzq1          drive 3
disk            rz12       at kzq1          drive 4
disk            rz13       at kzq1          drive 5
disk            rz14       at kzq1          drive 6
disk            rz15       at kzq1          drive 7

tape            tz8       at kzq1             drive 0
tape            tz9        at kzq1          drive 1
tape            tz10       at kzq1          drive 2
tape            tz11       at kzq1          drive 3
tape            tz12       at kzq1          drive 4
tape            tz13       at kzq1          drive 5
tape            tz14       at kzq1          drive 6
tape            tz15       at kzq1          drive 7
```

1.4.5 Secure Console Mode

Even a system that is running in secure console mode can be interrupted during a reboot and brought up in single-user mode. Thus, if your system is running in secure console mode, the system administrator should edit the `/etc/rc` file to prevent reboots from being interrupted.

To edit the `/etc/rc` file to prevent reboots from being interrupted, follow these steps:

1. Add the following line to the top of the `/etc/rc` file. This line must be the first line in the file:

```
trap '' 1 2 3
```

2. Find the case statement in the `/etc/rc` file that reboots the system after `fsck` is performed on the file system. The case statement looks like this:

```
echo Automatic reboot in progress... >/dev/console
/etc/fsck -p >/dev/console
case $? in

    0)
            ;;
    4)
            /etc/reboot -n
            ;;
    8)
            echo "Automatic reboot failed... help!" >/dev/console
            exit 1
            ;;
   12)
            echo "Reboot interrupted" >/dev/console
            exit 1
            ;;
    *)
            echo "Unknown error in reboot" > /dev/console
            exit 1
            ;;
esac
```

3. Change all occurrences of `exit 1` in the preceding case statement to `/etc/halt`. Making this change will cause the system to shut down again if there are any attempts to gain access to the system by interrupting the boot.

   After you have edited the preceding case statement, it should look like this:

```
echo Automatic reboot in progress... >/dev/console
/etc/fsck -p >/dev/console
case $? in

    0)
            ;;
    4)
            /etc/reboot -n
            ;;
    8)
            echo "Automatic reboot failed... help!" >/dev/console
            /etc/halt
            ;;
    12)
            echo "Reboot interrupted" >/dev/console
            /etc/halt
            ;;
```

```
    *)
            echo "Unknown error in reboot" > /dev/console
            /etc/halt
            ;;
esac
```

If a system problem arises that warrants booting to single-user mode, you can change the console secure mode and boot the system to single-user mode at the console command line. See the *Guide to System Shutdown and Startup* for the boot string specific to your processor.

1.4.6 Configuration of Q-bus Terminal Multiplexer Lines

The installation process creates 8 terminal lines for each `cxa16` or `cxb16` terminal multiplexer and 16 lines for the `cxf32` terminal multiplexer. The correct number of lines is 16 for a `cxa16` or `cxb16` and 32 lines for a `cxf32`.

To correct this problem, remake the lines as follows:

1. Log in as `root` or become superuser.
2. Remove the `/dev/tty` lines that correspond to the `cxa16`, `cxb16`, or `cxf32` multiplexers by using the `rm` command. You can identify the lines that need to be removed by using the `ls -l` command and looking for the major number 33, as follows:

   ```
   # ls -l /dev/tty* | grep 33,
   ```

   Note

   The `dhv`, `dhq`, and `cxy08` also share the major number, 33. If you delete these lines, they will also have to be remade.

3. Remake the correct number of terminal lines with the `MAKEDEV` command.

   For example, if your system has only two `cxa16` devices, enter the following command in the `/dev` directory:

   ```
   # MAKEDEV cxa0 cxa1
   ```

   The `cxf32` is configured as two `cxa16` devices. If your system has a `cxf32`, enter the following command in the `/dev` directory:

   ```
   # MAKEDEV cxa0 cxa1
   ```

4. After you create the correct number of lines, update the `/etc/ttys` file to include the previously missing lines. For more information on updating the `/etc/ttys` file, see the *Guide to System and Network Setup.*

1.4.7 System Configuration When Disk Controllers Are in Floating Address Space

When performing a system configuration for any system that has UNIBUS/QBUS adapters with devices in floating address space, the `sizer` utility functions differently. For example, if the device in floating address space is a disk controller, the `sizer` utility does not assume that drives are attached to it. The `sizer` utility cannot correctly size the number of drives attached to this disk controller in floating

address space as you can note from the following example of an incorrect configuration file:

```
controller      uda0      at      uba0
controller      uq0       at      uda0    csr  0172150  vector  uqintr
disk            ra0       at      uq0     drive 0
controller      uda1      at      uba1
controller      uq17      at      uda1    csr  0160334  vector  uqintr
disk            ra1       at      uq17    drive 0
disk            ra2       at      uq17    drive 1
disk            ra3       at      uq17    drive 2
disk            ra5       at      uq17    drive 3
```

In the previous example, there is a disk controller (uq17 at uda1) in floating address space. In previous versions of the `sizer` utility, `sizer` assumes that four disk drives are attached to this disk controller. However, for the KFQSA controller, this is an incorrect assumption by the `sizer` utility. Because the `sizer` utility cannot correctly determine the number of drives connected to this disk controller, it assumes there are none.

Here is an example of the correct configuration file:

```
controller      uda0      at      uba0
controller      uq0       at      uda0    csr  0172150  vector  uqintr
disk            ra0       at      uq0     drive 0
controller      uda1      at      uba1
controller      uq17      at      uda1    csr  0160334  vector  uqintr
```

In the previous example, there is a disk controller (uq17 at uda1) in floating address space. However, the `sizer` utility now assumes that no drives are attached to it.

Hence, you must edit the configuration file during the installation process to add the correct number of drives for disk controllers that exist in floating address space.

Select the ADVANCED installation option during installation, and when the system asks you if you want to edit the configuration file, type `y` and complete the following steps:

1. For all disk controllers in floating address space add the correct number of `ra` and `rz` drives attached to it.
2. Make the `ra` and `rz` numbers sequential.

Note

If you are unable to determine the correct drive numbers, contact customer service.

1.4.8 The Console Entry in the ttys File

The default terminal type for the console entry in the `/etc/ttys` file is set to vt100. This works if you have a terminal console. However, if you have a hardcopy console, you must change the entry in the `ttys` file to match your console terminal type.

1.4.9 Synchronization Errors for Autodial Modem on a DMF32 Interface

If you have an autodial modem connected to a DMF32 interface and you are using the generic dialer routines in `acucap`, `tip`, or `uucp`, the system may not be able to

open the modem and may print either of the following messages:

```
tip: can't synchronize
uucp: can't synchronize
```

The problem occurs because the DMF32 interface cannot return characters to the system until a carrier is detected by the modem.

If you encounter this problem, edit each entry in the `/etc/acucap` file that refers to a modem connected to a DMF32 interface to include the `si` Boolean flag. The `si` Boolean flag disables checking of responses from the modem until the carrier is detected. For more information, see `acucap`(5) in the *ULTRIX Reference Pages.*

1.4.10 Shared Lines Do Not Work over Direct Connections

The use of shared lines requires modem control. Carrier detection must be raised upon receipt of an incoming connection and must be dropped when the remote party hangs up. Direct connects that use modem eliminators do not obey this protocol and cannot be used for shared lines. If you try to use shared lines on a direct connect line that has Carrier Detect strapped high, you will disable the line.

1.4.11 Reactivating Hardwired Terminals

Hardwired terminal ports other than the console port may hang as a result of electrical noise appearing on the line when the terminal is turned off and then turned back on. When the port is hung, the terminal does not respond to keyboard input. To reactivate the terminal line, follow these steps:

1. Log in to the console as `root` or become superuser.
2. Determine the number of the hung terminal line by entering the `last` command with the user's login name as an argument.
3. Determine the process identification (PID) of the `getty` process associated with the hung terminal line by entering the `ps` command with the `-ax` option.
4. Use the `kill` command with the `-9` option to kill the process.

The following example shows how to reactivate a hung terminal line. Assume that the login name of the user is kafka:

```
# last kafka

kafka    tty03      Mon Nov 18 10:00   still logged in
kafka    tty03      Mon Nov 18 08:35 - 09:08  (00:33)
kafka    tty03      Mon Nov 18 00:26 - 02:00  (01:33)

# ps -ax

  PID TT STAT  TIME COMMAND
    0 ?  D     0:01 swapper
    1 ?  I     0:34 init
    2 ?  D     0:00 pagedaemon

  159 03 I     0:00 - 2 tty03 (getty)
  160 04 I     0:00 - 2 tty04 (getty)

# kill -9 159
```

1.4.12 Leave Terminals Power On

Improperly terminated terminal lines can cause the associated `getty` process to use the CPU heavily. Line interference causes the `getty` process to assume that a user is attempting to log in. This problem will be repeated continuously, causing degradation in system performance. Either keep your terminals powered on at all times, or if a terminal line is not used, specify it as "off" in the `/etc/ttys` file to prevent a `getty` process from being started on the line.

Processor-Specific Notes 2

This chapter discusses processor-specific issues related to the ULTRIX and ULTRIX Worksystem Software. The following processors are discussed:

- MicroVAX II, VAXstation II, and VAXstation II/GPX
- VAXstation 2000 and MicroVAX 2000
- VAXstation 3520 and VAXstation 3540
- VAX 11/780 and 11/785
- VAX 6000 Model 400 and Model 500 Series
- DECstation/DECsystem 2100 and 3100
- DECstation/DECsystem 5000 Series
- Personal DECstation Series
- DECsystem 5100
- DECsystem 5400
- DECsystem 5500
- DECsystem 5800 Series
- DECsystem 5900

2.1 MicroVAX II, VAXstation II, and VAXstation II/GPX

The following sections discuss the MicroVAX II, VAXstation II, and VAXstation II/GPX.

2.1.1 Disabling Bootable Disks on MicroVAX II, VAXstation II, and VAXstation II/GPX Systems

The boot programs residing in PROM search for a bootable disk using a specific priority scheme. Removable disks are searched first, followed by nonremovable disks. For example, if your system resides on disk unit 1 but disk unit 0 always boots after a power failure or as a result of entering BOOT at the console prompt (>>>), you can disable disk unit 0's boot block.

To disable a particular disk unit's boot block, log in as `root` and enter the following command:

```
# dd if=/.profile of=/dev/rraNa count=1
```

Replace the *N* in the preceding example with the actual RA number of the disk you are disabling. Once you have disabled the disk, the boot program will skip the disabled disk in its search for a bootable disk.

2.1.2 Console Port Printer Procedure: MicroVAX Systems

The following procedure explains how to attach a console port printer to a MicroVAX system in a BA23 or BA123 enclosure. This procedure applies to MicroVAX systems that do not have a multiplexer at the time that the ULTRIX operating system is installed.

To connect the printer after installing ULTRIX software, follow these steps:

1. Open the back of the enclosure, if applicable.
2. Set the HALT ENABLE/DISABLE switch on the back of the system to the DISABLE position.
3. Set the console select switch to the proper speed for your printer.
4. Log in as `root` or become superuser.
5. Enter these commands:

```
# cd /dev
# MAKEDEV ttycp
# ln ttycp lp
# chown daemon lp
# chmod 660 lp
```

6. Use the printer setup procedure as described in the *Guide to System and Network Setup*.

2.1.3 XLoadQueryFont Function

The `XLoadQueryFont` function loads the specified font and returns a font structure specifying the properties of that font. An `XLoadQueryFont` function call causes the `Xqdsg` server to crash if the requested font is not available in a compiled form.

To create compiled PCF fonts from uncompiled fonts in BDF format, do the following:

1. Run the `dxfc` command in that directory to compile the fonts. Put the output in a directory that is on the font path.
2. Use the `dxmkfontdir` command to create a `fonts.dir` file listing font names and font files for the X server.

2.2 VAXstation 2000 and MicroVAX 2000

The following sections discuss the VAXstation 2000 and the MicroVAX 2000.

2.2.1 Special File Usage: VAXstation 2000 and MicroVAX 2000

On VAXstation 2000 and MicroVAX 2000 processors, do not create or attempt to use the `/dev/tty00` special file, because it interferes with the operation of the console device.

For the VAXstation 2000, do not attempt to use the `/dev/tty01` special file because it interferes with the operation of the mouse. You can use this file on the MicroVAX 2000.

Refer to `ss`(4) in the *ULTRIX Reference Pages* for information on how the `/dev/tty??` files map to the four ports on the basic serial line unit (SLU).

2.2.2 Changing Speed on the Console Device

On the VAXstation 2000 and MicroVAX 2000, `stty` cannot change the speed on the console device. The console subsystem firmware requires the console terminal to operate at a fixed speed of 9600 bits per second (bits/s) for a CRT or hardcopy terminal, or 4800 bits/s for a graphics display device, such as a VR260 monitor. The console device must be set for 8-bit character length with one stop bit and no parity. The `ss` driver enforces these restrictions by disallowing some functions of the `stty` command, such as changing the line speed on the console port. For more information, see `ss`(4) in the *ULTRIX Reference Pages.* These restrictions apply only to the console device.

2.2.3 Problem with the Xqdsg Server and the Grayscale Visual Class

If, when setting a visual class with the `-class` option to the `Xqdsg` server, the server is invoked with the `GrayScale` visual class, the Session Manager may crash.

2.2.4 XLoadQueryFont Function

The `XLoadQueryFont` function loads the specified font and returns a font structure specifying the properties of that font. An `XLoadQueryFont` function call causes the `Xqdsg` server to crash if the requested font is not available in compiled form.

To create compiled PCF fonts from uncompiled fonts in BDF format, do the following:

1. Run the `dxfc` command in that directory to compile the fonts. Put the output in a directory that is on the font path.
2. Use the `dxmkfontdir` command to create a `fonts.dir` file listing font names and font files for the X server.

2.3 VAXstation 3520 and VAXstation 3540

The following sections discuss the VAXstation 3520 and VAXstation 3540.

2.3.1 The X Server and Clients

The following sections discuss the X server:

- During the initialization of the X server, a cursor block may appear in the middle of the screen. As soon as the X server has started, this cursor disappears.
- The `ico` applications from MIT do not generate a faceted display on a 24-plane VAXstation 3520/3540. This is a problem within the `ico` application, as the `ico` application assumes that the default colormap is writable.
- If you draw wide dash lines one pixel long with projecting caps, the server may crash.
- Three-dimensional perspective projections from inside objects are not clipped properly.
- On the VAXstation 3520/3540 processors, some incompatibilities exist between shared-memory transport and 3-dimensional graphic applications. Do not use shared memory if you are running 3-dimensional graphic applications (`local:0` and `:0`). If you are running 3-dimensional graphics locally, the best

performance occurs when you use `iphost:0` over the TCP/IP network or `dnhost::0` when running over the DECnet network.

- The print screen option of the session manager does not work on 24-plane systems. It appears to work but, upon completion, does not create the output file.

2.3.2 The Xgb Server

The `Xgb` server uses a different font set, font compiler, color database, and PostScript Previewer than the other servers in this release. To avoid confusion, these components have been renamed for use with the `Xgb` server. These names are resolved during installation, but when running the font compiler or the Previewer you need to know the new names, which are listed in the following table:

Component	UWS Versions 2.0/2.1	UWS Versions 2.2(or higher)
Fonts	`/usr/lib/dwf`	`/usr/lib/dwf`
Font compiler	`/usr/bin/dxfc`	`/usr/bin/dxfc3d`
Color database	`/usr/lib/X11/rgb.*`	`/usr/lib/rgb.*`
PostScript previewer	`/usr/bin/dxpsview`	`/usr/bin/dxpsview3d`

2.3.3 Visual Classes Supported by the Xgb Server

The `Xgb` server only supports the `PseudoColor` visual class when setting a visual class with the `-class` option to the X server. The `Grayscale`, `StaticGray`, and `TrueColor` visual classes are not supported.

2.3.4 Console Messages on VAX Color Displays

When the X server is running and a console window is not provided, system messages that are sent to the console on 8-plane systems are displayed as lines beginning at the left edge of the screen.

2.3.5 Problem with the Xqdsg Server and the Grayscale Visual Class

If, when setting a visual class with the `-class` option to the `Xqdsg` server, the server is invoked with the `GrayScale` visual class, the Session Manager may crash.

2.3.6 XLoadQueryFont Function

The `XLoadQueryFont` function loads the specified font and returns a font structure specifying the properties of that font. An `XLoadQueryFont` function call causes the `Xqdsg` server to crash if the requested font is not available in compiled form.

To create compiled PCF fonts from uncompiled fonts in BDF format, do the following:

1. Run the `dxfc` command in that directory to compile the fonts. Put the output in a directory that is on the font path.
2. Use the `dxmkfontdir` command to create a `fonts.dir` file listing font names and font files for the X server.

2.4 DECstation/DECsystem 2100 and 3100

The following sections discuss the DECstation/DECsystem 2100 and DECstation/DECsystem 3100.

2.4.1 Getting a Memory Dump from a Hung DECstation/DECsystem 3100

If a DECstation/DECsystem 3100 hangs, you can press the reset button to enter console mode. The default action on the DECstation/DECsystem 3100 is for the reset to reinitialize memory. To prevent this (preserve memory), set the bootmode to debug by entering the following command in console mode:

```
>>> setenv bootmode d
```

Then, if a hang occurs, you can press the reset button to return to console mode (with contents preserved) and obtain a memory dump. The memory dump routine can be run by entering the `go` command with the following address:

```
>>> go 0x80030008
```

If the system is in multiuser mode when the reset button is pressed, then the dump will occur silently and no messages will be printed. The memory dump will take several minutes, then the console prompt will reappear. After the dump is completed, you can reinitialize the system and reboot as follows:

```
>>> init
>>> auto
```

Note

When bootmode is set to `d` it is important to enter `init` before entering `boot` or `auto` when the system has been shutdown to console mode or reset to console mode. Failure to use the `init` command may cause the system boot to fail.

2.4.2 Terminal Emulator Windows

There is a problem on the DECstation/DECsystem 2100 and the DECstation/DECsystem 3100 that manifests itself when the login shell is `/bin/sh` and the user is `root`. It may take as long as 3 minutes before terminal emulator windows appear.

You can work around this problem as follows:

1. Edit the file `/.profile`.
2. Move the line beginning with "echo ..." and reinsert it directly after the line beginning with "stty ..."

The file should now appear similar to the following:

```
#       @(#).profile    4.3     ULTRIX  11/18/88
stty dec crt new
echo 'erase ^?, kill ^U, intr ^C'
umask 22
PATH=/usr/ucb:/bin:/usr/bin:/etc:/usr/local:/usr/new:/usr/hosts:.
export PATH
```

2.5 DECstation and DECsystem 5000 Series Processors and DECsystem 5900

This section discusses the DECstation and DECsystem 5000 Series processors and the DECsystem 5900.

2.5.1 Personal DECstation 5000 Mouse Configuration

After an installation or upgrade, the mouse for the Personal DECstation takes some time to become configured. When the mouse finally comes online, you will see a message similar to the following:

```
0x54 A V1.0        DEC        VSXXXX-BB - 3884
```

2.5.2 Personal DECstation 5000 Series Floppy Device

The Personal DECstation 5000 processors have new devices for the floppy drive. The devices, found in `/dev`, are the following:

- The buffered device: `fd0`
- The raw device: `rfd0`

These devices are created during installation. Should you need to recreate the devices run the `MAKEDEV` utility:

```
MAKEDEV fd0
```

2.5.3 Keyboard and Mouse Behavior with SERIAL.bus

If your system is a Personal DECstation 5000 Series, you may encounter the following behavior when using the SERIAL.bus keyboard (LK501) and mouse (VSXXX-BB):

- Pressing Ctrl/Alt/Del causes the machine to be reset (as if you pushed a reset button) whenever the machine determines it is hung. Otherwise, pressing Ctrl/Alt/Del causes a CTRL/x to be generated. This is expected behavior.
- Pressing Ctrl/Alt/Ret causes the machine to be immediately halted, the current PC to be displayed, and a return of control to the ROM console. This is expected behavior.
- When the system is operating in single-user mode, holding down both shift keys causes the system to produce lowercase characters. After both shift keys are released, the system returns to the correct shift key functions. Shift keys, when held down independently, produce the correct results.

2.5.4 Installing Multiscreen

If you are installing a Personal DECstation processor with multiscreen that uses the baseboard video enabled, you must set the `-mg` option when setting the boot environment variable during the installation.

For example, the installation script asks you to set the environment variable with a

prompt like the following,

```
*** BOOTSTRAP COMMAND SEQUENCE ***

Issue the following console commands to set your default bootpath
variable and to boot your system disk:

     >> setenv boot "3/rz3/vmunix -a"
     >> boot
```

Instead of entering the command as it appears in the prompt, add the `-mg` option at the end of the command line as follows:

```
>> setenv boot "3/rz3/vmunix -a -mg"
```

2.5.5 Adding Support for Prestoserve

For information about adding support for Prestoserve to a DECstation/DECsystem 5000 Series processor or a DECsystem 5900 processor, see Section B.2 in the *Guide to Prestoserve.*

2.5.6 Floppy Disk Drivers and Default Density

The Personal DECstation 5000 Series floppy driver only supports using floppies at their default density. It is not possible to, for example, format a high density (1.44 Mbytes) floppy at double density (760 Kbytes).

Due to hardware constraints, specially-formatted media is required to get reasonable performance from these floppy drives. By default, the `fddisk` utility formats media for optimum performance in the system (about 30 Kbytes/second). If a floppy that has been formatted in another system is used, the performance will be about 2 Kbytes/second.

2.5.7 Setting an Application's Visual Class on a DECstation 5000

Some X11 applications that run on an 8-bit X server may not run properly on a DECstation 5000 with a 24-bit frame buffer. If the application crashes with a `BadMatch` protocol error, the problem may be related to the way the application sets its Visual class. Many applications simply select the default Visual with the `DefaultVisual` macro.

The default Visual class on 8-plane systems is `PseudoColor`. The default Visual class on 24-plane systems is `TrueColor`. `TrueColor` uses a read-only, statically allocated, direct color map.

One quick workaround is to change the default visual class with the `-class` option on the `Xwst3d` command line to `PseudoColor`.

However, it is likely you will also need to modify some applications to ensure that they select the appropriate visual on every X server. A good tutorial that describes how to approach this problem is "Visualizing X11 Clients" by David Lemke and David Rosenthal. It is available in the `doc/tutorials/visuals` directory on the X11 R3 and R4 release tapes. The "Xlib Programming Manual" by Adrian Nye (O'Reilly and Associates, Sebastopol, CA) also contains some explanation of this problem.

2.5.8 Limitations on Pixmap Sizes for Xwst3d and Xws Servers

On a DECstation with a PX or PXG-family graphics options, pixmap sizes are limited to 1024 pixels by 1280 pixels. Any application that attempts to allocate pixmaps larger than 1280x1024 pixels will get an insufficient resources error from the X server.

2.5.9 Running Applications Under the Xwst3d Server Without a Window Manager

If your application sends window properties to the `Xwst3d` server, the server may crash if the window manager is not running.

2.5.10 CDA Viewer

On DECstations with PXG or PXG Turbo options running a `Truecolor` visual, the CDA Viewer may display a `.DDIF` file incorrectly. To work around this problem, restart the server with a visual class of `PseudoColor`.

2.5.11 XDrawLine Function

If you use the `XDrawLine` function to define a line with a width of zero, a cap-style of `CapNotLast`, and coincident endpoints, the line will not be drawn. With the PX, PXG, or PXG Turbo options, the endpoints of `CapNotLast` lines are not drawn, and, as a result, lines are not drawn if the endpoints are coincident. To ensure that a line is drawn, either specify lines with a width of one or do not specify coincident endpoints.

2.5.12 XDrawSegments Function

The `XDrawSegments` function draws multiple, unconnected lines. If you use `XDrawSegments` to define line segments with a width of zero and a cap-style of `CapButt`, and define coincident endpoints for the line segments, the line segments are not drawn. Servers for the PX, PXG, and PXG Turbo graphics options treat all zero width lines as having a cap-style of `CapNotLast`. Because the endpoints of `CapNotLast` lines are not drawn, the lines are not drawn if the endpoints are coincident. To work around this, either specify line segments with a width of one or do not specify coincident endpoints.

2.5.13 Cardfiler

On DECstations with PX, PXG, or PXG Turbo options, the Cardfiler scroll bar and slider may become dashed lines rather than solid lines.

2.5.14 Viewing a PostScript File on DECstations with PX, PXG, or PXG Turbo Options

To view a PostScript file with the PostScript Previewer on these DECstations, choose the Sheet Selection dialog box from the Options menu and adjust the scale factor to 0.9 or smaller. To rotate a file, it is recommended that you use a scale factor of 0.7. Rotating a file at a scale factor above 0.7 may cause `dxpsview` to hang, cause the screen to freeze, or, infrequently, cause the server to crash.

The cause of this problem is that the servers for PX, PXG, and PXG Turbo graphics options cannot allocate pixmaps larger than 1280x1024 pixels. Due to a lack of server resources, the PostScript Previewer's normal mode of operation is not supported by the PX, PXG, or PXG-turbo hardware configurations.

Use the following procedure to preview files on these systems:

1. Invoke the previewer without specifying a PostScript file on the command line.

   ```
   % dxpsview
   ```

2. Select the Watch Progress option from the Options pull-down menu. This tells the Previewer to interpret the PostScript commands directly into the window (normally it draws to a pixmap, and copies the image to the window).
3. Select the file to preview using commands from the File pull-down menu.

Use the `watchProgress` resource to view PostScript files on DECstations with PX, PXG, or PXG Turbo options. This resource can be set in the PostScript Previewer profile file, `/usr/lib/X11/app-defaults/DPSViewer`, as follows:

```
*watchProgress: on
```

In addition, you can also set the resource in the system `Xdefaults` file `/usr/lib/X11/app-defaults/Xdefaults`, or in the user `.Xdefaults` file as follows:

```
DPSviewer*watchProgress:  on
```

In addition, the `dxpsview` command sets `watchProgress` from the command line as follows:

```
dxpsview -xrm "*watchProgress: on"
```

You can also select the Watch Progress option from the Options menu in the PostScript Previewer. A complete list of user-settable resources for the PostScript Previewer can found in `/usr/lib/X11/app-defaults/DPSViewer`.

2.5.15 Display PostScript on PXG-turbo Models

Halftoning in Display PostScript does not work correctly on PXG-turbo models and 8-bit color PXG models.

2.5.16 Exceeding the Per-Process Virtual Size Limit in DECstations with PX, PXG, or PXG Turbo Options

Opening multiple double-buffered and/or Z-buffered windows may crash the server. For each double buffer (or Z-buffer) of a window, the server allocates virtual memory adequate to hold the buffer's contents, so that the associated VRAM can be freed for other uses (such as pixmaps).

The amount of virtual memory required for several large buffered windows can be prohibitive. A full-screen Z-buffered window, for example, requires 4.8 megabytes of memory. A double-buffered version of the same window requires twice this amount (4.8 megabytes for each buffer). The default storage limit is 65 megabytes (text segment + data segment + stack size) per process, so creating eight such windows will cause the `sbrk()` system call to fail, crashing the server.

To avoid the problem, raise the per-process data segment size limit, defined in the system configuration file by `MAXDSIZ`.

2.5.17 Off-Screen Memory Limitations Involving Large Pixmaps

Simultaneous use of multiple large pixmaps results in slowed performance.

2.6 DECsystem 5100

The following sections discuss the DECsystem 5100.

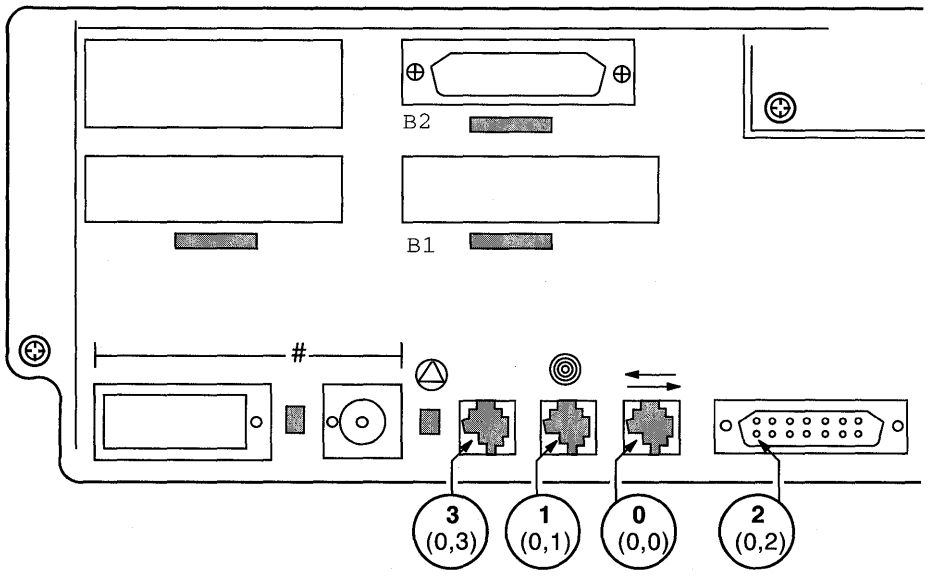
2.6.1 Backplate Labeling

The factory labeling of the console and terminal ports on the backplate of the DECsystem 5100 does not refer to the device major and minor numbers of the device special files made at installation time.

The device special file names can change at installation, but the major and minor numbers will always be tied to the hardware line number.

Figure 2-1 shows how each console and terminal port corresponds to the device major and minor numbers and the device special files made at installation time.

Figure 2-1: DECsystem 5100 Console and Terminal Ports



ZK-0224U-R

Table 2-1 shows the correspondence between the labels on the backplate of your DECsystem 5100 and the corresponding device major and minor numbers of the device special files made at installation time.

Table 2-1: DECsystem 5100 Console and Terminal Ports

Label	Installed ULTRIX Device Name	Major/Minor Number
0	/dev/console	0,0
1	/dev/tty00	0,1
2	/dev/tty01	0,2 (modem support)
3	/dev/tty02	0,3

Figure 2-2 shows how each console and terminal port corresponds to the device major and minor numbers and the device special files with the KN230 asynchronous communication option card added.

Figure 2-2: DECsystem 5100 KN230 with Async Terminal Ports

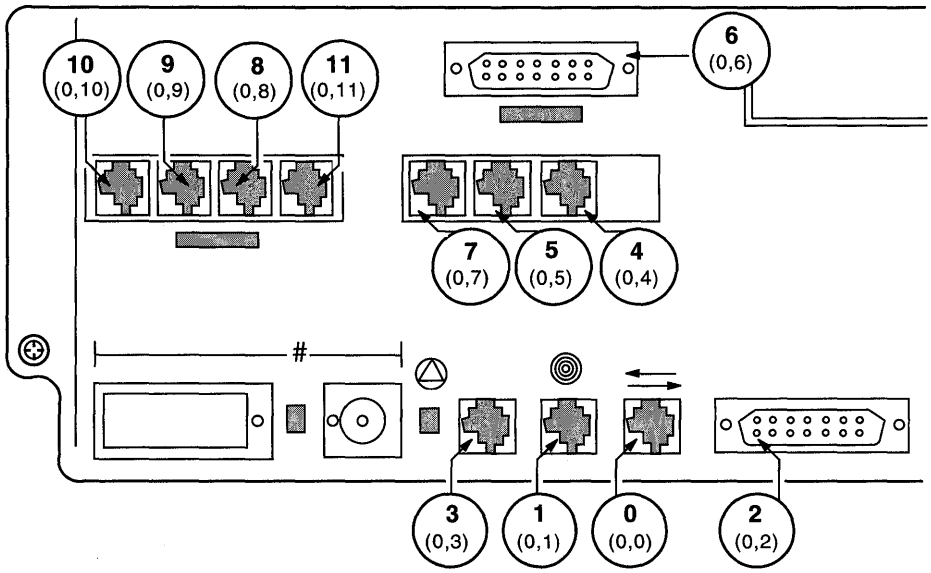


Table 2-2 shows the labels on the backplate of your DECsystem 5100 and the corresponding device major and minor numbers of the device special files if you have added the KN230 asynchronous communication option card.

Table 2-2: DECsystem 5100 KN230 with Async Terminal Ports

Label	KN230 ULTRIX Device Name	Major/Minor Number
0	/dev/console	0,0
1	/dev/tty00	0,1
2	/dev/tty01	0,2 (modem support)
3	/dev/tty02	0,3
4	/dev/tty03	0,4
5	/dev/tty04	0,5
6	/dev/tty05	0,6 (modem support)
7	/dev/tty06	0,7
8	/dev/tty07	0,8
9	/dev/tty08	0,9
10	/dev/tty09	0,10
11	/dev/tty10	0,11

2.6.2 Configuring Terminal Devices for the KN230 Asynchronous Communications Card

To configure terminal devices for the KN230 asynchronous communications card, follow these steps:

1. Add support for the new option card, as described in Section Section 2.6.3.
2. Log in as `root` or become superuser.
3. Rebuild the kernel to add support for the two new devices that reside on the KN230 asynchronous option card. To rebuild your kernel, enter the following command, replacing *HOSTNAME* with the name of your system typed in capital letters:

   ```
   # /etc/doconfig -c HOSTNAME
   ```

   The `-c` option specifies that the new kernel be built using the configuration file that already exists in the `/sys/conf/mips` directory.

   The `doconfig` command allows you to edit the configuration file. The following prompt appears immediately after you invoke the `doconfig` command with the `-c` option:

   ```
   Do you want to edit the configuration file (y/n) [n]?
   ```

   Answer yes to this prompt.
4. The `doconfig` program then places you in the `ed` editor. When you are in the `ed` editor, type the following sequence of commands to modify the configuration

file:

```
/mdc0/a
device          mdc1 at ibus?           vector mdcintr
device          mdc2 at ibus?           vector mdcintr
.
w
q
```

5. After you exit the `ed` editor, the `doconfig` program displays the following message as it begins to rebuild your kernel:

   ```
   *** PERFORMING SYSTEM CONFIGURATION ***
      .
      .
      .
   ```

   When the `doconfig` program finishes, it reports the location of the newly built kernel as follows:

   ```
   The new kernel is /sys/MIPS/HOSTNAME/vmunix
   ```

6. Move the new kernel to the `root` directory. Enter the following command, replacing *HOSTNAME* with the name of your processor in capital letters:

   ```
   # mv /sys/MIPS/HOSTNAME/vmunix /vmunix
   ```

7. The terminal devices are activated when you reboot your system using the new kernel. To reboot your system, use the `shutdown` command with the `-r` option and alert your users that the system will be going down, as follows:

   ```
   # /etc/shutdown -r +30 "Reboot to configure more terminal lines"
   ```

   For more information on shutting down your system, see the `shutdown`(8) reference page.

2.6.3 Adding Support for a New Option Card

The DECsystem 5100 CPU board (KN230 CPU board) provides support for one option card with two interrupts available for the expansion option. This allows the expansion option to have a maximum of two devices.

The Option ID Number is read from the OID (Option ID) register at startup time. This value must correspond to the value stored in the `iooption` console environment variable. To check or set the Option ID number, use the console environment commands, as follows:

```
>> printenv iooption
```

You must make an entry for each device in the kn230_option table in the `/sys/data/kn230_option_data.c` file. This provides a means of mapping the information needed to configure the devices to the ID value in the OID register. The first two entries in the kn230_option table of the `/sys/data/kn230_option_data.c` file are for the KN230 async card. You can use the following fields in these entries as guidelines when adding another option card. The fields in the kn230_option table that you need to use when adding another option card are as follows:

option ID number — The value that will be read out of the Option ID register for the option card.

driver name	The device driver name as it appears in the system configuration file.
type	The type is either "D" if the device driver uses a device uba structure or "C" if the device driver uses a controller uba structure. No other values are allowed.
expansion0 csr	The csr address of the device that will interrupt on the first line. If there is only one device on the card, it uses the first available interrupt; you place its csr address in the expansion 0 field.
expansion1 csr	The csr address of the device that will interrupt on the second line.
option_name	The option name string printed at boot time.

Note

Use only one expansion csr field for each entry; the other expansion card field should always be zero.

For example, assume that you are adding a new device driver called "new," which will be using only one interrupt. Because the driver will be using only one interrupt, you will need to make only one entry for the option card. In the following example, assume that the new option card has an ID number of 0x2. To add the device driver named "new," you would edit the kn230_option table in the `/sys/data/kn230_option_data.c` file as follows:

```
struct kn230_option kn230_option [] =
{
/* option driver       expansion 0  expansion 1        option name     */
/*  id#   name   type  csr address csr address           string       */
/* ====== ====== ====  =========== ===========  ====================  */

{  0x1,     "mdc",  'D',  0x15000000, 0x0, "Async comm (8 ports)"  },
{  0x1,     "mdc",  'D',  0x0, 0x15200000, "Async comm (8 ports)"  },
/* add additional option card devices here */
{  0x2,  "new",  'D',  0x15000000, 0x0, "New option card"       },
/*
 * DO NOT DELETE the Null entry which marks the end of the table
 * or your system will not configure properly.
 */
  {  -1,  "",  '0', 0,        0,   "0   }
};
```

2.6.4 Using the Halt Button on the DECsystem 5100

You can use the halt button on the back of the DECsystem 5100 to interrupt the ULTRIX operating system for the purpose of debugging the system.

However, to enable halts on the DECsystem 5100, you must first set the bootmode at the console level to halt. The default bootmode is set to reset, which will reset the system and run the diagnostics.

To set the bootmode at the console level to enable halts, enter the following

command at the console:

```
>> setenv bootmode h
```

After the ULTRIX operating system boots with halts enabled, you can press the halt button to get to the console level where system status can be debugged and evaluated. To return to the ULTRIX operating system at the point where you pressed the halt button, use the `continue` command at the console as follows:

```
>> continue
```

Note

Due to the hardware constraints of the system architecture, the halt interrupt cannot be guaranteed to interrupt a system hang and bring you back to the console level.

2.6.5 Characters Output to Terminal Line Connections on Power Up

When you power up the DECsystem 5100 and have terminals or printers attached to the asynchronous terminal ports, the hardware diagnostics output a series of characters to the attached terminals or printers. This problem does not affect the console port.

2.7 DECsystem 5400

The following sections discuss the DECsystem 5400.

2.7.1 Possible Performance Problem During N-Buffered I/O Use

If you use n-buffered I/O and the data buffers involved in the I/O are not properly aligned, you might experience a performance problem. Consider the page size of the underlying system architecture when you set up your data buffers. If you fail to consider the underlying page size, the effect of using n-buffered I/O is negated.

To avoid this performance problem, align all user buffers that are involved in n-buffered I/O on system page boundaries. On a DECsystem 5400, a system page boundary is 4096 bytes.

The following program fragment demonstrates using the `getpagesize` system call to obtain the system page size. The fragment uses the result to align the buffer for optimum performance.

```
int pgsize;
int bufsize = 512;
char *cp;

pgsize = getpagesize();
cp = (char *)sbrk( bufsize + pgsize )
cp = (char *)((unsigned)((unsigned)cp + pgsize ) & ~( pgsize - 1 ));
```

For applications that use `malloc`, you can use the `valloc` routine as a direct

substitute, as follows:

```
#include <stdlib.h>

extern size_t  bufsize;
char       *bufp;

bufp = (char *)valloc(bufsize);
```

For more information on allocating size bytes aligned on a page boundary, see the `valloc`(3) reference page.

2.7.2 Forcing a Crash Dump on the DECsystem 5400

If your system hangs for any reason, you can interrupt the operating system and force a crash dump. To interrupt the operating system, set the function switch on the CPU cover panel to the up (dot inside the circle) position. Then, press the break key, which sends an interrupt signal to the operating system. When the operating system processes the interrupt signal, it transfers control to the console program.

Once control transfers to the console program, the following appears on the display (the values on your system may be different from those shown here):

```
HALT PC 800dc968
Memory Size: 33554432 (0x2000000) bytes
Ethernet Address: 08-00-2b-0f-8e-42
>>
```

When the console prompt appears, issue the following command to force a crash dump:

```
>> go 0x80030008
```

Once the crash dump is complete, you can reboot your system. However, be sure to set the function switch to the down (no dot inside the circle) position before resuming the normal operation of your system. Failure to set the switch properly may result in accidental system interruptions.

2.8 DECsystem 5800

The following sections apply to the DECsystem 5800 processor.

2.8.1 Possible Performance Problem During N-Buffered I/O Use

If you use n-buffered I/O and the data buffers involved in the I/O are not properly aligned, you might experience a performance problem. Consider the page size of the underlying system architecture when you set up your data buffers. If you fail to consider the underlying page size, the effect of using n-buffered I/O is negated.

To avoid this performance problem, align all user buffers that are involved in n-buffered I/O on system page boundaries. On a DECsystem 5810, a system page boundary is 4096 bytes.

The following program fragment demonstrates using the `getpagesize` system call to obtain the system page size. The fragment uses the result to align the buffer for

optimum performance.

```
int pgsize;
int bufsize = 512;
char *cp;

pgsize = getpagesize();
cp = (char *)sbrk( bufsize + pgsize )
cp = (char *)((unsigned)((unsigned)cp + pgsize ) & ~( pgsize - 1 ));
```

For applications that use `malloc`, you can substitute the `valloc` routine as follows:

```
#include <stdlib.h>

extern size_t  bufsize;
char      *bufp;

bufp = (char *)valloc(bufsize);
```

For more information on allocating size bytes aligned on a page boundary, see the `valloc`(3) reference page.

2.8.2 Interrupting the Operating System on a DECsystem 5800 Series Processor

If your system hangs for any reason, you can interrupt the ULTRIX operating system when it is running on a DECsystem 5810. To interrupt the operating system, set the top key switch to the ENABLE position. Then, press Ctrl/P, which sends an interrupt signal to the operating system. When the ULTRIX operating system processes the interrupt signal, it transfers control to the console program.

Once control transfers to the console program, enter the following at the console prompt to force a crash dump:

```
>> d %ra 0x80030008
%ra:    0x80030008          - 2147287032  '\b'
>> continue
dumping to dev 909, offset 65548
Dump of 8190 pages
   .
   .
   .
```

Once the crash dump is complete, you can reboot your system. However, be sure to set the Key switch to the SECURE position before resuming the normal operation of your system. Failure to set the switch properly may result in accidental system interruptions.

If you interrupt your ULTRIX system and immediately decide to return control from the console to the ULTRIX system, issue the `continue` command. You must issue this command before you issue any other command at the console prompt. If you issue a command other than the `continue` command, you must reboot your system to return control to the ULTRIX system.

Communication with the console terminal may stop after you issue the `continue` command. If communication stops, press the Return key. Pressing the Return key reestablishes communication and allows your system to operate normally.

ULTRIX Software Notes 3

This chapter discusses issues and known problems with the software and, when possible, provides solutions or workarounds to the problems.

The following topics are discussed in this chapter:

- User Commands
- Administrative Commands
- System Calls
- Library Routines
- Security Notes
- DECrpc
- Message Handler (MH) Notes
- Mail
- Network and Communications
- Printing
- Customer Device Drivers

3.1 User Commands

The following sections discuss issues and known problems with user commands.

3.1.1 The adb Shell Scripts are Outdated

The `adb` shell scripts in `/usr/lib/adb` need to be updated to the current release.

3.1.2 The ar Command (RISC)

When used to extract all files from an archive, `ar` creates a file named `__________ELEL__` with permissions 000 (it is a symbol table that is automatically created by `ar`). If a second `ar` command is run with the `x` key, `ar` displays the following message:

```
ar: Error: __________ELEL__cannot create
```

You can ignore this message. You can avoid receiving this message by deleting the `__________ELEL__` file.

3.1.3 The as Command (RISC)

An assembly language program can have more text instructions than the assembler can handle. To resolve this, divide the file in half.

3.1.4 The cpio Command

When you use the `-c` option to archive files from an NFS-mounted file system, the archive produced will be corrupt. The only way to prevent the problem is to first copy the NFS-mounted files to a local file system and then archive them.

3.1.5 The find Command Searches Remote Mounted File Systems

The `find` command does not have an option to restrict search to locally mounted file systems.

To restrict a search to the local file sytem, create a text file that contains the pathnames of the directories you want to search. In the following example, the `sh` command line performs the search:

```
for i in 'cat filelist'; do df $i | grep -s /dev && find $i -perm 01000 \
-print ; done
```

The `df` command, when provided a command name, returns a mount point that is prefixed by `/dev` if it is a locally mounted file system. You must use a text file to list the pathnames because `df` returns a mount point name that differs from the directory name requested, and `find` does not search `/dev` directories. This solution may not work in some instances.

3.1.6 The s5make Command

The `s5make` command does not rebuild a source file if the source file is a new link to an old file.

3.1.7 The sccs Command

The `sccs` command behaves incorrectly in the following situations:

- When the `sccs` command is issued with a flag, but without a command, a segmentation fault occurs.
- The `sccs get -e` command takes too long to return an error when the SCCS file is write protected.
- The `sccs unedit` command should have the `-r` option added.

3.1.8 The sed Command

The `sed` command does not handle all possible incorrect expressions properly. The following command line syntax can cause a core dump:

```
sed "s//"
sed "s//abc
```

3.1.9 The size Command Messages

The `size` command on RISC machines can generate the following error messages:

```
ldopen: cannot read magic number filename
size: cannot open filename
ldinitheaders: magic number incorrect (0x0)
size: cannot open filename
```

These errors have the same meaning as the following VAX-based `size` message:

```
size: filename not an object file
```

3.1.10 The tar Command

There are four `tar` problems:

- When a TK50 multivolume tar set is read on a TK70 drive that is not SCSI, the `tar` command reports a read error, then terminates without requesting that you load the next volume.
- If a multivolume tar set contains a file that finishes exactly at the end of tape, the `tar` command reports a directory checksum error when the file is read.
- When a multivolume tar set is being read and the media is loaded out of sequence, the `tar` command rejects the rest of the media as out of sequence whether it is or not.
- When an input file specification to the command has an embedded symbolic link, the `tar` command fails to archive any files after the first embedded symbolic link.

There are no workarounds to these problems.

3.2 Administrative Commands

The following sections discuss issues and known problems with administrative commands and utilities.

3.2.1 The restore Command

The following sections describe `restore` command problems.

3.2.1.1 Spurious Problems During Extracts

During an `x` command or an interactive extract operation of files that exist on later volumes of a multivolume tape, the `restore` command displays the following error messages:

```
Tape is not a dump tape
Read error while skipping inode
```

You can force the extract to complete successfully by continuing the extract after these errors. Probably, the spurious errors will recur several times before the extract completes.

3.2.1.2 Symbolic Links Restored Incorrectly

A restore operation always restores symbolic links with the incorrect owner.

3.2.2 The crash Utility

The following sections discuss the crash dump facility.

3.2.2.1 Crash Dumps and the ps Command

Using the `ps` command on system crash dumps may not display the command arguments and the user environment.

The new system crash dump strategy does not dump user data for the default case of a partial dump strategy. The `ps` command determines whether it is examining a system crash dump produced by the partial dump strategy. If this is true, the command makes no attempt to acquire the process' command line arguments or the process' environment strings.

The only workaround when the process argument strings and/or the process environment strings are required is to use the full dump strategy.

3.2.2.2 Dump Device Configuration Restrictions

The crash dump facility restricts the type of dump device configurations permitted for the `rl`, `rk`, and `hp` devices. This restriction is identical to that for the `ra` type disks. For crash dumps to occur, the device that is specified in the host system configuration file as the dump device must be configured on the controller that will be used to boot the system device.

3.2.3 Effects of New File System Timeout Algorithm on fsck

The policy of changing the value of the clean byte to ensure checks of earlier file systems was replaced in Version 4.2 and higher systems with a new system timeout algorithm. This scheme limits the amount of time a file system is believed clean, no matter how the file system was mounted.

The timeout factor is initially set to 20 and is decremented when any one of the following three events occur:

- A file system is mounted
- 10,000 updates have occurred
- A file system was updated and `fsck` occurred more than 60 days earlier

When the timeout factor reaches zero, the following message is printed, and the next invocation of `fsck -p` will check the file system indicated in the message:

```
Warning, /dev/rxxx has exceeded %d %s threshold, fsck(8) is advised
```

The `%d` is replaced by the default factor, and `%s` is the event that crossed the threshold.

For example, if the timeout factor is 20, and a file system is mounted 20 times, the

final mount will produce the following:

```
Warning, /dev/rxxx has exceeded 20 mount threshold, fsck(8) is advised
```

This message is a warning. The mount will succeed, and operations to the file system will continue. However, the invocation of `fsck -p` will check the file system.

The timeout factor can be set on an individual file system. When a file system is made by `newfs` or `mkfs`, a default value of 20 is used. However, if timeouts occur too frequently or infrequently, the factor can be altered with `tunefs`. Refer to the `tunefs`(8) reference page for more information about changing the value of the clean byte timeout factor.

3.2.4 Deletion of Formatted Reference Pages with /etc/crontab

The default `/etc/crontab` file installed on DMS clients and on workstations deletes formatted reference pages from local (ufs-mounted) `/usr/man/cat?` directories (if they exist), if the formatted reference pages have not been accessed in 28 days. It does this on the first day of each month, at least 28 days after the last execution of the `catman` command.

If you want to disable this feature, you can either remove the line in the `/etc/crontab` file that deletes the reference pages, or comment out that line. To comment out that line, place a number sign (#) in column one before the following entry:

```
15 2 1 * * for i in /usr/man/cat[1-8]; do df $i | \
grep -s /dev && find $i -type f -atime +28 -a -exec rm {} \; ; done
```

The backslash (\) at the end of the first line in this example does not actually exist in `/etc/crontab`. It is used here to indicate only that this is a very long line.

If you want to enable this feature on a server system, add that line to the server's `/etc/crontab` file.

To protect the `/usr/man/cat?` directories on server systems from client systems that are running ULTRIX-32 V3.1 or older software, run `catman` on all reference page sections, then set the permissions on the `/usr/man/cat?` directories to 775, or export `/usr/man` with read-only permissions.

3.2.5 Errors with tapex Utility

The following sections discuss the `tapex` utility.

3.2.5.1 Failures Using SCSI TZK10 Tape Drive

The `tapex` utility was originally written for variable length record tape devices. The TZK10 (QIC) tape drive uses fixed length records, which causes all variable length tests to fail. Many of the `tapex` tests can still be run with the TZK10, but tests such as the Random Record Size test, which you specify with the `-g` option, will fail with the TZK10.

In addition, the Append To Media test, which you specify with the `-d` option, also fails with the TZK10, because the TZK10 tape drive does not support data overwrite.

When the Append To Media test fails, it generates the following errors:

```
Append to media testing.

This test simulates the behavior of the "tar r"
command by writing 20 records to the tape.
Next the tape is repositioned back one record and then
20 more records are written.
All records are of size 10240.
Finally the resulting tape read in for verification.

Aborting this test due write errors when trying to
append records to the media.
ERROR: 20 write errors occurred.
```

3.2.5.2 Record Size Validate Errors on DECsystem 5100

The Record Size test fails on DECsystem 5100s with "validate errors" because the SCSI `sii` driver copies back more bytes than were actually read. In this test, the `tapex` utility initializes its read buffer to a known value, then attempts to read 1010 bytes to records of 1000 bytes. After the read, all bytes including the ten extra bytes are validated, causing the following validate errors:

```
Performing record size testing.  This test verifies that
at most one record is returned by a read system call.

Record size subtest #1:
Test read requests larger than the record size.
Request a read of 1010 bytes to records of size 1000 bytes.
The following errors were encountered when trying to
read more than a full record.  Read errors indicate
that more than a full record has been returned.
Read errors could also indicate that fewer bytes than
requested were returned.
FAILURE: 2000 validate errors
```

3.2.6 System Exerciser and syscript

Some of the system exercisers in `/usr/field` require their log files to be local to the exerciser. Because the `/usr` file system should be mounted read-only in a diskless environment, this prevents the creation of the clients' log file.

The workaround is to copy the desired exerciser to `/var/tmp` before executing it. This will move the exerciser and its associated log file into the clients' writable root area.

To run the `syscript` script, you should edit the file and globally change `/usr/field` to `/var/tmp/field`.

3.3 The ptrace System Call

Programs using `ptrace` to write into the instruction space of a traced program prevent that image file from being executed until the traced program has terminated. See `ptrace`(2), `dbx`(1), and `adb`(1) in the *ULTRIX Reference Pages.*

3.4 Library Routines

The following sections discuss library routines.

3.4.1 Nonblocking Input from a Terminal

Nonblocking input from a terminal (setting FNDELAY) does not function if an application is linked with the `pcrt0.o` startup routine.

3.4.2 Undocumented Internal Functions of dbm.a Removed

The dbm.a library has been modified to use the ndbm.a library. Several undocumented functions that were strictly internal to the old dbm.a library, and were not supported, no longer exist. These include the following: additem, calchash, chkblk, cmpdatum, dbm_access, dbmflush, delitem, firsthash, getbit, hashinc, makdatum, and setbit.

Code utilizing the dbm.a library should be independent of the specifics of the hashing algorithms used by dbm.a.

3.4.3 Support for Multiple Databases

The `ndbm` library is provided as part of the libc library to augment the `dbm` functionality. The `ndbm` function allows multiple databases to be concurrently accessed. Existing `dbm` calls can either be replaced by `ndbm` calls or they can continue to use `dbm`, which has remained in the libdbm library. See the `dbm`(3) and `ndbm`(3) reference pages for more information.

3.4.4 Correct Declarations for Functions in <math.h>

The following functions are a part of the ULTRIX math library but are not part of the ANSI C standard or X/OPEN math libraries. Hence, these functions are not declared in <math.h> as described in the reference pages. Declare the following functions as needed in your programs:

```
typedef struct { double r, i; } complex_double;
typedef struct { double r, i; } complex_float;

double acosh(double);
double asinh(double);
double atanh(double);
double cabs(complex_double);
double cbrt(double);
double copysign(double, double);
double drem(double, double);
double expm1(double);
float facos(float);
float fasin(float);
float fatan(float);
float fatan2(float,float);
float fcabs(complex_float);
float fceil(float);
float fcos(float);
float fcosh(float);
float fexp(float);
float fexpm1(float);
float ffloor(float);
float fhypot(float, float);
```

```
int   finite(double);
float flog(float);
float flog10(float);
float flog1p(float);
float fsin(float);
float fsinh(float)
float fsqrt(float);
float ftan(float);
float ftanh(float);
float ftrunc(float);
double log1p(double);
double logb(double);
double rint(double);
double scalb(double, int);
double trunc(double);
```

3.4.5 The qsort Function and Comparison Routines

When you supply a comparison routine to `qsort` that can return different results for the same pair of keys at different times, `qsort` does not always execute properly. The `qsort` function may write beyond the bounds of the array being sorted.

The workaround to this problem is to provide a comparison routine that will consistently return the same result for any given pair of keys.

3.4.6 The lint Library strncmp Function

In the `lint` libraries, the third parameter to the `strncmp` function is incorrectly declared to be of type `int` instead of type `size_t`.

3.4.7 A printf Problem (RISC)

The `printf %f` format (and the `fcvt` function) incorrectly rounds down if all digits after the decimal point are zero. For example, with a %.1f format, the number .07 incorrectly prints as 0.0 instead of 0.1, but the number .17 prints as 0.2, the correct value.

3.4.8 The j1 Function Returns Incorrect Value (RISC)

The `j1()` function does not return ERANGE when the input argument is too large.

3.5 Security Notes

The following sections discuss ULTRIX security.

3.5.1 Auditing the adjtime System Call

The `utc_adjtime` system call is not audited by default. An entry does not exist in the `/etc/sec/audit_events` file for `utc_adjtime`. In addition, the `auditmask` utility with the `-f` option does not turn on audit for the `utc_adjtime` system call.

If an audit of `utc_adjtime` is needed, type the following line into the

/etc/sec/audit_events file:

```
utc_adjtime     succeed fail
```

When the /etc/rc.local file is executed, the auditmask utility will adjust the system auditmask file with the information provided in the /etc/sec/audit_events file.

To initiate the audit of utc_adjtime on currently running system, log into the root account and type the following:

```
auditmask utc_adjtime
```

3.5.2 Login and Security Restrictions

There are limitations with remotely served authorization (auth) databases.

The auth database cannot be served through YP, only by BIND/Hesiod. Do not attempt to serve the auth database through YP. Instead, use svcsetup and secsetup to specify your services.

When auth entries are served through BIND/Hesiod, login fail count maintained by login is not supported. Login attempts that fail on BIND clients will not increment the login fail count if the account is not on the local machine. Repeated login failures for any account are always recorded in the local system log.

When the security features are enabled, it is not possible to su to root on lines that are not marked as secure in the /etc/ttys file.

3.5.3 Activating and Configuring the Audit Subsystem

The /usr/adm/auditlog file is the default fall-back destination for logging information. The discussion of a fall-back destination for logging information in Section 4.5 of the *Security Guide for Administrators* should contain the following information.

If no fall-back destination for the audit log is specified, or if the fall-back destination is specified but unavailable, the auditd daemon defaults to its builtin fall-back destination, the /usr/adm/auditlog file.

Always keep the /usr/adm/auditlog file available for potential use by the auditd daemon.

3.6 DECrpc rrpc Routines Require an Explicit Call into the Entry Point Vector Table

On the client side, because of the way the rrpc_ calls are defined and implemented in the libnck.a run-time library, you must explicitly call into the entry point vector table for the rrpc_ interface to send an rrpc_ request across the network. The following is an example of a call that works as desired:

```
(*rrpc_$client_epv.rrpc_$inq_interfaces)(handle,
            (unsigned long) max_ifs, ifs, &l_if, &status);
```

The server side stub routines call the entry point rrpc_$inq_interfaces on behalf of the client. The results of the call are then passed back to the client.

3.7 Mail

The following sections discuss the various mail utilities.

3.7.1 sendmail Unable to Handle Large Numbers of Recipients

The `sendmail` program may return an error message when mail is sent to large lists of recipients or large distribution lists (1000 or more). The error message is:

```
Cannot exec '/bin/mail' errno=14
```

This error is caused by a restriction in `/bin/mail`. To resolve this problem, edit the local mail channel in the `sendmail` configuration file, `/usr/lib/sendmail.cf`. The local mailer is specified on the line that begins with "Mlocal," while the flags are specified as "F=..m.." and so on. Remove the `m` from the list of flags, refreeze the configuration file, and restart `sendmail`.

Note, unless needed, the `m` flag should not be removed as it can cause an extra load on your system by invoking `/bin/mail` separately for every local recipient on any recipient list.

3.7.2 sendmail Does Not Set the $x Macro on Received Mail

The `$x` macro, which represents the full personal name of the sender, is not set by the `sendmail` program when the sender is not local. Users who receive mail through the DECnet-Internet Gateway do not see the personal name of the person who sent the message.

This behavior is due to the way mail headers are constructed and cannot be changed. The `sendmail` program uses information in the `/etc/passwd` file to get personal names; the program has no access to remote password files.

3.7.3 Creating Aliases that Exceed 1024 Characters in /usr/ucb/mail

The `/usr/ucb/mail` program limits private user aliases to 1024 characters.

The string that you are aliasing in your `.mailrc` file cannot exceed 1024 characters. Aliases defined in a `.mailrc` file that exceed this length limit cause the `mail` program to core dump through a segmentation fault.

To work around this problem, redefine the long alias using aliases that are less than 1024 characters each. The original alias can then be constructed using those smaller aliases.

```
alias a a1,a2,a3...aN (list 1 of less than 1024 characters)
alias b b1,b2,b3...bN (list 2 of less than 1024 characters)
alias c a,b (complete alias list)
```

For example, assume that you want to define an alias c that exceeds 1024 characters in length. First, break alias c into aliases a and b. Then define alias a to include list 1 and alias b to include list 2. Finally, define alias c to include the lists that are defined by aliases a and b.

3.7.4 sendmail Address Parsing Problem

There is a problem with the sendmail configuration file rules for parsing addresses of

the form:

```
nnnn::uuuu@hhhh
```

If this address is interpreted according to the RFC822 specification, the canonical form is:

```
nnnn::uuuu<@hhhh>
```

This is often incorrect if the mail actually originated from `uuuu@hhhh` and passed via `nnnn` on the way to the receiver's host. This typically happens when two ULTRIX users communicate via the DECnet mail mailer because the recipient has a `.forward` file on host `nnnn` specifying a DECnet mail forwarding address.

If this is the case, then the required canonical form is:

```
uuuu@hhhh<@nnnn.enet>
```

Implementing this functionality violates the RFC822 specification. To help resolve this problem, the following rule is included in rule set 3:

```
R$-::$+<@$~S>      $1::$2@$3          defocus - not local host
```

This rule switches between the two forms of the canonical address depending on whether the host `hhhh` is known to be a local host (one listed in `/etc/hosts` or `/etc/hosts.local`) or not.

If `hhhh` is local, the canonical form is assumed to be:

```
nnnn::uuuu<@hhhh>
```

Otherwise it is:

```
uuuu@hhhh<@nnnn.enet>
```

This presumes that local users will use tcp mail (user@host) rather than using DECnet mail (host::user) to forward mail.

It is recommended that users forward mail using tcp local mail (user@host) when possible. If full adherence to RFC822 is required, then the preceding rule can be commented out of rule set 3.

3.7.5 sendmail Sender Name Problem

The `sendmail` program uses information in `/etc/utmp` to determine the contents for the `From:` line in outgoing mail that has originated locally.

The contents of `/etc/utmp` change frequently and `sendmail` can, occasionally, read inconsistent data. This results in bad `From:` line specifications on outgoing mail.

3.8 Network and Communications

The following sections discuss network and communications.

3.8.1 Disabling LAT tty Lines

After a LAT tty line state is changed from on to off in the `/etc/ttys` file and after the `kill -1 1` command is executed, a connect to the tty is still accepted. The symptoms are a system hang before a login prompt can be produced.

Should the problem occur, you have to clear the tty status to resume normal operations. To do so, run the `file` command on the tty. For example, to clear

tty32:

```
> file /dev/tty32
```

3.8.2 Using Symbolic Links with /etc/exports and /usr/etc/mountd

Check your `/etc/exports` files for paths that contain symbolic links because the mount daemon for Version 4.2 and higher systems does not allow clients to mount these paths. For example, if `/var` is a symbolic link to `/usr/var` on your server and you want to export it, the `/etc/exports` file should contain the full path `/usr/var`. Clients can still request to mount `/var`, because the mount daemon expands the requested path to `/usr/var` and checks for that in the `/etc/exports` file.

3.8.3 The ne Network Device

The `ne` (SGEC) device driver reports more packet collisions than the `ln` (LANCE) driver under the same network environment. The LANCE chip can only report at the most two collisions when transmitting a packet. The SGEC chip reports the real counter of the collisions, from 1 to 15.

As a result, the `netstat` command with the option `-i` will report a higher collision rate when run on an `ne` network device.

3.8.4 Writing to a Remote a.out File

If a remote a.out image file is written to on a server while one or more clients are using that image file, further references on that file by the currently executing client processes will cause those processes to be killed. Under these conditions, the system responds with the message:

```
pid <number> killed due to text modification
```

The *<number>* argument is the PID number of the process that was killed.

If another process is started on a given client while processes on that client are being killed, the new process fails and the system responds by displaying the message:

```
remote text modified and not yet cleaned up
```

In this case, retry the process.

3.8.5 Maintaining the BIND/Hesiod Root Name Server Data File

There are currently seven BIND root name servers. These servers know about all the top-level BIND domains on the Internet network. It is necessary to know about these servers when making queries about hosts outside of your local BIND domain. Be aware that the host names and IP addresses of these machines do periodically change. It is imperative that these changes are reflected in the date file of the BIND/Hesiod root name server, `/var/dss/namedb/named.ca`.

To maintain the file, at least once a month connect to the system nic.ddn.mil which has an IP address of 192.112.36.5, and is managed by the Network Information Center. Use the `ftp` command with a login "anonymous" and password "guest." Retrieve the `NETINFO:ROOT-SERVERS.TXT` file, and examine it against your existing `named.ca` file. If any differences exist, incorporate them into the existing format in the `named.ca` file.

For example, the following `ftp` session is from host `chicago.cities.dec.com` with IP address 128.11.22.33 and user name jones:

```
chicago.cities.dec.com> ftp nic.ddn.mil.
Connected to nic.ddn.mil.
220 NIC.DDN.MIL FTP Server Process 5Z(47)-6 at Wed 11-Apr-90 08:24-PDT
Name (nic.ddn.mil.:jones): anonymous
Password (nic.ddn.mil.:anonymous):
331 ANONYMOUS user ok, send real ident as password.
230 User ANONYMOUS logged in at Wed 11-Apr-90 08:24-PDT, job 40.
ftp> get netinfo:root-servers.txt /tmp/root-servers
200 Port 11.175 at host 128.11.22.33 accepted.
150 ASCII retrieve of TS:<NETINFO>ROOT-SERVERS.TXT.18 (1 page) started.

226 Transfer completed. 673 (8) bytes transferred.
local: /tmp/root-servers remote: netinfo:root-servers.txt
673 bytes received in 0.09 seconds (7.3 Kbytes/s)
ftp> quit
221 QUIT command received. Goodbye.
chicago.cities.dec.com>
```

If you change your `named.ca` file, keep a copy of the original file and name it `named.ca.nic`. You should do this because if the `bindsetup` command is rerun, it will overwrite the `named.ca` file. If the `bindsetup` command is rerun, be sure to copy back the `named.ca.nic` file to `named.ca`.

If you are unable to `ftp` to `nic.ddn.mil`, send mail to `hostmaster@nic.ddn.mil` or call the NIC's toll-free number, (800) 365-3642.

3.8.6 Placement of Yellow Pages Master Files

YP master files must be placed in the directory `/var/yp/src`.

3.8.7 Yellow Pages Server Environment

If the number of groups that `root` belongs to exceeds eight, the YP service may not work correctly in an environment where ULTRIX servers in Version 4.2 and higher are mixed with earlier ULTRIX servers. To avoid this, make sure `root` belongs to eight or less groups.

3.8.8 Added Initial Bind Option (-X) to ypbind

The Initial Bind option (-X) forces `ypbind` to bind to a YP server at the time that the `ypbind` command is executed, instead of waiting until Yellow Pages is used. If no server is available, `ypbind` will try for several minutes and then exit.

The Initial Bind (-X) Option enables a system that does not exclusively depend on YP to boot and to allow logins when there are no YP servers available. Without this option, such a system would hang.

3.8.9 Protecting YP and BIND/Hesiod Files and Directories

To protect your YP maps from other users, change the modes of your `/var/yp/<` *your Domain name* > directory to 700 (drwx------).

To protect your BIND/Hesiod databases specified in the file `named.boot` file on secondary servers from other users, change the modes of the database files in the `/var/dss/namedb` directory to 600 (-rw-------) .

To protect your YP and BIND/Hesiod source files on your YP master and BIND/Hesiod primary server, change the modes of your `/var/yp/src` and your `/var/dss/namedb/src` directories to 700 (drwx------).

3.8.10 Improve Your Yellow Pages Makefile

In the beginning of the Makefile are variable definitions; the following line does not expand properly:

```
DOM='domainname'
```

As a result, whether or not a YP source file has been changed, the map will always be updated.

To work around the problem of unnecessary map updates, modify the `/var/yp/Makefile` and substitute your YP domain name. For example, if your YP domain name is "yourYPdomainname," the new line looks like this:

```
DOM=yourYPdomainname
```

3.8.11 Recommendation for Placement of NFS Mount Points

It is recommended that you do not place NFS mount points to different servers in the same directory. If mount points to different servers are placed in the same directory and one of the servers is hard mounted and goes down, all NFS requests will hang until the server that is down comes back up.

When computing the pathname string of a directory, `getwd` moves up the directory tree from the current working directory to the root. When `getwd` passes through a mount point, it will check entries in the directory until it finds the mount point it just traversed. If any of the entries in the directory are mount points to a server that is hard mounted and down, `getwd` will block until the server responds.

3.8.12 Denied Access to Files Over NFS

If you attempt to access a file over NFS and are denied access with your current user id, and you then change to the user id that allows you access, you may still be denied access to the file for a short period of time.

3.8.13 NFS Security

If you are concerned with NFS security, use the `nodev` option to mount all UFS file systems exported via NFS. Use the `nosuid` option to mount all UFS file systems exported via NFS with the "map client root uid to 0 export option (-r=0)." See the `mount`(8nfs) and `exports`(5nfs) reference pages for more information.

3.8.14 NFS Mounts to Pre-Version 4.2 NFS Servers

Increasing the number of groups that a user is in beyond eight can affect services that use SUN RPC. The version of SUN RPC shipped with ULTRIX Version 4.2 and higher products supports up to 32 groups. Previous versions of ULTRIX supported up to eight. Users that increase the number of groups that they are a member of to more than eight will not be able to NFS mount file systems from ULTRIX servers older than Version 4.2. This incompatibility can only occur if a user or `root` group membership is increased to more than eight groups. See the `group`(5) reference page for more information.

Reduce the number of groups that you are a member of to eight or less to make NFS mounts to pre-Version 4.2 NFS servers work.

3.9 Printing

The following sections discuss the print software.

3.9.1 Filters Not Available on RISC Processors

The following filters are not available on RISC processors:

- `/usr/lib/lpdfilters/regis_ps`
- `/usr/lib/lpdfilters/tek_ps`

3.9.2 Configuring the System for an LA324 Printer

As there is no entry for the `la324` printer type in the `lprsetup` program, select the `lj250` printer type to add an LA324 print queue to your `/etc/printcap` file. When prompted for a symbol name, enter `fs` and then change `03` to `023`. There are no other changes to make.

The following shows a sample entry for an LA324:

```
# This is for an LA324
lp5|5:\
        :af=/usr/adm/lp5acct:\
        :br#4800:\
        :ct=dev:\
        :fc#0177777:\
        :fs#023:\
        :if=/usr/lib/lpdfilters/lj250of:\
        :lf=/usr/adm/lp5err:\
        :lp=/dev/tty05:\
        :mc#20:\
        :mx#0:\
        :pl#66:\
        :pw#80:\
        :rw:\
        :sd=/usr/spool/lpd5:\
        :sh:\
        :uv=4.0:\
        :xc#0177777:\
        :xf=/usr/lib/lpdfilters/xf:\
        :xs#044000:\
```

3.9.3 Printing Without an Input Filter

If an entry in the `/etc/printcap` file does not supply a value for either the input filter, `:if:`, or for the output filter `:of:`, jobs that take more than one minute to print will fail and then retry indefinitely.

To avoid this, specify `:of=xf:` in the `/etc/printcap` file.

3.9.4 Removing Current Jobs

If current jobs are removed using `lprm` the print queue may be left with no daemon present. To restart the default queue, enter:

```
% lpc restart lp
```

To restart a specific queue, enter:

```
% lpc restart queuename
```

This problem may also arise if the queue is remote, in which case users must issue the commands on the remote host.

Queues can be restarted by submitting more print jobs.

3.9.5 Truncated Lines with lpr -p

Using the `lpr -p` option to print a document can cause lines on the printed page to be truncated if the document contains embedded tabs. These are tabs that occur anywhere except at the start of a line. This is due to incorrect behavior in the `/bin/pr` utility.

To enable the `pr` utility to expand these tabs into spaces and to calculate the line length correctly you need to make some changes. For all print queues, except those for PrintServers, set the `pp` field in the `/etc/printcap` file to `/bin/pr -e`. Use the `modify` option within the `lprsetup` program to do this.

For all PrintServer queues edit the `/usr/lib/lpdfilters/pr_call` file and change the line:

```
$DEBUG_ECHO /bin/pr -w$width -l$length -h $title
```

to read:

```
$DEBUG_ECHO /bin/pr -e -w$width -l$length -h $title
```

You do not need to make any changes for print queues to remote printers.

3.10 Customer Device Drivers: Recompile Potential

Customer device drivers might require recompiling and relinking if they use the kernel memory allocator interface defined in `/sys/h/kmalloc.h`.

Customer device drivers that use the memory allocator interface defined in `/sys/h/kmalloc.h` will no longer work if the KM_ALLOC, KM_FREE or KMEM_DUP macros are used. The allocator C routines `km_alloc`, `km_free` and `km_memdup` support the same formal arguments as in earlier versions of the operating system, and hence do not require changes to their usage. The macros have retained their original formal arguments; however, the macro implementation has changed.

A workaround for drivers that use the macros defined in `/sys/h/kmalloc.h` is to recompile dependent modules and then relink the kernel.

ULTRIX Worksystem Software Notes 4

This chapter discusses known problems with ULTRIX Worksystem Software and, when possible, provides solutions or workarounds to the problems. It also presents useful information that may not be located elsewhere.

The notes in this chapter cover the following topics:

- Miscellaneous UWS Problems
- The X Window System
- The Display Postscript System
- Fonts
- The User Environment

4.1 Miscellaneous UWS Software Problems

This section describes various UWS problems.

4.1.1 Xws Does Not Maintain the Keyboard Shift Lock Light Correctly

The Shift Lock Light (LED) on LK201 and LK401 keyboards indicates the state of the shift lock key. The `Xws` server does not maintain the shift lock light correctly. Switching focus to another window when shift lock is on causes the light to be turned off even though shift lock is still on.

This is particularly a problem when entering a password or other text that is not echoed on the screen because there is no indication that you are entering shifted characters.

4.1.2 MIT Client xrdb

The MIT client xrdb does not permit modifications to specific screens in a multiscreen configuration. Instead, all changes are directed to screen zero.

There is no workaround to this problem.

4.1.3 Some dxpsview(1X) Problems

The `dxpsview`(1X) command does not run when the default visual class is set to `StaticGray` on a PXG-family graphic option.

There is no workaround to the problem: to avoid it, do not set the default visual class to `StaticGray`.

The `dxpsview`(1X) command produces a segmentation fault when the default visual class is set to `StaticGray` or to `GrayScale` on a PX graphics option.

There is no workaround to the problem: to avoid it, do not set the default visual class to either `StaticGray` or to `GrayScale`.

4.1.4 Session Manager and Arrow Keys

The session manager Customize Application Definitions window ignores the left and right arrow keys because the standard session manager key bindings ignore them.

You can work around the problem two ways:

- Using the mouse to position the insertion point
- Adding appropriate key mappings to your `.Xdefaults` file

4.1.5 Window Manager Visual Class Defaults Ignored

When different visual class types are used on different screens, the default visual class type of the `Mwm` window manager is ignored.

There is no workaround to this problem.

4.1.6 Visual Class Options for TX Graphics

The TX option supports two depths, 8 and 24 bits, and six possible visual classes in the following combinations:

Depth	Class
8	StaticGray
8	GrayScale
8	StaticColor
8	PseudoColor
8	TrueColor
8	DirectColor
24	TrueColor
24	DirectColor

By default, the server is started with a root window depth of 8 bits and a root visual class of PseudoColor. The "-class <classname>" command line option can be used to change the default root visual class and depth as follows:

<classname>	Depth	Class
StaticGray	8	StaticGray
GrayScale	8	GrayScale
StaticColor	8	StaticColor
PseudoColor	8	PseudoColor
TrueColor	24	TrueColor
DirectColor	24	DirectColor

Alternatively, there is another command line option that can also be used to change the default root visual class and depth:

```
-txRootVisual <depth> <classnumber>
```

Here <depth> can be 8 or 24 bits, and <classnumber> is specified as follows:

Visual class	<classnumber>
StaticGray	0
GrayScale	1
StaticColor	2
PseudoColor	3
TrueColor	4
DirectColor	5

4.1.7 Restarting the X Server or Recovering from a Hung Server

You do not have to reboot a system to recover from a hung server. The best way to recover is to kill the server process. When you do so, the `init` process detects the event and restarts the server for you.

To kill the server process, follow these steps:

1. Log in to the system through an alternate route, such as `rlogin`, `dlogin`, or `lat`.
2. Change your user ID to `root` using the `su` command.
3. Locate the server process by issuing the `ps -ax` command as root.

   There are five possible server names: Xws, Xtx, Xwst3d, Xqvsm, or Xqdsg.
4. Kill the server process by issuing the `kill -9` command, using the process ID you obtained from the `ps` command.

The server should be restarted within 30 seconds.

If a server does hang, you might want to check the `/usr/adm/X?msgs` file for any warnings and act on them before you restart the server.

Several Xwst3d server "hanging" problems are induced by using save-unders. You may eliminate hangs by disabling save-unders. See Section 4.2.1.4 for more information about save-unders.

4.1.8 Workstation Monitor Resolution

The ULTRIX graphics driver cannot determine the type of monitor connected to a workstation, so the driver assumes default values for the screen size and resolution based on the graphics option installed in the system. If these defaults are not correct for your monitor, specify the correct value for the resolution with the `-dpi` (dots-per-inch) option when starting the server. This will also set the correct screen size.

You can use the `xdpyinfo` utility from the Unsupported X11R4 Components subset to determine the server's current values for the screen size and resolution. For

example, if you run the /usr/bin/X11/xdpyinfo program, the following text might be displayed:

```
screen #0:
  dimensions:    1280x1024 pixels (280x230 millimeters)
  resolution:    116x113 dots per inch
```

To change these values, add the -dpi option to the server startup line in the /etc/ttys file. For example:

```
:0 "/usr/bin/login -P /usr/bin/Xprompter -C /usr/bin/dxsession -e"
none on secure window="/usr/bin/Xws bc -dpi 75"
```

Beginning with ULTRIX and UWS Version 4.2A, the default screen size and resolution for some graphics options was changed to be correct for the 16-inch monitors that are included with many of the newer workstations. If you have one of these larger monitors, you may now have to set the -dpi option even though this was not necessary with earlier releases.

4.1.9 Current Xwst3d Server Problems

Under certain circumstances the server gets into an infinite loop where it runs out of memory and exits. This problem occurs when the server calculates what appears to be a bad normal whose three vertices are very close together. This problem occurs in the fillarea tessellation part of the server. There is no workaround for the problem.

Nurb surfaces having multiple identical knot points do not show a discontinuity at the corresponding point on the surface. It appears that the normals are flipped (incorrectly) in one of the paths within the server nurb code. Normally, the problem will not be noticed if the front and back surfaces have the same interior style and cull mode. There is no workaround for the problem.

4.2 X Window System

The following sections discuss the ULTRIX Worksystem Software X servers, Xlib functions, and other X-related issues. For additional notes on X servers specific to individual processors, see Chapter 2.

4.2.1 ULTRIX Worksystem Software X Servers

The ULTRIX Worksystem Software, starting with Version 4.2A, contains the Xws X server. Xws is an X server compliant with X11 Release 4 specifications for the DECsystem 3100 and DECstation/DECsystem 5000 homogeneous multiscreen configurations for Models CX, MX, HX and PX.

4.2.1.1 Specifying a Port for Graphics Tablet Communications

On a DECstation 5000 Model 200 Series processor, you can use the third or fourth port for graphics tablet communications. Open the third port for graphics tablet communication with the Xws -tb 0 command and the fourth port with the Xws -tb 01 command. Do not specify the fourth port for the graphics tablet with the Xws -tb 4 command, because if you do, the server will hang. This will cause the Xws server to hang.

4.2.1.2 Server-Client Interaction and DECnet Addressing

Starting with ULTRIX Worksystem Software Version 4.0, DECnet addresses are specified with a variable length. The `dn_naddr` structure holds the address length in the first two bytes and the address itself in the remaining bytes.

In earlier releases of ULTRIX Worksystem Software, DECnet addresses were two bytes long. The `dn_naddr` structure was always four bytes long; the first two bytes contained the length of the address, and the second two bytes contained the address. X servers that expect DECnet addresses to be fixed at two bytes in length will not communicate properly with some UWS X clients under ULTRIX Worksystem Software Version 4.2A and higher systems.

Note that DECnet Phase V specifies that the address within `dn_naddr` is a variable-length address. DECnet code within the DECwindows Session Manager reserves space for DECnet Phase V addresses in the X protocol. This ensures compatibility with DECnet Phase V. To use the DECwindows Session Manager with a third-party (non-Digital) server requires that the server interpret `dn_naddr` as a variable-length structure.

For more information on `dn_naddr` and the X Protocol, see pages 463 and 734 in the *X Window System: The Complete Reference to Xlib, X Protocol, ICCCM, XLFD*, Second Edition, Scheifler, Robert W. and James Gettys.

4.2.1.3 Default Keyboard Keymap

Specifying a foreign keyboard layout using a ROM boot code does not affect the keyboard layout used in X. To change the keyboard layout defined by a keymap in X, you must first log in as superuser and then create a `keymap_default` symbolic link in the `/usr/lib/X11` directory, which points to the keyboard keymap you want to load. The US LK201-LA keyboard keymap is the default. The following example shows you how to set the default keyboard keymap to the Swedish LK201:

```
# cd /usr/lib/X11
# ln -s keymaps/swedish_lk201lm.decw_keymap keymap_default
```

You must restart the X server after changing the default keyboard keymap. Refer to Section 4.1.7 for information on restarting the server. In addition, you need to set DECterm to VT300 mode, with 8-bit controls, or VT300 mode with 7-bit controls and place the "stty pass8" command in your `.login` file to enable 8-bit ASCII character support. When the server is restarted, the requested keyboard mapping will be available. Note that you cannot use alternate keymaps while in single-user mode (single-user mode offers no 8-bit support).

To set the default keyboard keymap to US LK201-LA, you must remove the `/usr/lib/X11/keymap_default` file.

In a diskless environment, the `/usr/lib/X11` directory is mounted read-only. Therefore, multiple diskless clients with different keyboards (German, French, English, Spanish) cannot have private `keymap_default` entries in `/usr/lib/X11`. To work around this problem, you need to make X11 look to somewhere else in the directory structure (for example, `/var/X11/keymap_default`), which is in the read-write area and is thus workstation specific. Another link can then point to different keyboard mappings. To do this, follow these steps:

1. Create a symbolic link from `/usr/lib/X11/keymap_default` to

/var/X11/keymap_default:

```
# ln -s /var/X11/keymap_default /usr/lib/X11/keymap_default
```

2. Then on each client workstation, create a symbolic link from /var/X11/keymap_default to the keyboard keymap that you wish to use on that workstation:

   ```
   # ln -s /usr/lib/X11/keymaps/keyboard /var/X11/keymap_default
   ```

   In this example, *keyboard* represents the physical keyboard on that workstation.

4.2.1.4 Save-Unders and Backing Store

All X servers except the Xgb server now support save-unders as well as backing store functionality. Save-unders and backing store off-load window refreshing to the server at the cost of increased memory usage.

By default, save-unders and backing store are disabled on VAX systems and enabled on RISC systems. The state (enabled or disabled) of save-unders and backing store is determined by a command line option to the server executable. The command to invoke this executable is usually located in the /etc/ttys file. To turn save-unders and backing store on or off, you must edit the line in the /etc/ttys file where the server is invoked. The server must be restarted for any changes made in /etc/ttys to take effect.

To enable save-unders and backing store, invoke the server *without* the -su (save-under) and -bs (backing store) command line options. For example, to enable save-unders and backing store in the Xws server, invoke the server with the following command:

```
# Xws
```

Save-unders can be disabled with the -su command line option. For example, to disable save-unders in the Xws server, invoke the server with the following command:

```
# Xws -su
```

Save-unders can be disabled selectively by putting the following line in the .Xdefaults file:

```
*saveUnder: FALSE
```

This action prevents a window manager (such as mwm) from using save-unders, yet allows applications which really need save-unders to override the default behavior.

Backing store and save-unders should be disabled for 3D graphics applications that utilize the PEX extension included as part of the Xwst3d X server. See Section 5.6.1 for more information.

To disable both backing store and save-unders, invoke the server with the following command:

```
# Xws -su -bs
```

4.2.1.5 Server Memory Usage Characteristics

The X server grows in size depending on the services demanded of it by application programs. As an application requests memory, the server process size increases. However, when the memory is freed, the server process size stays the same rather than getting smaller. For example, if the PostScript Previewer is used to view a file and the scale factor is set to four, a 20 megabyte pixmap is created. This memory remains allocated to the server, even when the user exits the PostScript Previewer.

The size of the currently running server can be determined by the `ps -aux` utility. Application programs typically cause the server to grow on initial startup, but eventually, when the server has met the needs of the application, the server size as shown by `ps` should not grow anymore.

4.2.1.6 Conserving Server Memory

When creating several PostScript contexts that share a single Graphics Context (GC), you can conserve server memory by recycling GCs whenever possible. To do this, create a new GC, copy the old GC into the new one, and use the new GC with the PostScript context.

4.2.1.7 Restarting the Server Automatically

If the X server is started from the `/etc/ttys` file, the `-once` command line flag can be used to restart the X server automatically every time the Session Manager exits. For example, assume the X server is started with the following options in the `/etc/ttys` file:

```
Xws -once bc
```

In this case, the server will exit when the user exits `dxsession`. A new server process is then immediately started, and a prompter window is displayed.

If you use operations which cause the X server process size to grow large periodically, such as scaling documents for viewing with the PostScript Previewer, use this option to automatically exit and restart the server.

4.2.1.8 Problems Due to Swap Space Size

If the system's swap space disk partition fills, an attempt by the server to access a previously unused page in the dynamically allocated portion of its data segment crashes the server with a segmentation fault. This problem can occur at various points during server execution, not just at those related to the allocation of server resources.

Swap space is allocated when the memory is first accessed, not when the corresponding memory segment is allocated.

To avoid the problem, configure your system with a larger swap space. Swap space requirements of the applications being used dictate the minimum acceptable swap partition size.

4.2.1.9 Host Names in X Server Access Control List

When the X server starts up, it places the string of the "localhost" and the name of the host where the X server is running into the server's access control list. In addition, if there is a `/etc/X0.hosts` file or `/etc/X1.hosts` file, its contents

are added to the access control list.

If you have the `sm.host_list` resource in your `.Xdefaults` file, the access control list replaces the server's list when `dxsession` starts up. If no such resource exists, `dxsession` will not change the server's list.

When you display the Session Manager Customize Security dialog box, the server's current access control list is listed. If this list is changed and the current settings are saved, a `sm.host_list` resource is placed into the `.Xdefaults` file. The next time you start `dxsession`, the server's list will be replaced by this resource.

4.2.1.10 X Server Messages File

X server messages are logged in the `/usr/adm/X#msgs` file. The file contains the date and time the X server restarts, the X error messages, and miscellaneous information about X server crashes and server restart. The number sign (#) indicates the number of the display where messages are being logged. For workstations with single displays, the messages file name is `X0msgs`. For workstations with two displays and keyboards (such as the VAXstation II/GPX) there are two messages files, `X0msgs` and `X1msgs`, for the first and second are displays, respectively.

Messages reported to the `X#msgs` file include:

- "Use" messages, which indicate that the line in the `/etc/ttys` file for starting the server has a typographical error. Check the `/etc/ttys` file for spelling or syntax errors.
- "Resource missing" or "resource installed improperly" messages, which indicate that fonts are not installed properly, or that the `-fp` switch in the `/etc/ttys` line is not correct. Check to see that the fonts you are using are installed properly, that the `-fc` option for setting the default cursor font is correct, and that the `-fn` option for setting the default text font is correct.

  For example:

  ```
  main: Could not open default font 'XXX'

  main: Could not open default cursor font 'XXX'
  ```

- "Could not open RGB_DB 'XXX'" messages, which indicate that the rgb database is not installed correctly. These messages refer to files that apply to color workstations only. For VAXstation 3520/3540s, the database files are in `/usr/lib/rgb.*`. For all other color workstations, the database files are in `/usr/lib/X11/rgb.*`.
- "Screen failed initialization" messages, which usually indicate that the file `/dev/mouse` is not protected correctly or there is already a server running.
- Other error messages that indicate that the X server is out of memory.

Some of the X messages are fatal errors.

4.2.2 Xlib Functions

The following sections discuss Xlib functions and related issues.

For overview information and tutorials on programming with Xlib and the X Toolkit, see *X Window System Toolkit: The Complete Programmer's Guide and Specification*, X Version 11, Release 4, Asente, Paul J. and Ralph R. Swick, or *X Window System: The Complete Reference to Xlib, X Protocol, ICCCM, XLFP,* Second Edition,

Scheifler, Robert W. and James Gettys.

4.2.2.1 The Xlib Shared Memory Transport (SMT)

There are two problems with Xlib SMT:

- Users should not use SMT with applications that intentionally generate X protocol errors. These applications are usually window managers. For performance reasons, SMT is not recommended with window managers.
- Users should not use SMT with applications that employ the SHAPE extension,

4.2.2.2 XDrawLine Function

If you use the `XDrawLine` function to define a line with a width of zero and coincident endpoints, the line may or may not be displayed. To ensure that a line is drawn, either specify lines with a width of one or do not specify coincident endpoints.

4.2.2.3 LockDisplay and UnlockDisplay Macros

The X Window System programming manuals mention use of `LockDisplay` and `UnlockDisplay` macros for writing multithreaded X clients that access buffers of the Display structure. These macros are often used when writing Xlib side extensions to the core X11 protocol.

However, ULTRIX Worksystem Software Version 4.0 and higher systems do not support these side extensions. Hence, the `LockDisplay` and `UnlockDisplay` macros are not included. If you require these macros, order the ULTRIX Worksystem Software source kit. Note that you may be able to accomplish display structure locking using the public domain macros provided by MIT's Xlib.

Note that if asynchronous event handling is used, supported core Xlib calls are safe for use by multithreaded clients. Asynchronous event handling may, however, affect client performance.

4.2.2.4 Memory Allocation Routines

Xlib defines its own internal versions of the `malloc()`, `calloc()`, `free()`, and `realloc()` routines. These are also used by the XUI Toolkit. All clients that link with Xlib or the XUI Toolkit should use the Xlib-defined versions of these routines. The `malloc()`, `free()`, and `realloc()` routines are defined in `XvmsAlloc.o` in `libX11.a`. The `calloc()` routine is defined in `XvmsCalloc.o` in `libX11.a`.

If you prefer to use your own version of the `malloc()`, `free()`, or `realloc()` routines, then you must take action to avoid multiple declaration errors when linking. In particular, some popular public domain software packages define their own versions of some of these memory allocation functions. You can compile these packages under UWS Version 4.2A (or higher) by doing one of the following:

- Disable the package's version of the functions. Most packages have an option or flag you can set to request that the package use the system memory allocation functions.
- Add function definitions for the functions not defined in the package. For example, if the package defines `malloc()` and `free()` but not `realloc()`,

add a `realloc()` routine. You do not have to worry about `calloc()`, as `libX11.a` defines this in a separate file.

- Disable the functions in Xlib by removing the `XvmsAlloc.o` and `XvmsCalloc.o` modules from `libX11.a`.

4.2.2.5 ULTRIX System V Emulation Library

To use the ULTRIX System V emulation library you must link `/usr/lib/XlibIntV.o` before `/usr/lib/libX11.a`. For example, you would enter this command:

```
% cc xsample.c /usr/lib/XlibIntV.o -lX11 -lcV
```

4.2.2.6 XCopyArea Function

An application may hang and freeze up the system with an `XCopyArea` function call followed by an `XIfEvent` call. To avoid the possibility of deadlock, use an `XSync` function call before issuing the `XIfEvent` function call. The `XSync` call will flush the output buffer and wait for all requests to be received and processed by the server. In general, if `XIfEvent` is waiting for an expected event generated by an `X` function call, use `XSync` first to guarantee that all requests to the server have been sent. The application must guarantee that all events are truly written out before the wait is begun.

4.2.2.7 XDrawArc and XDrawArcs Functions

Line style modes of `LineOnOffDash` and `LineDoubleDash` do not work with the `XDrawArc` or `XDrawArcs` functions. The `XDrawArc` function is used to draw a single circle or ellipse. The `XDrawArcs` function is used to draw multiple circles and ellipses. These functions are used by some MIT applications.

4.2.2.8 Data Structures and Constants

In ULTRIX Worksystem Software Version 4.2, additions were made to the `XSizeHints` and `XStandardColormap` data structures and a new structure was added for manipulating properties containing text.

4.2.2.8.1 X Size Hints – The following routines are used for getting and setting size hints:

- The `XGetWMSizeHints` routine returns the size hints stored in the indicated property on the specified window. If the property is of type WM_SIZE_HINTS, of format 32, and is long enough to contain a size hints structure, the various fields of the `hints_return` structure are set and a nonzero status is returned. Otherwise, a status of zero is returned. To get a window's normal size hints, the `XGetWMNormalHints` routine can be used instead. For example:

```
Status XGetWMSizeHints (dpy, w, hints_return, property)
    Display *dpy;
    Window w;
    XSizeHints *hints_return;
    Atom property;
```

- The `XGetWMSizeHints` routine can generate BadWindow, BadAtom, or BadValue errors.

- The XGetWMSizeHints routine supersedes XGetSizeHints.
- The XSetWMSizeHints routine replaces (or sets if the property does not exist) the size hints for indicated property on the specified window. The property is stored with a type of WM_SIZE_HINTS and a format of 32. To set a window's normal size hints, the XSetWMNormalHints routine can be used instead. For example:

```
void XSetWMSizeHints (dpy, w, hints, property)
    Display *dpy;
    Window w;
    XSizeHints *hints;
    Atom property;
```

- The XSetWMSizeHints routine can generate BadWindow, BadAtom, or BadValue errors.
- The XSetWMSizeHints routine supersedes XSetSizeHints.
- The XGetWMNormalHints routine returns the size hints stored in the WM_NORMAL_HINTS property on the specified window. If the property is of type WM_SIZE_HINTS, of format 32, and is long enough to contain a size hints structure, the various fields of the hints_return structure are set and a nonzero status is returned. Otherwise, a status of 0 is returned. For example:

```
Status XGetWMNormalHints (dpy, w, hints_return)
    Display *dpy;
    Window w;
    XSizeHints *hints_return;
```

- The XGetWMNormalHints routine can generate BadWindow, BadAtom, or BadValue errors.
- The XGetWMNormalHints routine supersedes XGetNormalHints.
- The XSetWMNormalHints routine replaces (or sets if the property does not exist) the size hints for the WM_NORMAL_HINTS property on the specified window. The property is stored with a type of WM_SIZE_HINTS and a format of 32. For example:

```
void XSetWMNormalHints (dpy, w, hints)
    Display *dpy;
    Window w;
    XSizeHints *hints;
```

- The XSetWMNormalHints routine can generate BadWindow, BadAtom, or BadValue errors.

4.2.2.8.2 XStandardColormap – Two new elements have been added to properties of type RGB_COLOR_MAP:

- The ID of the visual from which the colormap was created
- An arbitrary resource ID that indicates whether or not the cells held by this standard colormap should be released by freeing the colormap ID or by doing a KillClient on the indicated resource (necessary for allocating out of an existing map)

These fields can be added to the end of the existing structure (defined in Xutil.h)

without disrupting the existing interfaces:

```
typedef struct {
        Colormap colormap;
        unsigned long red_max;
        unsigned long red_mult;
        unsigned long green_max;
        unsigned long green_mult;
        unsigned long blue_max;
        unsigned long blue_mult;
        unsigned long base_pixel;
        VisualID visualid;       /* NEW */
     XID killid;           /* NEW */
} XStandardColormap;
```

In addition, a new symbolic value has been added to `Xutil.h` indicating resources have been released by freeing the colormap:

```
#define ReleaseByFreeingColormap ((XID) 1L)
```

4.2.2.8.3 XTextProperty – Many of the properties of type STRING were changed to allow a variety of types and formats. Because the data stored in these properties is no longer simple null-terminated strings, a new data structure describes the encoding, type, length, and value of the text as well as its value. The following structure was added to `Xutil.h`:

```
typedef struct {
    unsigned char *value;     /* property data */
    Atom encoding;            /* type of property */
    int format;               /* 8, 16, or 32 */
    unsigned long nitems;     /* number of items in value */
} XTextProperty;
```

4.2.2.8.4 WithdrawnState Constant – Even though interfaces to the WM_STATE property are not being defined, the `WithdrawnState` constant was added to `Xutil.h` with a value of zero.

4.2.3 MIT X11 Release 3 Files

MIT X11 Release 3 files are no longer supported or shipped with ULTRIX Worksystem Software. MIT X11 Release 3 fonts that were previously in `/usr/lib/X11/fonts/MIT` have been removed from the kit, and MIT X11 Release 3 bitmaps in `/usr/lib/X11/bitmaps` have been removed from the kit. Header files and miscellaneous other files that were in `/usr/include/mit/X11` have also been removed from the kit.

4.2.4 MIT X11 Release 4 Server and Applications

The following sections discuss the development process for those customers who want to use MIT applications, libraries, and development tools.

Note

The MIT X11 Release 4 applications, libraries, and development tools, which are utilized in an Athena-based development environment, are provided by Digital on an as-is basis. This software is not warranted by Digital, and no Software Performance Reports will be acknowledged or answered on any component of the Athena software. Support service cannot be purchased for this software.

4.2.4.1 MIT Fonts

Third-party X applications may contain references to MIT X fonts supplied with the X Window System from the X Consortium. The naming scheme for the MIT fonts changed between Release 3 and Release 4 of the X Window System by the X Consortium. Although the MIT X fonts are not supported by Digital, Digital does supply the Release 4 fonts from the supported portion of the X Consortium X11 Release 4 tape. These fonts are installed from the ULXUNFONTS430(VAX) subset or the UDXUNFONTS430(RISC) subset.

If you are using an application that assumes the X11 Release 3 naming conventions for any of the fonts mentioned below, you should either install the sample `fonts.alias` file `fonts.alias.mit` discussed in Section 4.4.8.4, or change your X Resource Database specifications to reflect the new font naming. The fonts are as follows:

```
fixed variable 5x8 6x9 6x10 6x12 6x13 6x13bold 7x13 7x13bold
7x14 8x13 8x13bold 8x16 9x15 9x15bold 10x20 12x24 k14 a14 r14
rk14 r16 rk16 r24 rk24 kana14 8x16kana 8x16romankana 12x24kana
12x24romankana
```

If you want to use MIT fonts, you must convert them to PCF format using the `dxfc` font compiler. Place the fonts in a font directory and run the `dxmkfontdir` utility in that directory. Include the new font directory in the server's font path using the `fp` switch, and restart the server.

4.2.4.2 Logging On Without the DECwindows Session Manager

Some users choose to bypass the standard DECwindows login sequence and either use a different login prompter, such as `xdm`, or to run `/bin/login` in an `xterm` window to allow users to log in. In such a configuration, the DECwindows Session Manager is generally not run.

By default, all console output on ULTRIX workstations is directed to the dumb terminal display driver. This driver, which provides console output at boot time and when in single user mode, displays text on the screen without regard to the presence of some other process that may be already controlling the display, such as the X server. When the X server is running and a message is written to `/dev/console`, the message will overwrite some portion of the X display.

The DECwindows Session Manager, `dxsession`, uses a device called `/dev/xcons` to redirect console output to itself. While it has `/dev/xcons` open for reading, any output to `/dev/console` is made available for reading on `/dev/xcons`, and no output is written to the workstation's screen.

Users who have overridden the standard DECwindows login procedure may find that portions of their X display are occasionally overwritten by console messages. To

prevent this from happening, run the following command:

```
/usr/bin/X11/xterm -e cat /dev/xcons &
```

For convenience, this command can be placed in a user's `.login` or `.X11Startup`, or, for `xdm`, added to the `xdm` startup script. Because `cat` does blocking reads on its input files, it will simply read any console messages from `/dev/xcons` as they appear and display them in the xterm window. The `/dev/xcons` is documented in the reference pages for most Digital display devices, such as `pm`(4), `cfb`(4), `qd`(4), `qv`(4), `sg`(4), `sm`(4), or `fg`(4).

Note

Using a non-DECwindows login procedure is not supported by Digital.

4.2.4.3 Unsupported MIT X11 Release 4 Clients

The unsupported subset of this release is essentially a build of the MIT X11R4 clients with all of the public patches available at the time the kit was put together. Very few local changes have been made to this code; only changes that were needed to make the clients function under the latest version of ULTRIX were made.

It is assumed that you know what the MIT clients do and why you want to use them; if not, and if the reference pages for the unsupported MIT software do not provide sufficient guidance, a wide variety of documentation on the X Window System from MIT's Project Athena is available at many bookstores.

4.2.4.3.1 Changes to MIT X11 Release 4 Clients – The user-visible changes that have been made from the MIT public distribution, patchlevel 18, are as follows:

- The `xterm` utility has an added resource, `.decCShellFix`, and an added option, `-deccshfix`. The use of either of these causes `xterm` to interpret character code 136 as backspace. This is a workaround for a problem in the `csh` shell, where the ^[is not erased during command completion. To enable the workaround, set the resource or command line option to `on` ; to disable the workaround, set them to `off`.
- The `xmh` command has had two changes. First, `xmh` uses `enscript` as a printer filter by default. Because the `enscript` command is not shipped with ULTRIX, the print option would silently fail. Therefore, the default print filter was changed to `lpr`.

  Secondly, `xmh` passes a `-file` option to the mh Mail Handler `inc` command. If you are reading mail using a version of `inc` that supports the Post Office Protocol (POP), the `-file` option overrides the `-host` option that might be specified in the `.mh_profile` file. Therefore, a new resource, `.popHost`, has been added. This can be used to specify a POP server to use, for example:

  ```
  Xmh.popHost: popserver.foo.com
  ```

A few other changes were required to make `xterm` compile with POSIX header files, but these changes are not visible to the user and do not result in any difference in functionality.

4.2.4.3.2 Changes for Athena-based Development – In order to provide a completely Athena-based development environment for those who might want it, the Athena versions of `libX11.a`, `libXt.a`, and `libXmu.a` have been provided as `libX11-mit.a`, `libXt-mit.a`, and `libXmu-mit.a`, and the Athena versions of the include files have been provided in `/usr/include/mit/X11/Xaw`. Other Athena libraries have been included with their standard names, since they do not conflict with DECwindows libraries.

In order to build completely with MIT versions of header files and libraries, you must add `-I/usr/include/mit` to any `cc` compile command, and also change any references to `/usr/lib/libX11.a`, `/usr/lib/libXt.a`, and `/usr/lib/libXmu.a` to `/usr/lib/libX11-mit.a`, `/usr/lib/libXt-mit.a`, and `/usr/lib/libXmu-mit.a`. Note that the version of `libX11-mit.a`, `libXt-mit.a`, and so on, included in this unsupported subset do not work with the DECwindows toolkit or the Motif toolkit. Both of these toolkits include their own versions of `libXt` and expect to link with the UWS supported version of `libX11.a`.

4.3 Display PostScript System

The Display PostScript System extends the entire PostScript language to bitmap displays. ULTRIX Worksystem Software implements the Display PostScript System as an extension to the worksystem server. Using this extension, DECwindows client applications can send both PostScript imaging requests and X requests to the same display using a single network connection.

The following sections discuss the Display PostScript System.

4.3.1 Correct Phone Number for Adobe Training Support

The phone number for Adobe Training Support is incorrect in the Adobe documentation. The correct number is (415) 962-4912.

4.3.2 New Release of Display PostScript

The ULTRIX Worksystem Software Version 4.2A (and higher) X11 Release 4 compliant servers support Display PostScript System (DPS) Release 1006.9.

4.3.3 Compatibility of Display PostScript Release 1006.5 With Release 1006.9

In Display PostScript release 1006.5, the standard 29 DPS fonts were preloaded into server memory at server startup time and were present in SharedFontDirectory. In DPS release 1006.9, in order to reduce memory usage, fonts are no longer preloaded. This means that at server startup, SharedFontDirectory does not contain any fonts.

As a result, any PostScript programs written for DPS release 1006.5 that try to list fonts by listing the contents of SharedFontDirectory will now list nothing, or will list only the fonts loaded by programs previously run.

Therefore, to provide compatibility with DPS 1006.5, Digital provides the following PostScript program that loads all 29 fonts. You must run this program before any

code that lists SharedFontDirectory.

```
currentshared true setshared
/AvantGarde-Book 1 selectfont
/AvantGarde-BookOblique 1 selectfont
/AvantGarde-Demi 1 selectfont
/AvantGarde-DemiOblique 1 selectfont

/Courier 1 selectfont
/Courier-Oblique 1 selectfont
/Courier-Bold 1 selectfont
/Courier-BoldOblique 1 selectfont

/Helvetica 1 selectfont
/Helvetica-Bold 1 selectfont
/Helvetica-BoldOblique 1 selectfont
/Helvetica-Oblique 1 selectfont

/LubalinGraph-Book 1 selectfont
/LubalinGraph-BookOblique 1 selectfont
/LubalinGraph-Demi 1 selectfont
/LubalinGraph-DemiOblique 1 selectfont

/NewCenturySchlbk-Bold 1 selectfont
/NewCenturySchlbk-BoldItalic 1 selectfont
/NewCenturySchlbk-Italic 1 selectfont
/NewCenturySchlbk-Roman 1 selectfont

/Souvenir-Demi 1 selectfont
/Souvenir-DemiItalic 1 selectfont
/Souvenir-Light 1 selectfont
/Souvenir-LightItalic 1 selectfont

/Symbol 1 selectfont

/Times-Bold 1 selectfont
/Times-BoldItalic 1 selectfont
/Times-Italic 1 selectfont
/Times-Roman 1 selectfont
setshared
```

4.3.4 Allocating a Colormap for Use with Display PostScript

The colormap cells allocated in a color cube or gray-scale ramp must be contiguous. The `XAllocColorCells` routine can be used to allocate contiguous planes, but not contiguous color cells. Using noncontiguous color cells produces undefined results.

The following routine allocates *n* contiguous colormap cells, assuming they exist in the specified colormap. Use this routine, rather than `XAllocColorCells`, when defining a colormap or gray-scale ramp for use with Display PostScript code:

```
static Bool
_AllocContiguousCells(dpy, cmap, pixels, npixels)
Display *dpy;
Colormap cmap;
unsigned long *pixels;          /* filled in by routine */
int npixels;
{
unsigned long *waste = (unsigned long *)NULL;
  int nwaste = 0;
  Bool contig = False;
  int status, i;
```

```
    unsigned long masks = NULL;

    while (!contig) {
        status = XAllocColorCells(dpy, cmap, False, &masks, 0, pixels,
                 npixels);
        if (!status)
            break;   /* can't get enough contiguous cells */
        for (i=0; i < (npixels-1); i++) {
            if (pixels[i] + 1 != pixels[i+1]) {
                /* isn't contiguous, keep trying */
                XFreeColors(dpy, cmap, &pixels[i+1], npixels - (i+1), 0);
                if (!waste)
                    waste = (unsigned long *)malloc((i + 1) *
                                sizeof(unsigned long));
                else
                    waste = (unsigned long *)realloc(waste, (nwaste +
                                (i + 1)) * sizeof(unsigned long));
                bcopy(pixels, waste+nwaste, (i+1) * sizeof(unsigned long));
                nwaste += (i+1);
                break;
            }
        }
        if (i == (npixels-1))
            contig = True;
    }
    /* clean up and return 'contig' */
    if (nwaste) {
        XFreeColors(dpy, cmap, waste, nwaste, 0);
        free(waste);
    }
    return (contig);
}
```

4.3.5 setrgbXactual Operator Name Changed to setXrgbactual

The `setrgbXactual` operator has been renamed to `setXrgbactual`, for consistency in naming with the other `X` specific operators. The name `setrgbXactual` remains as an alias.

4.3.6 Contexts Created Using the Default Colormaps

Display PostScript programs that use `XDPSCreateSimpleContext()` to create display PostScript contexts with the default colormaps have slightly inferior color rendition. Colors are chosen from a palette of 64 colors (plus eight grays) instead of 125 colors. Applications that need to use more than 64 colors can get them by using the `XDPSCreateContext()` context, which allows the use of an unlimited number of colormap cells.

4.3.7 Changing the Default XStandardColormap

If you want to alter the `XStandardColormap` default used by simple Display PostScript clients, or if you want to free the colormap cells used by the `XStandardColormap` routine, you can do so by following the methods described in the ICCCM.

Note

Before you attempt to change the `XStandardColormap` default, you must be certain that no clients are using `XStandardColormap`.

4.3.8 Automatic PostScript Garbage Collection

Automatic PostScript garbage collection is turned on by default. It is a global setting to the server, but it is turned on every time `start` executes, that is, each time a client creates a context.

Any client can turn off automatic garbage collection, but doing so turns it off for all clients. Garbage collection is turned back on the next time a client creates a context.

There is no way to determine the current state of automatic garbage collection.

4.3.9 Example Programs Using the Display PostScript System

The `/usr/examples/dps` directory contains example programs that use the Display PostScript system. All worksystem examples are provided on an as-is support basis. For more information about these example programs, see the *Guide to Developing Applications for the Display PostScript System* and the Software Product Description.

4.3.10 Additional Documentation

In addition to the documentation described in the *Guide to Developing Applications for the Display PostScript System,* developers can refer to Adobe's Document Structuring Conventions specification, available free of charge from Adobe Systems, Inc..

To request a copy by electronic mail, send a mail message to the Adobe file server at either of the following network addresses:

Internet	`ps-file-server@adobe.com`
UUCP	`...!decwrl!adobe!ps-file-server`

4.4 Fonts

The following sections discuss how fonts are installed, named, and organized in directories. It also provides notes for application programmers on using fonts. For more information on fonts, see the *X Window System: The Complete Reference to Xlib, X Protocol, ICCCM, XLFD*, Second Edition, Scheifler, Robert W. and James Gettys.

Fonts are installed by the ULTRIX Worksystem Software (or by individual applications) and are read by the X servers on behalf of applications. An X Consortium standard defines the bitmap distribution format (BDF) in which font sources are distributed. However, X servers and applications use fonts in a compiled format, which is not standardized.

Most of the ULTRIX Worksystem Software Version 4.2 and 4.2A servers use fonts in the X11 portable compiled font (PCF) format. This represents a change from the ULTRIX Worksystem Software Version 2.1 release, where compiled fonts were in DECwindows format (DWF) or server natural format (SNF). ULTRIX Worksystem Software Version 4.0 made changes to the directories in which fonts are located and

expanded the directory structure for user-supplied fonts and application-supplied fonts.

ULTRIX Worksystem Software Version 4.0 and higher also adds fonts for use with the X implementation of Display PostScript (XDPS). These fonts have their own format, which is a PostScript-compatible ASCII format. If you have additional PostScript-compatible fonts, you can install them for use with Display PostScript.

If you are a system manager or applications programmer, Digital recommends that you familiarize yourself with how the font format change and new directories might affect your environment.

4.4.1 Fonts and Font Utilities

The ULTRIX Worksystem Software includes font sets for 75 dots-per-inch (dpi) and 100 dpi displays.

To use fonts other than those supplied in this kit, you must compile their `.bdf` font source files. Use the `/usr/bin/dxfc` font compiler to create `.pcf` files. Use `/usr/bin/dxmkfontdir` to create a list of the fonts in the directory for use by the X server. For more information, refer to `dxfc`(1X) and `dxmkfontdir`(1X) in the reference pages.

4.4.2 Default Font Directories

To be usable, a font must be installed in a directory on the X server's font search path. The `Xws` server uses 75 dpi fonts as the default. The `Xtx` and `Xwst3d` servers use the 100 dpi fonts as the default.

The server's font search path does not include font directories. You must alter the font path to select a font directory to use if it is not the default font.

4.4.2.1 75 dpi Fonts

If you install just the mandatory subsets, you install compiled 75 dpi fonts (the default fonts). The server directory search path for these fonts is as follows:

```
/usr/lib/X11/fonts/decwin/75dpi/
/usr/lib/X11/fonts/local/75dpi/
/usr/lib/X11/fonts/apps/75dpi/
/usr/lib/X11/fonts/private/75dpi/
```

When a font is requested (by the Xlib routine `XQueryFont` for example), the DECwindows X server looks for the font first in `/usr/lib/X11/fonts/decwin/75dpi/`, then in `/usr/lib/X11/fonts/local/75dpi/`, and so on.

The 75 dpi fonts for the `Xgb` server are in DWF format; they are installed in the following directories:

```
/usr/lib/dwf/75dpi
/usr/lib/dwf/compX10
/usr/lib/dwf/mitX11
```

4.4.2.2 100 dpi Fonts

The installation procedure allows you to install 100 dpi fonts from an optional subset. The server search path for 100 dpi fonts is as follows:

```
/usr/lib/X11/fonts/decwin/100dpi/
/usr/lib/X11/fonts/local/100dpi/
/usr/lib/X11/fonts/apps/100dpi/
/usr/lib/X11/fonts/private/100dpi/
```

The 100 dpi fonts for the `Xgb` server are installed in the following directories:

```
/usr/lib/dwf/100dpi
/usr/lib/dwf/compX10
/usr/lib/dwf/mitX11
```

To use the 100 dpi fonts, you must put them on the X server's search path by editing the workstation's `/etc/ttys` file. The server startup line looks like the following:

```
:0 "/usr/bin/login -P /usr/bin/Xprompter -C /usr/bin/dxsession" none on
secure window="/usr/bin/Xws"
```

Modify it to include the `-fd` and `-dpi100` switches as follows:

```
:0 "/usr/bin/login -P /usr/bin/Xprompter -C /usr/bin/dxsession" none on
secure window="/usr/bin/Xws -fd 100 -dpi 100"
```

The ULTRIX Worksystem Software font search path uses only the option specified in the `/etc/ttys` file. After editing the `/etc/ttys` file, restart the X server to change the default font search path.

4.4.2.3 Font Directory Contents

This section describes the contents of the following font directories:

- `/usr/lib/X11/fonts/decwin/75dpi/`
  `/usr/lib/X11/fonts/decwin/100dpi/`

  ULTRIX Worksystem Software uses these font directories for its supported fonts. Reserve these directories for use by the UWS software (consider them as being for read only use). These directories are shared in the diskless environment and might be moved or deleted in future releases of UWS.

- `/usr/lib/X11/fonts/local/75dpi/`
  `/usr/lib/X11/fonts/local/100dpi/`

  These font directories are a local version of `.../fonts/decwin`. You can use these directories to install site-specific fonts, such as the corporate logo. The directories are unsupported, and no maintenance is performed during the installation. However, they are in the default server font path, so fonts placed here are automatically available. These directories are not intended for use by applications. They are shared in the diskless environment.

- `/usr/lib/X11/fonts/apps/75dpi/`
  `/usr/lib/X11/fonts/apps/100dpi/`

  Shared applications can install fonts in these directories. Like the app-default file or class names, there is no registry; applications must use unique names to avoid collisions. The directories are unsupported, and no maintenance is performed during installation. However, they are in the default server font path, so fonts placed here are automatically available. They are shared in the diskless environment.

- /usr/lib/X11/fonts/private/75dpi/
  /usr/lib/X11/fonts/private/100dpi/

  Applications that install fonts that are licensed on a per-workstation basis can install fonts in these directories. Like app-default files or class names, there is no registry; applications must use unique names to avoid collisions. These directories are unsupported, and no maintenance is performed during installation. These directories are in the default server font path. They are not shared in the diskless environment.

- /usr/lib/DPS/outline/
  /usr/lib/DPS/outline.9/

  These directories contain subdirectories for Display PostScript (DPS) fonts. DPS fonts are found in the /usr/lib/DPS/outline.9/decwin directory if you are using the Xws server. If you are using the Xtm or the Xtm2d servers (for DECstation 5000 Model 200PX, PXG, or PXG Turbo workstations) or the Xgb server, the DPS fonts are found in the /usr/lib/DPS/outline/decwin directory.

  The DPS font directories differ from the previously-described server font directories. The DPS fonts are not .pcf files, and they are not on the server search path; they are used by DPS only. The decwin subdirectory holds the default Display PostScript fonts installed by ULTRIX Worksystem Software. The local subdirectory is where you can install site-specific fonts for DPS; this directory is shared in the diskless environment. The apps subdirectory is for applications that install fonts for sharing in the diskless environment. The private directory is where applications can install fonts that are not shared.

4.4.2.4 Installation Subsets and Server Font Directories

The following tables list the directories into which installation subsets install supported server fonts, unsupported server fonts, and Display PostScript fonts, respectively.

Subset	Location(s) for Supported Server Fonts
UDTAFM430	/usr/lib/font/metrics
ULTAFM430	/usr/lib/font/metrics
UDWFONT430	/usr/lib/X11/fonts/decwin/75dpi
UWSFONT430	/usr/lib/X11/fonts/decwin/75dpi
UDWFONT15430	/usr/lib/X11/fonts/decwin/100dpi
UWSFONT15430	/usr/lib/X11/fonts/decwin/100dpi
UWS3DFONT430	/usr/lib/dwf/75dpi, /usr/lib/dwf/100dpi
UDWFONTSTR430	/usr/lib/X11/strokefonts/pex (3D only)

Subset	Location(s) for Unsupported Server Fonts
UWS3DFONT430	/usr/lib/dwf/mitX11
UDXUNFONTS430	/usr/lib/X11/fonts/MIT
ULXUNFONTS430	/usr/lib/X11/fonts/MIT

Subset	Location(s) for Display PostScript Fonts
UDWSER430	/usr/lib/DPS/outline(or outline.9)/decwin
UWSSER430	/usr/lib/DPS/outline(or outline.9)/decwin

4.4.3 New Large Fonts for 100 dpi Monitors

By default, DECterm uses 14 point fonts for 100 dpi monitors when the Big Font button is selected in the DECterm Customize Window dialog box. At this release, you can specify 18 point fonts for 100 dpi monitors that are 19 inch or larger. To use these fonts, add the following bigFontSetName resource definition to your `default.DECterm` file:

```
DXterm.main.terminal.bigFontSetName: \
-bitstream-terminal-*-r-*--*-180-100-100-C-*-*-*
```

4.4.4 Application-Specific and Custom Fonts

The X servers can read fonts in PCF format, compressed PCF format, and BDF format (except for the `Xgb` server, which reads only DWF format fonts, as in a previous UWS release). It is best to compile fonts into PCF format. Compressed PCF files cause slightly slower performance; the uncompiled BDF files substantially degrade the server's performance.

The servers do not read the following:

- Fonts produced by the MIT X11 Release 3 compiler (whose file suffix is `.snf`)
- Fonts produced by the DECstation/DECsystem 3100 ULTRIX Worksystem Software Version 2.1 (RISC) font compiler (whose file suffix is `.snf`)
- Fonts produced by the VAX ULTRIX Worksystem Software Version 2.1 font compiler (whose file suffix is `.dwf`)

To compile and install BDF fonts, perform the following steps:

1. Compile the fonts using the `dxfc` command. Put the output in a directory that is on the default font path, such as `/usr/lib/X11/fonts/local/75dpi/`.
2. Use the `dxmkfontdir` utility to create a list of the fonts in the directory. Put the list into a file named `fonts.dir`.

You can now access the fonts when you log in. To compress fonts into `.pcf.Z` files, use the `compress` utility.

To ensure that your fonts are always available in the X server's font search path, place fonts in the default font directories. If you choose to install your fonts elsewhere, alter the search path to make the fonts available by editing the

/etc/ttys file for each workstation.

In the /etc/ttys file, add the new directory to the server startup line. Separate multiple font directories with commas. Use the +fp command line option and the following format to prepend elements to the font path:

```
+fp path[,path...]
```

Use the fp+ option and the following format to append elements to the font path:

```
fp+ path[,path...]
```

For example:

```
:0 "/usr/bin/login -P /usr/bin/Xprompter -C /usr/bin/dxsession" none on
secure window="/usr/bin/Xws fp+ /udir/susan/toyfonts/"
```

Note that if the /etc/ttys file is ever deleted, as it could be during a system software update, the X server will be unable to find these fonts until you again edit the file. For this reason, using the Digital-supplied font directories is the preferred practice.

4.4.5 Display PostScript Fonts

With the Version 4.2 and higher ULTRIX Worksystem Software, you obtain all of the fonts you need to use Display PostScript (DPS). These fonts are in the /usr/lib/DPS/outline.9/decwin directory if you are using the Xws server.

If you want to install additional PostScript-compatible ASCII fonts to use with Display PostScript, place them in the directories listed in Section 4.4.2.3. Restart the X server after installing the fonts.

4.4.6 German Standard Fonts

German Standard (TUV) fonts are monospaced screen fonts that meet the German standards for character shape and spacing and that are readable with an optical character scanner. To meet the TUV physical size specifications, two uncondensed fonts are provided: a small font and a large font.

4.4.6.1 Specifying TUV Fonts for all DECterm Windows

To specify TUV fonts as the default fonts for all DECterm windows, modify your .Xdefaults file by commenting out the bigFontSetName and littleFontSetName resource definitions, if present, and adding the following resource definitions:

```
DXterm*littleFontSetName: \
-dec-terminal-medium-r-normal-gs-*-*-*-*-c-*-iso8859-1
DXterm*bigFontSetName: \
-bitstream-terminal-medium-r-normal-gs-*-*-*-*-c-*-iso8859-1
DXterm*condensedFont:        off
```

Reload the .Xdefaults file to enable these changes. When DECterm is started, the font corresponding to littleFontSetName is automatically selected. To change to the large font in a DECterm window, open the Window Customize dialog box and click on the Big Font radio button. You can also add the following resource definition to

the `.Xdefaults` file to define the large font as the default font:

```
DXterm*fontSetSelection: 0
```

4.4.6.2 Specifying TUV Fonts for a Single DECterm Window

To specify TUV fonts when running DECterm from a shell, key in the `dxterm` command as follows:

```
dxterm -xrm 'DXterm*littleFontSetName: \
-dec-terminal-medium-r-normal-gs-*-*-*-*-c-*-iso8859-1' \
-xrm 'DXterm*bigFontSetName: \
-bitstream-terminal-medium-r-normal-gs-*-*-*-*-c-*-iso8859-1' \
-xrm 'DXterm*condensedFont:off'
```

To set the font to the large font, add `-customization 'DXterm*fontSetSelection:0'` to the `dxterm` command. For example, to invoke DECterm with the font corresponding to bigFontSetName, type the following:

```
dxterm -xrm 'DXterm*littleFontSetName: \
-dec-terminal-medium-r-normal-gs-*-*-*-*-c-*-iso8859-1' \
-xrm 'DXterm*bigFontSetName: \
-bitstream-terminal-medium-r-normal-gs-*-*-*-*-c-*-iso8859-1' \
-xrm 'DXterm*condensedFont:off' \
-xrm 'DXterm*fontSetSelection:0'
```

The condensedFont resource flag is either off or on. Off is the default. As TUV fonts have no condensed representation, the condensedFont resource flag must be off for these fonts to work.

4.4.7 Application Font Information for Developers

There is no need to recompile fonts when upgrading applications from ULTRIX Worksystem Software Version 4.1 to ULTRIX Worksystem Software Version 4.2 or 4.2A or 4.3. Applications should ship fonts in BDF format and compile them as part of the installation. This is necessary because the format of compiled fonts is variable. Font formats have changed from previous versions of the ULTRIX Worksystem Software; therefore, compiling fonts as part of the installation procedure protects your application from future format changes.

4.4.7.1 Font Properties

The font properties generally include the font name fields and other useful global font information, such as the height of capitals (CAP_HEIGHT), setwidth (SETWIDTH), and so on. All ULTRIX Worksystem Software fonts (except Terminal) have at least the following properties:

```
FONT_ASCENT          FONT_DESCENT
DEFAULT_CHAR         X_HEIGHT
FACE_NAME            COPYRIGHT
FOUNDRY              WEIGHT_NAME
SLANT                SETWIDTH_NAME
ADD_STYLE_NAME       PIXEL_SIZE
RESOLUTION_X         RESOLUTION_Y
AVERAGE_WIDTH        CHARSET_REGISTRY
CAP_HEIGHT           NOTICE
FAMILY_NAME          POINT_SIZE
CHARSET_ENCODING
```

Fonts derived from PostScript printer fonts also have the _DEC_DEVICE_NAMES property. This property sets up the correspondence between printer fonts requested by an X Display PostScript (XDPS) client (such as a Courier font requested in a PostScript file) and the screen fonts available to the server. During a font lookup, XDPS chooses a font whose property is "PS=Courier" when a client file requests Courier font. Some examples are as follows:

```
_DEC_DEVICE_NAMES "PS=AvantGarde-Demi"
_DEC_DEVICE_NAMES "PS=Century-Schoolbook-Bold-Italic"
```

Any application (and not just Display PostScript applications) can use this property to ensure that a screen font corresponds to a specified PostScript printer font. To do so, the application searches for a screen font with the _DEC_DEVICE_NAMES property specifying that printer font.

4.4.7.2 Changes to the Terminal Font

Since ULTRIX Worksystem Software Version 2.1, line-drawing characters (glyphs) no longer reside within the terminal font. They can be found within the terminal_dectech font.

The following fonts contain the line drawing glyphs:

```
/usr/lib/X11/fonts/decwin/75dpi/terminal_dectech18.pcf
/usr/lib/X11/fonts/decwin/75dpi/terminal_bold_dectech18.pcf
```

4.4.8 Font Names and Aliases

Font names consist of a series of parameter values separated by dashes, describing the typographic characteristics of the font. However, by using aliases, you can use a set of less cumbersome names.

4.4.8.1 Font Names

ULTRIX Worksystem Software uses the naming convention specified by the standard *X Logical Font Description Conventions, X Window System, Version 11.* Worksystem font names are specified using the logical font descriptions for the X protocol (XLFDs). A sample font name is as follows:

```
-Adobe-ITC Avant Garde Gothic-Book-R-Normal--14-100-100-100-P-80-ISO8859-1
```

In left-to-right order, these are the fields in a font name and the values in the sample name:

Field	Sample Value
Foundry	Adobe
Family_Name	ITC Avant Garde Gothic
Weight_Name	Book
Slant (Roman (R), Italic (I), Oblique (O), Reverse Italic (RI), Reverse Oblique (RO) or Other (OT))	R
Setwidth_Name	Normal
Pixel_Size	14
Point_Size, in decipoints	100

Field	Sample Value
Resolution, horizontal and vertical, in pixels/dots per inch	100 dots per inch (horizontal)
	100 dots per inch (vertical)
Spacing (Proportional (P), Monospaced (M), CharCell (C))	P
Average_Width, in decipoints	80
Charset_Registry	ISO8859
Charset_Encoding	1

A comparable font for a 75 dpi screen has the following name:

```
-Adobe-ITC Avant Garde Gothic-Book-R-Normal--10-100-75-75-P-59-ISO8859-1
```

This font needs 10 pixels to appear as 10 points. This font differs from the previous sample font only in pixel size, resolution, and character width.

You can use wildcards in font names to specify the values of certain font characteristics and allow the server to provide the appropriate values for those fields that differ on different workstation screens. A question mark (?) wildcard substitutes for a single character, and an asterisk (*) wildcard substitutes for one or more fields. The following font name specifies a 10-point ITC Avant Garde Gothic font of book weight, roman style, and normal spacing for display on either 75 or 100 dpi systems:

```
-Adobe-ITC Avant Garde Gothic-Book-R-Normal--*-100-*-*-P-*
```

When you use the asterisk, be sure that the substitutions resolve clearly. For example, in the following font name, the leftmost asterisk substitutes for two fields before the 100:

```
-Adobe-ITC Avant Garde Gothic-Book-R-Normal--*-100-*-P-*
```

The name resolves to two fonts:

```
-Adobe-ITC Avant Garde Gothic-Book-R-Normal--11-80-100-100-P-80-ISO8859-1
-Adobe-ITC Avant Garde Gothic-Book-R-Normal--14-100-100-100-P-80-ISO8859-1
```

The first is an 8-point font; the second is a 10-point font. The server chooses one of the fonts. ULTRIX Worksystem Software Version 4.2 and higher servers choose the first font (in ASCII-sorted order). Have your applications avoid dependence on this algorithm, as it could change in a future release.

4.4.8.2 Specifying Fonts

The X server supports monitors having different screen resolutions. Applications that care about the actual physical sizes of displayed fonts should use the height and width information associated with a screen to select an appropriate font.

For example, a WYSIWYG editor could do the following:

1. Find the resolution of the screen, by using the DisplayHeight(), DisplayHeightMM(), DisplayWidth(), and DisplayWidthMM() Xlib macros. For

example:

```
y-resolution = DisplayHeight() / DisplayHeightMM()
```

2. Construct a font name specification by specifying the FAMILY_NAME, WEIGHT_NAME, and POINT_SIZE fields, but use the asterisk (*) wildcard character for the RESOLUTION_X, RESOLUTION_Y, and PIXEL_SIZE fields. For example:

```
-adobe-helvetica-medium-r-normal-*-*-120-*-*-P-*-*-*"
```

3. Use XListFonts() to query the server for a list of matching fonts that conform to the specification in step 2.
4. Choose a font in the list that matches closest to the screen resolution determined in step 1 and use this font in an XLoadFont() call. If no listed font is acceptable, inform the user.

4.4.8.3 Font Name Aliases

ULTRIX Worksystem Software font files have lowercase names that indicate the contents of the file, and they have a `.pcf` suffix. Applications and defaults files do not reference fonts by font file name; they use font names. Each directory for ULTRIX Worksystem Software fonts has an alias file, called `fonts.alias` (FILE_NAMES_ALIASES) to define the name of each `.pcf` file in the directory as an alias for the font it contains. For example, the following font name:

```
DEC-Adobe-Helvetica-Bold-R-Normal--12-120-75-75-P-70-DEC-ISOLATIN1
```

is stored in the following file:

```
/usr/lib/X11/fonts/decwin/75dpi/helvetica12.pcf
```

The font can be referred to in Xlib routines or defaults files as either of the following:

```
DEC-Adobe-Helvetica-Bold-R-Normal--12-120-75-75-P-70-DEC-ISOLATIN1
```

or

```
helvetica12
```

You can set up additional aliases in the `fonts.alias` file. For more information, refer to the `dxmkfontdir(1X)` reference page and the example file in the next section.

4.4.8.4 Sample Font Alias Files

Problems with fonts may occur if you use non-Digital servers with Digital applications or Digital servers with non-Digital applications. Digital has provided two `fonts.alias` files to solve these problems. One file `(fonts.alias)` enables you to use DECwindows applications with non-Digital servers. The other file `(fonts.alias.mit)` enables you to use non-Digital applications with Digital servers. Both are installed from the ULXUNFONTS430(VAX) or the UDXUNFONTS430(RISC) subset into the directory `/usr/lib/X11/fonts/MIT`.

If you use non-Digital servers with Digital applications, problems with missing fonts may occur, since Digital applications may depend on fonts supplied on the DECwindows server kit, which a non-Digital server does not have access to. This

problem can be solved by using the `fonts.alias` file to direct the server to use alternate fonts.

The `fonts.alias` file maps fonts used by Digital's applications to fonts supplied on the MIT X11 Release 4 tape. Therefore, the following font families must already be installed on the server's node for this workaround to be successful:

- Courier
- Helvetica
- New Century Schoolbook
- Symbol
- Terminal
- Times

To use the `fonts.alias` file, install it under the name `fonts.alias` in your X server's font path. If you already have a `fonts.alias` file, you can safely concatenate your file with this file. Be sure that the `fonts.alias` file resides on the workstation running the X server rather than on the machine running the DECwindows application. To force the server to read the new `fonts.alias` file when the `font.alias` is in the server path, enter the following command:

```
% xset fp rehash
```

The `xset` command is provided in the U[DL]XUNMIT430 subset; be sure this is installed before attempting to use `xset`.

If you do not know your X server's font path, or if you do not have write access to it, create a directory on the machine that is running the X server, install the `fonts.alias` file as previously described, and then enter the following command:

```
% xset +fp <dir>
```

The directory variable should be an absolute pathname. For example, use the `/usr/users/jruser/xfonts` pathname, instead of the `./xfonts` pathname.

X servers running on X terminals may have to access the `fonts.alias` file in a different way than described above. If you use an X terminal and you want to install a `fonts.alias` file, consult the documentation for your X terminal to determine how to do so.

If you use Digital servers with non-Digital applications, problems may occur because some fonts present on the X11 Release 3 tape and in previous versions of the UWS Unsupported MIT Clients subset on the UWS Unsupported tape have been given new names on the X11 Release 4 tape from the X Consortium. This problem can be solved by using the `font.alias.mit` file to map the Release 3 font names to the Release 4 font names.

To use the `fonts.alias.mit` file, become `root` and enter the following commands:

```
% cd /usr/lib/X11/fonts/MIT
% mv fonts.alias.mit fonts.alias
```

If you do not have the U[DL]XUNMIP430 subset installed, either log out and log

back in to the system, or use the `rehash` command as follows:

```
% xset fp rehash
```

The `fonts.alias` file is documented in more detail in the `dxmkfontdir` reference page.

Note

Use of non-Digital X servers with Digital applications is not supported by Digital.

4.5 User Environment

The following sections discuss applications connected to the user environment.

4.5.1 Bookreader and Screen Resolution

Bookreader books are built for display on workstations using 75-dpi (dot-per-inch) fonts on monitors with a screen resolution of 75 dpi. The spacing in books may be irregular when they are displayed using 100-dpi fonts or on monitors with a screen resolution of 100 dpi.

You may also see problems with spacing if the resolution of your workstation monitor is set incorrectly. Beginning with ULTRIX and UWS Version 4.2A, the default screen resolution for some monitors was changed. If you see problems with spacing in books on Version 4.2A or later releases that didn't occur on earlier releases, see Section 4.1.8 for information on setting the correct resolution for your monitor.

4.5.2 DECwindows Applications

Renaming the binaries of DECwindows applications is not supported.

4.5.3 Window Manager - dxwm

The following sections discuss the DECwindows Window Manager.

4.5.3.1 Lowering a Window with dxwm

If an application creates a window with the `override_redirect` attribute set to true, and then that window is lowered by the application or user, the window will seem to vanish. Actually, the window has been lowered behind an intermediate level `dxwm` window that covers the screen. There is presently no workaround for this problem, except to use a window manager other than `dxwm`.

4.5.3.2 UnmapNotify Event Not Generated

The `dxwm` window manager does not generate an `UnmapNotify` event when a window is iconified.

4.5.3.3 Delay in Appearance of Windows

The window manager may prevent applications from mapping drawables to the display until an existing application window is unfocused and refocused.

4.5.3.4 Naming Windows and Icons

The dxwm window manager currently lets you name windows and icons. Client programs can specify their names by using the XA_WM_NAME and XA_WM_ICON_NAME properties defined by DECwindows.

The dxwm window manager uses the values of these properties when it decorates the client window or icon. If the client does not specify a value for the XA_WM_ICON_NAME property, dxwm uses the one set for the XA_WM_NAME property for the icon as well.

The following example shows how to define the name of a window by using the Xlib XChangeProperty function:

```
main ( ) {
Window win;
int winW, winH;
int winX, winY;
XSetWindowAttributes xswa;
       /*  open the display  */
winW = 600;
winH = 600;
winX = (DisplayWidth(dpy,0)-winW)>>1;
winY = (DisplayHeight(dpy,0)-winW)>>1;
xswa.event_mask = 0
xswa.background_pixel = BlackPixel(dpy,0);
win = XCreateWindow(dpy,RootWindow(dpy,0),winX,winY,winW,winH,0,
       DefaultDepth(dpy,0),InputOutput,DefaultVisual(dpy,0),
       CWEventMask | CWBackPixel, &xswa);
XChangeProperty (dpy,win,XA_WM_NAME,XA_STRING,8,PropModeReplace,
       "My Window",9);
       .
       .
       .
}
```

In this example:

- The XA_WM_NAME property specifies that property is to be changed, while XA_STRING specifies its data type.
- The argument 8 indicates that the data is in 8-bit format.
- The PropModeReplace argument indicates that the previous associated information is to be discarded.
- The string "My Window" is the new name for the XA_WM_NAME property, and the argument 9 indicates that the string has nine characters.

To change the icon name, substitute XA_WM_ICON_NAME for XA_WM_NAME.

4.5.4 Operator Account Inaccessible from the Start Session Window

The operator cannot log in to the operator account from the Start Session login prompt. To invoke operator services, the operator can log in to any account, then use the su command to log in as *operator*. You can create a special account for the

operator or use the root account. Remember to use the passwd command to create a password for the operator account. This is necessary even though the operator does not log in directly to the operator account.

4.5.5 Viewing and Mailing DDIF Files with Missing External References

DDIF files (created by applications such as DECwrite), may contain references to external files containing fonts or other data. The view fails if you try to view a DDIF file with any application linked with libdvr.a (such as dxvdoc and dxmail) containing references to external files that cannot be located on your system.

The dxvdoc application returns a message such as "Unexpected error converting aggregate" or "CDA could not open file." The dxmail application displays a blank viewer window, and no error message. If you have an application using libdvr.a, the application will receive an error status such as CDA$_OPENFAIL.

If you mail a DECwrite file that contains references to a system-library style file to a user on a system without DECwrite installed, the mail message will not be viewed successfully on the receiving end. (System-library style files are installed as part of a DECwrite installation in /usr/lib/cda/*.doc_style.) The user will see a blank viewer window, and no message in dxmail.

System-library style files are not packed up in the mail message because they are assumed to be a system resource that is present on each system. Normally, files referenced externally are packed up in the mail, unless the references are stored as "no-copy" such as DECwrite references to system-library style files.

To get around the problem, make sure your files with external references have all of their externally referenced files present on the machine on which you want to view them.

Use the cdoc utility to determine if your file refers to external files. The following command converts a DDIF file to analysis format and searches the output for any external reference labels:

```
# cdoc -s ddif -d analysis myfile.doc | grep ERF_LABEL
```

If you find that there are references to files not present on your system, copy over the files to be able to successfully view the entire document. If you are missing DECwrite system-library style files, copy them from a system with DECwrite installed from /usr/lib/cda/*.*.doc_style.

4.5.6 Calendar - dxcalendar

The following notes discuss the dxcalendar program:

- Repeat entries are lost when you convert a dxcalendar data file from ULTRIX Worksystem Software Version 2.2 (VAX) to higher versions of ULTRIX Worksystem Software.
- The dxcalendar program appears to produce PostScript files that do not conform to Adobe PostScript conventions, but this is not the case. For example, if you use the dxcalendar File Print... menu to create an image file of the calendar and then load this file into the PostScript Previewer, the Previewer may display a warning message, but the image is then displayed.

4.5.7 Visual Differences Program - dxdiff

Selecting files in the dxdiff program does not automatically cause the differences to be shown. The correct procedure for using the dxdiff program to list the differences between two files is as follows:

1. Select the first file (left).
2. Select the second file (right).
3. Select the Do Differences item from the Differences pulldown menu. Note that this may take some time and no clock face icon (an indication that the computer is busy) becomes visible.

4.5.8 DECwindows Debugger - dxdb

The dxdb program does not allow you to view a module that has a left bracket character ([) as a part of its name.

4.5.9 Compound Document Viewer (CDA) - dxvdoc

When you invoke dxvdoc, you may get the following error message, but the CDA Viewer is invoked successfully:

```
X Toolkit Warning: Idb__BM_GetRecord: Get block failed - DRMFailure
```

4.5.10 Mail - dxmail

The following notes discuss the dxmail program:

- The dxmail command uses the .Mail file to store user customization information. If a user specifies the .Mail file in the Path: entry of the .mh_profile file, a conflict is created. To avoid this conflict, do not use .Mail as the mail directory for the MH mail handler.
- The MH directories have been moved. As a result, to use dxmail, mh, mhe, or any MH-based mailer, you will need to add the /usr/bin/mh file to your existing path. If you plan to work on a version earlier than Version 4.2, you will also need to include the /usr/new/mh file after the /usr/bin/mh file.
- If you are an MH user, and you plan to begin using the dxmail command, be aware of the following problems:
  - If you have been using the $HOME/.Mail directory to store your MH folders, you must rename this directory before using dxmail. Using this directory to store MH folders will clash with the $HOME/.Mail file, which is the application resource file used by mail applications. When you have renamed the $HOME/.Mail directory, add the new directory name to the Path: entry in your $HOME/.mh_profile file. For example, if you rename your directory to Newmail, you would make the following entry in your $HOME/.mh_profile file:

    ```
    Path: Newmail
    ```

  - If you are running dxmail, you must not use the MH commands at the same time. The dxmail application is not automatically updated when you use an MH command. For example, if you create a folder using an MH command, dxmail does not see the new folder automatically. Therefore, if you use

dxmail on your workstation and then need to access your mail using MH commands from a nongraphical terminal, exit dxmail before using the MH commands.

To exit dxmail, choose the Exit item from the File menu. This closes the Main window and commits any changes such as moving or deleting messages. If you have more than one dxmail Main window open, you must close each one separately.

- The behavior of the Sort By Date option on the dxmail Maintenance menu has changed. Sort By Date sorts the messages within a folder into chronological order, but it does not remove gaps in the numbering sequence. In order to renumber messages sequentially and remove gaps in the numbering, use the Renumber Messages item on the Maintenance menu.

  For example, assume a mail folder contains the following messages:

```
 1  08/03 smith@waste        Meeting on Thursday
15  23/03 garett             New information
23  04/03 docherty           A brief update
```

  The Sort By Date option sorts the messages as follows:

```
 1  04/03 docherty           A brief update
15  08/03 smith@waste        Meeting on Thursday
23  23/03 garett             New information
```

  The Renumber Messages option renumbers the messages as follows:

```
1  04/03 docherty           A brief update
2  08/03 smith@waste        Meeting on Thursday
3  23/03 garett             New information
```

- The dxmail command is susceptible to file system space shortages. If the file system is full, dxmail is not able to write the output of a folder scan operation into a cache file in the folder directory. The dxmail program thinks that the cache file contains the correct information and does not display an error message; however, no messages are displayed in the folder. If no messages are displayed in a folder when you know that it contains messages, delete the .decxmailcache file and the .mh_sequences file from the folder directory, then rescan the folder. The messages should be displayed.
- If mail displays the wrong message numbers, rescan the folder.
- The dxmail application does not notify users of problems viewing compound documents. Using the CDA Viewer for example, unresolved external reference problems result in an empty view window being displayed in the dxmail read window. No message is given to the user when a bad file is sent.

  In addition, if dxmail crashes and the Quit option is selected from the Continue/Quit/CoreDump&Quit dialog box, all workstation windows will freeze until all dxmail windows disappear.

4.5.11 Paint - dxpaint

The following notes discuss the dxpaint program:

- Drawing rectangles (or squares) whose line length is smaller than the specified line width may leave some garbage in the pixmap. If while rubberbanding a rectangle you decide that you do not want it, complete the rectangle by releasing

MB1, and then perform an Undo.

However, if you attempt to rubberband a rectangle whose line is zero width or length, garbage might appear in the pixmap that an Undo will not clear.

- The tilde (~) is not translated as the user's login directory if it is used as part of the file specification. As a result, if a tilde is used, `dxpaint` responds with a message that the file could not be saved (or opened). Always use the full pathname when opening or saving files.

  A tilde is parsed as the user's login directory if it is used as part of the file filter.

- When you invoke `dxpaint`, you may get the following error message, but Paint is invoked successfully:

```
X Toolkit Warning: Idb__BM_GetRecord: Get block failed - DRMFailure
```

4.5.12 User Executive - dxue

When you invoke `dxcardfiler`, `dxpaint`, or `dxvdoc` from within `dxue`, a warning dialog box may appear before the application is invoked. The warning dialog box may be one of two types, depending on the hardware. In one type of warning box, the X Toolkit warning message is displayed in a line across the top of the dialog box. The OK button is not selectable, and if you click on the Cancel button to acknowledge the warning and dismiss the warning box, the application exits. If you exit the application, the warning box is still displayed, and you must select the OK button to remove the box; the Cancel button is not selectable.

The other type of warning dialog box displays the application name and error message in column fields. If you click on the Cancel button to acknowledge the warning and remove the warning box, the button is made inactive, but the application is unaffected and the warning box is still displayed. In this case, you will need to quit the application to remove the warning box.

4.5.13 PostScript Previewer - dxpsview

The following sections discuss the PostScript Previewer.

4.5.13.1 PostScript Previewer and PostScript Comments

The PostScript Previewer requires comments in PostScript documents to adhere to the Adobe Document Structuring Conventions if the comments are specified in the Conventions. For example, the comment `%%Page: 1` will cause the Previewer to crash. The correct form for this comment is `1%%Page: 1`.

4.5.13.2 Viewing Concatenated Documents with the PostScript Previewer

By default, the `dxpsview` program only recognizes the first PostScript document in a series of PostScript documents. To view a series of PostScript documents, click on the Use Comments option in the Options menu. You can then view concatenated PostScript documents.

4.5.13.3 Paging Problem with PostScript Previewer

When paging through a PostScript file with `dxpsview`, the page counter may incorrectly report the page being viewed. This can be caused by clicking quickly and repeatedly on the Next Page or Previous Page button. To avoid this, pause after each Next Page or Previous Page button click to allow the Previewer to stay synchronized, or use the Page menu, Display... item to jump directly to the page desired. To resynchronize the page number display when it becomes inaccurate, use the File Revisit option.

4.5.13.4 Scale Factors Larger than 2.0

Due to a swap space memory limitation, selecting a scale factor greater than 2.0 can crash the X server.

4.5.13.5 PostScript File Identification

Many PostScript files created by document formatters, such as those used with `ditroff`, adhere to the Adobe page description commenting conventions. You can verify that a file is properly commented (and thus positively identifiable as a PostScript file) by checking to see that the following is the first line:

```
%!PS-Adobe-X.0
```

The X.0 is a PostScript version number.

4.5.13.6 Viewing Uncommented PostScript Files

The PostScript Previewer now lets you view files whose first two characters are not the following:

```
%!
```

If you open such a file for viewing, a warning message appears, and you are asked to confirm that the file is a PostScript file.

4.5.14 User Account Manager - dxuam

The following notes discuss the User Account Manager:

- Only users with superuser privileges can create accounts, modify accounts, or remove accounts.
- On VAXstation 3100 workstations, selecting Update Password and Group Source from the Options pulldown menu may cause `dxuam` to hang.
- The text in the button labels may be displayed incorrectly on workstations with 100 dpi monitors, such as DECstation 5000s.

4.5.15 Cardfiler - dxcardfiler

When you invoke `dxcardfiler`, you may get the following error message, but

Cardfiler is invoked successfully:

```
X Toolkit Warning: Idb__BM_GetRecord: Get block failed - DRMFailure
```

4.5.16 Session Manager - dxsession

The following notes discuss the Session Manager, `dxsession`.

4.5.16.1 Limitations on DECnet Nodenames Used as Host Names

You can use the Session Manager Customize Security dialog box to add and delete names of host computers that are allowed to connect with your workstation. If the connection between your workstation and the remote host is through DECnet, two colons (::) are required after the host name, as in the following example:

```
localhost
dahlia::
peony.tech.com
```

4.5.16.2 Login with User's Home Directory as an NFS-Mounted File System

A problem may occur if a user's home directory is on an NFS-mounted file system that is not available when that user attempts to log in (for example, the file server is down). If the file system is mounted "hard,intr", the Start Session dialog box remains displayed, and the user must press the interrupt key (Ctrl/C by default) to complete the login. If the file system is mounted "soft" the login times out and proceeds without user intervention. In either case, an error dialog box displays the following message:

```
"No directory! Logging in with home=/"
```

Under these conditions the `$HOME` shell environment variable is set to the root directory, because the normal home directory is not available. However, if the root shell startup file attempts to set `$HOME` to some NFS-mounted file system, or if that file system was mounted other than "hard,intr" or "soft," then the login process may hang indefinitely.

4.5.16.3 Pause Feature in Enhanced Security Mode

Under BSD security, the password required to unpause a workstation is the current password. If you change your password during a work session and then pause the session using the Pause feature under the Session Manager's Session menu, the new password will unpause the workstation. However, when running under enhanced security, the password required to unpause a workstation is the password used to log on to the workstation. In this case, if you change your password during a work session and then pause the session, you must use your old password, rather than your new one, to regain access to the workstation.

4.5.16.4 Specifying Multiscreen Function in the Customize Menu

If you are working with multiple display units, you can use the Session Manager Screen Number item in the Customize pulldown menu to specify a default screen for all subsequently executed applications, or you can specify that you want to be prompted for the screen number each time you run an application.

4.5.16.5 Setting Pointer Acceleration

The session manager pointer acceleration value includes a value for threshold and overrides the server `-t` option. The `-t` option for the server command defines how many pixels the mouse can move before mouse acceleration begins. If you set the mouse threshold using the `Xws -t` command and then use the Session Manager Customize Pointer... dialog box to change the pointer acceleration, the value for Session Manager pointer acceleration overrides the value defined by the threshold option in the server command. The Session Manager uses default threshold and acceleration values to define the pointer acceleration; the Session Manager threshold value cannot be set by the user.

4.5.16.6 Print Screen

The following notes discuss the Session Manager Print Screen feature:

- Prior to Version 4.2, Print Entire Screen and Print Portion of Screen options created an output file `decw_printscreen.tmp` in the user's home directory, and deleted it when the file was printed or when the Cancel button was clicked.

  Starting with Version 4.2, the file name specified in the Customize Print Screen dialog box is always used when creating an output file (whether using a capture or print option). This file will not be deleted under any circumstances, even if the user selects Cancel. Because the file name is reused in later Print Screen operations, it is recommended to rename a capture file if it is to be saved indefinitely.

- Correct behavior that may be unexpected or not clearly documented:
  - Color PostScript is not supported.
  - Grayscale PostScript and grayscale and color sixel output are dithered.

    Video screens and printers use different color rendering methods: additive for video screens and subtractive for printers. Color impact, inkjet, and desktop PostScript (thermal wax) printers use nonblending, opaque inks, so colors cannot be blended the way an artist mixes paints (or the way the eye looking at a video screen mixes colored light). Instead, the printers lay down patterns of colored dots and lines (called "dithering"). The density of the various inks in a given area "fools the eye" into seeing color blends, but the dots and lines are still visible. As a result, some image detail is lost on the page, and the dots and lines create extraneous detail of their own. This is a characteristic of subtractive color printing technology rather than a deficiency in the Print Screen software.
  - Black and white output selections are "high contrast" in nature. Objects that appear gray or in color on the monitor will be shown as black (for objects that are darker than 50 percent gray) or white (for objects that are lighter than 50 percent gray). Adjacent objects that are insufficiently contrasting will blend into each other or disappear in the final output (such as light red text on a light blue background).
- Monochrome (black and white) display workstations:
  - Color and grayscale output selections are not disabled in the Customize Print Screen dialog box. Selecting one of them results in a black and white screen capture.
  - Sixel output may appear with inverted tonal values (that is, like a photographic negative) on some workstation configurations. Use the "Ribbon

and Toner Saver" buttons in the Customize Print Screen dialog box to correct this problem.

- DECsystem 5000 Model 200 workstations with 24-plane color display options:

  Sixel output (all types) is severely distorted when displayed in a terminal window with the `cat` command.

  On a 24-plane system, the colors in Print Screen output may be incorrect if an application has installed a pseudocolor map.

- All color workstations with 8-plane color display hardware:

  Color DDIF screen captures, when viewed with the CDA Viewer, may have color interference patterns superimposed on the screen background if the default (or similar) black and white herringbone screen background pattern is used.

- All color workstations:

  Color sixel output displays with incorrect colors in a terminal window (using the `cat` command). It will also be displaced to the right by a number of character spaces, and a string of extraneous characters will appear to the immediate left of the top left corner of the sixel image.

- All workstations:

  Sixel output of all types, when displayed in a terminal window using the `cat` command, may be followed by random alphabetic characters, appearing on a separate line to the left of the shell prompt.

- When performing Print Portion of Screen or Capture Portion of Screen operations on a multiscreen system, you must specify the screen on which you want to define the portion to print or capture. You can either specify a default screen for all subsequent Print Screen operations or specify that you want to be prompted for the screen number for each Print Screen operation. Once you specify a screen, all Print Screen operations are performed on that screen. You can move the pointer to another screen during the Print Screen operation, but the rectangle that defines the portion of the screen to print or capture remains on the specified screen.

  To specify a default screen for a Print Screen operation, select screen 0 or 1 under the Print Screen Use Screen Number option in the Session Manager Customize Screen Number dialog box. To specify that you want to be prompted to define a screen for a Print Screen operation, select the Prompt for Screen Number option under Print Screen in the Session Manager Customize Screen Number dialog box.

- Specifying `/dev/null` as the destination file for a Print Screen Capture operation kills the session.

4.5.16.7 Setting the Window Screen Background with the Customize Menu

When you are using multiple displays, the Window Screen background pattern option under the Customize menu does not work properly. It does not reflect the pattern until the Apply or OK option is selected. It does not update if the Default option is selected.

4.5.17 DECterm Terminal Emulator - dxterm

The following sections discuss the DECterm terminal emulator, `dxterm`.

4.5.17.1 DECterm Resize Problem

If you change the `DECterm.main.terminal.borderWidth` resource in your `.Xdefaults` file or in a file created to save customized DECterm settings, such as `default.DECterm`, and the Customize Window options Auto Resize Terminal and Auto Resize Window are enabled, the DECterm window will resize itself until it becomes very small. To avoid this, either do not change the default value of the border width, or, if you need to reset the width of the border, click on the Auto Resize Window option in the Customize Window menu to disable automatic window resizing.

4.5.17.2 Compose Character Sequences

In a DECterm window, you can use compose sequences to create special characters. A compose sequence is a series of keystrokes that creates characters that do not exist as standard keys on your keyboard. To compose a special character, follow these steps:

1. In the DECterm Customize Keyboard dialog box, click on the VT300 Mode, 8-Bit Controls button. Make sure that the 8-Bit Multinational Characters option is also active. Apply these changes.
2. Enable 8-bit ASCII character support by placing the following command in your `.cshrc` file or keying it in at the keyboard:

   ```
   stty pass8
   ```

3. Compose a character by simultaneously pressing and releasing the Compose key and the space bar, and then typing in the characters for the desired special character. For information on the available special characters, see Appendix A in the *ULTRIX Worksystem Software DECwindows Desktop Applications Guide.*

4.5.17.3 New DECterm Customize Keyboard Option

The DECterm Customize Keyboard dialog box now includes an option to set the F11 key. On the North American and Dutch keyboards, the F11 Key Sends F11 option is set. The default setting causes the F11 key to send F11. You can override this by clicking on the F11 Key Sends Escape option. This causes the F11 key to function as an escape key.

4.5.17.4 DECterm VT300 Emulation

The `dxterm` application is a terminal emulator that supports many of the features of Digital VT-series terminals. By default, `dxterm` emulates a VT300-series terminal. However, the model that `dxterm` uses to handle specific terminal operating attributes may not be the same as the model that a VT300 uses; there may be visual differences in the way terminal operating attributes are handled that are due to differences in workstation and character-cell terminal imaging models, as well as to differences caused by variations in workstation capabilities.

As a result, some terminal features may be implemented differently or not at all. For example, while a VT300 may change the color of text to show bolding, `dxterm`

substitutes a different font set to show bolding.

Note also that, while you can use the Terminal ID option of the dxterm Customize General dialog box to change the terminal ID of the dxterm you are using, the set of features of dxterm is the same for all terminal IDs, although the the way in which the features are implemented may vary. For example, the VT100 has no function keys, so the F11 key sends Escape when dxterm is configured to send a VT100 ID.

4.5.17.5 VT420 Terminal Emulation

DECterm does not support emulation of the new Digital VT420 video terminal. To run an application intended for use on VT420 terminals, make sure that the DECterm window is emulating a VT300-series terminal. Most VT420 control codes are compatible with VT300 emulation. To make sure the DECterm is in VT300 mode, perform the following steps:

1. Select the Customize General... menu option.
2. Choose either VT300 Mode, 7-Bit Controls or VT300 Mode, 8-Bit Controls. Note that, in general, selecting VT300 Mode, 8-Bit Controls is unnecessary, as dxterm can display 8-bit characters without being in 8-Bit Controls mode.
3. Click on the the Apply button.
4. Click on the OK button.
5. Change the TERM environment variable to vt400 or vt420 as follows:
   - If you are using the csh shell, enter the following command:

     ```
     % set term=vt420
     ```

   - If you are running a sh, sh5, or ksh shell, enter the following command:

     ```
     % TERM=vt420 ; export TERM
     ```

6. Unset the TERMCAP environment variable in order to execute the resize command; then execute the resize command.
7. Execute the tset command, as follows:

   ```
   % tset
   ```

Note that these steps are a workaround. If the application depends on features unique to the VT420 terminal, these steps will not make them work with DECterm.

4.5.17.6 The resize Command

The following notes discuss the resize command.

- The resize command does not reset the TERMCAP environment variable correctly if the TERM environment variable has been reset. To work around this problem, you first need to unset the TERMCAP environment variable. To unset the TERMCAP environment variable if you are using a csh shell, enter the

following command:

```
% unsetenv TERMCAP
```

If you are using a `ksh`, `sh`, or `sh5` shells, enter the following command:

```
% TERMCAP= ; export TERMCAP
```

- The `resize` command does not work if the TERMCAP environment variable has been previously set. For example, a user's `~/.login` or `~/.profile` file may preset TERMCAP to point to a private termcap file. If TERMCAP is already set, the `resize` command will fail with the following error message:

  ```
  resize: No 'co#'
  ```

  A partial workaround for this problem is to unset the TERMCAP environment variable. Note that unsetting the TERMCAP environment variable means, however, that the `resize` command will use the system `/etc/termcap` file instead of a private termcap file. There is no way to use `resize` with a private termcap file.

4.5.17.7 The DECterm Auto Repeat Option

You can enable or disable the Auto Repeat character option using the Session Manager Customize Keyboard dialog box or the DECterm Customize Keyboard dialog box. Using the Session Manager Customize Keyboard dialog box to change Auto Repeat may result in inconsistent Auto Repeat settings in multiple DECterm sessions. Note also that the Auto Repeat character setting of other windows, such as the Mail window, is set to that of the last DECterm window that had input focus.

4.5.17.8 User-Defined Key Definitions (UDKs)

UDK definitions supported by `dxterm` allow you to define shifted function keys (F6 to F20, including Help and Do) using escape sequences.

Because UDKs use shifted function keys, you can define these keys for any desired purpose without conflicting with the unshifted function keys that are reserved for operating system and application use. You can choose to lock your UDKs through `dxterm` by selecting the General item from the Customize menu, or by using the UDK Definition Device Control String (DCS). Thus, applications cannot count on being able to redefine these keys.

Do not confuse UDKs with operating system features such as the DEFINE/KEY command on the VMS system and `termcap` and `terminfo` databases on the ULTRIX system. Those features work with unshifted function keys (or, in the case of `termcap` and `terminfo`, with any keyboard key that sends a known input sequence). UDKs use the shifted function keys that send input sequences that look to the application and operating system as if they were typed by you.

UDKs work only when `dxterm` is operating in VT300 mode. To check this, go into the Customize General dialog box, select General from the Customize menu, and make sure that the Terminal Mode is VT300 Mode, 7 Bit Controls or VT300 Mode, 8 Bit Controls.

In the description that follows, 8-bit characters are given in terms of their hexadecimal ASCII values using the C notation. For example, `0x41`, decimal 65, is the ASCII code for the letter A. (The 8-bit ASCII character set is given in the *UWS DECwindows Desktop Applications Guide.)*

UDK definitions are not typed directly at the shell prompt. (If they were, they would be intercepted by the operating system and not seen by dxterm.) They must be output from the computer to dxterm. This can be accomplished in a number of ways. One way is to create a file, using any text editor that allows you to enter nonprinting characters such as ESCAPE. Then copy the file to the terminal, using cat. Another way is to output text strings directly containing the UDK definitions, that is, from a shell script using echo. An additional way is to output the strings to the terminal using a program. See the following examples.

To define one or more UDKs, use the following escape sequence:

```
DCS Pc ; Pl | Ky1/St1 ; ... Kyn/Stn ST
```

- DCS is the Device Control String Introducer (ASCII 0x90). In a 7-bit environment, DCS can be sent as ESC P (ASCII 0x1b and ASCII 0x50, with no space in between).
- Pc is the clear parameter. A value of 0 clears all keys before loading new values (that is, sets them all to empty). A value of 1 clears just the keys that you are loading. If you do not specify Pc, it defaults to 0 (that is, all keys that are not defined in the device control string are cleared).
- Pl is the lock parameter. A value of 0 locks the keys. If you want to load new values into the keys later, you must unlock the keys from the Customize General menu within DECterm. A value of 1 does not lock the keys. The keys are unlocked and can be redefined with another DECUDK string. If you do not specify Pl, it defaults to 0 (that is, the keys are locked after the device control string takes place).

Note

If Pl is 1 and the keys are already locked, nothing happens. This is because once the keys are locked they can only be unlocked through the Customize General menu in dxterm (that is, they cannot be unlocked from a program).

Note that Pc and Pl are separated by a semicolon.

The following sequence clears all UDKs without locking them:

```
DCS 0 ; 1 | ST
```

The following sequence locks all UDKs without clearing them:

```
DCS 1 ; 0 | ST
```

- The vertical bar (|) is the final character (ASCII 0x7c). The final character separates the clear and lock parameters from the key definition string.
- Ky1/St1;...Kyn/Stn are the key definition strings; you can have zero or more of these, each of which defines a single shifted function key. Each string consists of a string selector number (Kyn) and a string parameter (Stn), separated by a slash (ASCII 0x2f). A semicolon (ASCII 0x3b) separates different strings.

  The key selector value (Kyn) indicates which key you are defining:

Key	Kyn Value	Key	Kyn Value
F6	17	Help	28

Key	Kyn Value	Key	Kyn Value
F7	18	Do	29
F8	19		
F9	20	F17	31
F10	21	F18	32
		F19	33
F11	23	F20	34
F12	24		
F13	25		
F14	26		

Note that these are not ASCII codes but digits, so the code for F18, 32, means the digit 3 (ASCII `0x33`) followed by the digit 2 (ASCII `0x32`).

The string parameter (`Stn`) for each string definition is the encoded definition of the key being defined; that is, the sequence of ASCII codes that will be sent to the application. String parameters consist of a series of hexadecimal pairs, one pair for each character in the definition. Each hexadecimal pair defines an 8-bit character according to its value in the ASCII table; the hexadecimal pair can be uppercase (for example, `4E` for the letter "N") or lowercase (for example, `4e` for the letter "N").

- `ST` is the string terminator (ASCII `0x9c`).

Note

You cannot define the shifted function key F11 through `dxterm`.

Consider the following guidelines when loading UDKs:

- Before loading new definitions, it is a good idea to clear the old key definitions without locking them and then load the new definitions in another DECUDK string. This will prevent the memory used for UDK definitions from becoming fragmented.
- If you redefine a key, the old definition is lost. This may free up some space if the new definition uses fewer bytes than the old one.
- There are two ways to lock UDKs, but only one way to unlock them. To lock UDKs you can use the Lock UDKs toggle button in Customize General or a DECUDK control string. To unlock UDKs, you must use the Lock UDKs toggle button.
- The default value for each key definition is empty. When you clear UDKs, they are empty.
- You cannot save UDK definitions using `dxterm`; the definitions are lost when you exit the `dxterm` window. Because of this, it is a good idea to load the key definitions that you want in your `.login` file.
- An invalid hexadecimal pair in a DECUDK string stops a UDK load sequence. When a load sequence stops (due to an error or other cause), `dxterm` saves any keys already loaded and displays the rest of the DECUDK sequence on the screen.

The following example is an ULTRIX shell script that demonstrates how to define more than one shifted function key in the same DECUDK Device Control String

(DCS). Note that DCSs can continue over more than one line, as shown in this program. This program was written to execute in VT300 mode, 7-bit controls.

This example defines the shifted function key F6 to be `ls-l` <CR>, where <CR> indicates a carriage return. It also defines the shifted function key F7 to be `date` <CR>. In the example that follows ^[is the escape character as it appears when inserted using the text editor `vi`. To enter the escape character in `vi` , while in insert mode, first enter Ctrl/V then press the Escape key.

After you execute the shell script the shifted functions keys F6 and F7 are defined.

The ULTRIX shell script follows:

```
echo '^[P'                  # DCS Introducer
echo '1;'                   # Pc = 1, Clear only keys that are defined
echo '1'                    # Pl = 1, Do not lock the shifted function keys
echo '|'                    # | = Final Character
echo '17/6c73202d6c0d'      # Ky1/St1 = F6/ls -l<CR>
echo ';18/646174650d'       # Ky2/St2 = F7/date<CR>
echo '^[\'                 # ST = String Terminator
```

The following example shows how to define these same two keys, function key F6 and function key F7, from a C program instead of a shell script on ULTRIX:

```
#define ESCAPE '\033'

main()
    {

    /*
     * Send the UDK introducer that does not clear or lock UDK
     * definitions.  Remember that the default for the clear and
     * lock parameters is 0, so if these parameters are omitted the
     * UDKs will be both cleared and locked.
     */

    printf( "%cP1;1|", ESCAPE );

    /*
     * Define shift-F6 to be "ls -l" terminated with a Return.
     */

    printf( "17/6c73202d6c0d" );

    /*
     * Define shift-F7 to be "date" terminated with a Return.
     */

    printf( ";18/646174650d" );

   /*
    * Terminate the DECUDK command with <ESC>\ and send a newline.
    */

    printf( "%c\\\n", ESCAPE );

    }
```

In this example the final `printf` was terminated with a line feed, but in fact, line feeds and spaces could have been inserted at any point in the DECUDK device control string except for within the introducer sequence (in this case <ESC>Pl). Note that 8-bit characters can be defined as well as 7-bit characters, although 7-bit ASCII has been used for convenience in this example. If you transmit 8-bit UDKs, then `dxterm` must be in 8-Bit Controls mode, and, if this is done on a remote system, the

communications path must be an 8-bit path.

4.5.17.9 Command-Line Resource Specification

The `-xrm` option of `dxterm` that specifies a resource string to be used does not work properly.

4.5.17.10 dxterm Does Not Clear /etc/utmp

Do not end your login session by selecting the Quit menu item from the Session menu in the Session Manager if there are one or more `dxterm` sessions running. First log out out of each `dxterm` session, then choose the Quit menu item.

Failure to follow this procedure will leave `ttyXX` entries in `/etc/utmp`. Thus, users remotely logged into your workstation will see erroneous data when using commands such as `w` and `who`.

4.5.17.11 Using ioctl with sigio Hangs dxterm

Simultaneously using `ioctl` and the `sigio` signal will hang the `dxterm` session on a workstation and will eventually hang the system such that nothing works (including any network connections). You will have to reset and reboot the system. The arguments to the `ioctl` and `fnctl` calls are as follows:

```
if((filed < 0) || (sigvec(SIGIO, &iovec, 0) == -1) ||
                  (fcntl(filed, F_SETFL, FNDELAY|FASYNC) == -1 ) ||
                  (ioctl(filed, FIONBUF, &_nbufcnt) == -1)) {
          return RETERR;
}
```

The program will access the tape and read data. The signal never arrives.

4.5.17.12 Using System V Shell (sh5) as Default

A DECterm window does not go away when you attempt to log out if you are using AT&T's System V shell (`sh5`) as the `/etc/passwd` default shell and start a DECterm session using `dxterm -ls`.

To remove the window from the screen, iconify it or end the session.

Layered Products Notes 5

This chapter discusses the layered products available for ULTRIX and UWS Version 4.3.

5.1 Product Changes Affecting Layered Products

The following sections describe changes in the product that affect layered products.

5.1.1 DEC PHIGS Incompatibility

DEC PHIGS Version 2.3B is incompatible with ULTRIX and UWS Version 4.3. Therefore, do not install (or upgrade) to ULTRIX and UWS Version 4.3 if you are running DEC PHIGS Version 2.3B. Instead, run your current software until you receive DEC PHIGS Version 2.3C, which is compatible with the new version of ULTRIX and UWS. Then, install ULTRIX Version 4.3 and DEC PHIGS Version 2.3C.

5.1.2 Reinstalling Layered Products

You may have to reinstall some layered products after you install or upgrade to ULTRIX and UWS Version 4.3. Any layered product that directly accesses kernel data structures must be reinstalled. Review the documentation for your layered products to determine whether they must be reinstalled.

5.1.3 Installing DECnet/OSI Version 5.1 on an Upgraded System

The DECnet/OSI for ULTRIX Version 5.1 product includes several replacement kernel objects, used to patch ULTRIX Version 4.2 systems for necessary functionality. Installing this product on an ULTRIX Version 4.3 system which has been upgraded from ULTRIX Version 4.2 can lead to an ULTRIX Version 4.3 kernel object being incorrectly replaced with a replacement ULTRIX Version 4.2 kernel object.

To correct the problem, you must copy the original file back and then rebuild the new kernel.

To copy the original file on RISC processors, enter the following command:

```
# cp /sys/MIPS/BINARY/uipc_socket.o.V42 /sys/MIPS/BINARY/uipc_socket.o
```

To copy the original file on VAX processors, enter the following command:

```
# cp /sys/VAX/BINARY/uipc_socket.o.V42 /sys/VAX/BINARY/uipc_socket.o
```

To rebuild the kernel, enter the following command:

```
# /usr/etc/decnetsetup kernel
```

5.1.4 Using libdwt.a when the OSF/Motif Development Environment Subset Is Installed

If you have installed the ULTRIX DECwindows for OSF/Motif Version 1.1.3 Development Environment subset (DXMXM11301 for RISC or DXVXM11301 for VAX) and you wish to compile programs using the DECwindows X User Interface (XUI) Toolkit, `/usr/lib/libdwt.a`, your programs must reference the header files in `/usr/include/X11/dwt` instead of those in `/usr/include/X11`. As a convenience, there is also a symbolic link from `/usr/include/dwt` to `/usr/include/X11/dwt`.

The Intrinsics header files included in the Worksystem Development Software subset (UDWXDEV430 for RISC or UWSXDEV430 for VAX) are at X11R3 level. When the Motif Development Environment subset is installed, the X11R3 Intrinsics header files are moved to `/usr/include/X11/dwt` and X11R4 Intrinsics header files are installed in `/usr/include/X11`. Programs linked against `libdwt.a` require the X11R3 Intrinsics header files.

5.1.5 Reinstall DES Encryption Kit After Upgrade

When a Version 4.2 or Version 4.2A system that is running DES Encryption is upgraded to Version 4.3, `libc` is overwritten (removing the `crypt` library functions).

You must reinstall the Version 4.3 Encryption Kit after you complete the upgrade to Version 4.3.

5.1.6 Upgrading an ULTRIX System with Disk Shadowing Installed

To upgrade ULTRIX Version 4.2 to Version 4.3 when ULTRIX Disk Shadowing has already been installed in Version 4.2, complete the following steps:

1. Back up your system disk before installing any software. See the *Guide to Backup and Restore* for more information about backing up your system.
2. Save your Version 4.2 configuration files by typing the following commands, substituting the name of your system in capital letters for *HOSTNAME*:

   ```
   # cp /sys/conf/mips/HOSTNAME /sys/conf/mips/HOSTNAME.V4.2
   # cp /sys/conf/mips/files.mips /sys/conf/mips/files.mips.V4.2
   ```

3. Follow the installation instructions for the upgrade as described in Chapter 5 of the *Guide to Installing ULTRIX*, through the Section 5.1.6, Loading the Software Subset Updates. However, do not rebuild the kernel. Instead, perform one of the following procedures. The procedure you perform depends on whether or not the required kernel is available.

If during the upgrade procedure the following message was displayed after the subsets were loaded, continue with the section titled Rebuilding a Kernel with

Shadowing when the Required Kernel Is Not Available:

```
Required kernel not available
```

If during the upgrade procedure the `Required kernel not available` message was not displayed, continue with the next section, Rebuilding a Kernel with Shadowing when the Required Kernel Is Available.

5.1.6.1 Rebuilding a Kernel with Shadowing when the Required Kernel Is Available

If you are upgrading a system and the required kernel is available, complete the following steps:

1. Edit the `/sys/conf/mips/files.mips` file to ensure the following lines are in the file, and that these lines are the same as the equivalent lines in the `files.mips.V4.2` file:

```
shad/shad_var.c          optional shd Binary
shad/shadow.c            optional shd Binary
shad/shad_common.c       optional shd Binary
shad/shad_root.c         optional shd Binary
shad/bc_var.c            optional shd Binary
shad/base.c              optional shd Binary
shad/bc_common.c         optional shd Binary
shad/shad_data.c         optional shd Binary
```

2. Be sure to insert the `shad_root.c` line, which must exist to configure your system for ULTRIX Disk Shadowing. (Normally, the `shad_root.c` line is removed by the installation and you must add it here.) If you are not shadowing the root or swap device, you must also insert the `shad_data.c` line. (Do not insert the `shad_data.c` line if you are shadowing the root or swap device.)
3. Follow the instructions in *Guide to Installing ULTRIX* in Section 5.1.7 to complete the upgrade.

5.1.6.2 Rebuilding a Kernel with Shadowing when the Required Kernel Is Not Available

If you are upgrading a system and the required kernel is not available, complete the following steps:

1. Save your file system configuration files by entering the following commands:

```
# cp /etc/fstab /etc/fstab.V4.2
# cp /etc/rc /etc/rc.V4.2
# cp /etc/rc.local /etc/rc.local.V4.2
```

2. If you are shadowing the `root` or the `/usr` partitions, remove all shadow members except the drive-zero shadow members from their associated shadow sets. For example, if you have `root` mounted on `shd8a` (consisting of `rz0a` and `rz1a`), and you have `/usr` mounted on `shd14g` (consisting of `rz0g` and `rz1g`), enter the following commands:

```
# shadrm /dev/shd8a /dev/rrz1a
shadrm: The device has been removed from the shadow set.
# shadrm /dev/shd14g /dev/rrz1g
shadrm: The device has been removed from the shadow set.
```

3. Locate all ULTRIX Disk Shadowing entries from your /etc/fstab file. In the example used in step 2, the /etc/fstab file contains the following entries for shadow sets shd8a and shd14g:

```
/dev/shd8a:/:rw:1:1:ufs::
/dev/shd14g:/usr:rw:1:2:ufs::
```

4. Edit the /etc/fstab Disk Shadowing entries to refer to drive zero. For example:

```
/dev/rz0a:/:rw:1:1:ufs::
/dev/rz0g:/usr:rw:1:2:ufs::
```

   If there are other /etc/fstab Disk Shadowing entries, remove them from the file.

5. Remove all shadowing entries in the /etc/rc and /etc/rc.local files. The entries include lines that run ULTRIX Disk Shadowing utilities, such as shadinit, shaderrd, and shadassm.

6. Complete steps 1 through 7 in Section 5.1.8 of the *Guide to Installing ULTRIX*.

7. At step 8, when the doconfig program prompts you to edit the configuration file, press Ctrl/c. When the machine prompts you to quit, enter y for yes.

8. Edit the /sys/conf/mips/*HOSTNAME* file to ensure the following lines are in the file, and that these lines are the same as the equivalent lines in the *HOSTNAME*.V4.2file:

```
   .
   .
   .
config   vmunix root on shd1a swap on shd1b dumps on rz0b
shadow          shd8a on rz0a and rz1a
shadow          shd9b on rz0b and rz1b
   .
   .
   .
pseudo-device   shd
```

   Use the device identifiers and number of shadow lines present in your *HOSTNAME*.V4.2 file. Note that the device identifiers on the config line do not match those on the shadow lines.

Note

To automatically complete the editing for the configuration file, run the Disk Shadowing script, shadconfigdev.

9. Edit the /sys/conf/mips/files.mips file to ensure that the following lines are in the file, and that these lines are the same as the equivalent lines in the files.mips.V4.2 file:

```
shad/shad_var.c          optional shd Binary
shad/shadow.c            optional shd Binary
shad/shad_common.c       optional shd Binary
shad/shad_root.c         optional shd Binary
shad/bc_var.c            optional shd Binary
shad/base.c              optional shd Binary
shad/bc_common.c         optional shd Binary
shad/shad_data.c         optional shd Binary
```

Be sure to insert the `shad_root.c` line, which must exist to configure your system for ULTRIX Disk Shadowing. (Normally, the `shad_root.c` line is removed by the installation and you must add it here.) If you are not shadowing the root or swap device, you must also add the `shad_data.c` line. (Do not add the `shad_data.c` line if you are shadowing the root or swap device.)

10. Complete the kernel configuration by entering the command to run the `doconfig` program:

```
# doconfig -c HOSTNAME
```

11. Restore your file system configuration files by entering the following commands:

```
# cp /etc/fstab.V4.2 /etc/fstab
# cp /etc/rc.V4.2 /etc/rc
# cp /etc/rc.local.V4.2 /etc/rc.local
```

12. Perform steps 9 through 13 of Section 5.1.8 in the *Guide to Installing ULTRIX* to complete the upgrade.

5.2 Qualified Layered Products

Qualified layered products are products which have been successfully installed and verified as operable on ULTRIX and ULTRIX Worksystem Software Version 4.3. Please note that these products are certified as of 15 August, 1992. Contact your Digital representative for information about other products which may have qualified after 15 August, 1992.

Table 5-1 lists the version numbers for layered products which have been qualified for Version 4.2, Version 4.2A, and Version 4.3.

Table 5-1: Layered Products Qualified on ULTRIX and UWS

Product Name	Version 4.2	Version 4.2A	Version 4.3
DCE Starter Kit for ULTRIX	--	--	1.0
DEC Ada for ULTRIX (RISC only)	--	--	1.1
DEC Ada Preprocessor (for ULTRIX/SQL)	2.0	2.0	--
DEC AVS (RISC only)	2.0	3.0	--
DECbridge 500 Software Microcode	3.0	3.0	--
DEC C (RISC only)	1.0	1.0	1.2

Table 5-1: (continued)

Product Name	Version 4.2	Version 4.2A	Version 4.3
DEC C++ for ULTRIX (RISC only)	--	--	1.0
DEC C Preprocessor (for ULTRIX/SQL)	2.0	2.0	--
DECelx Realtime Tools for ULTRIX (RISC only)	--	--	1.0
DEC Faxserver	--	--	1.0a
DEC FDDIconcentrator 500 Software Microcode	3.0	3.0	--
DEC FDDIcontroller 700 Software Microcode (RISC only)	1.1	1.2	--
DECforms for ULTRIX (RISC only)	--	--	1.4
DEC Fortran (RISC only)	3.0	3.0	3.1
DEC Fortran Preprocessor (for ULTRIX/SQL)	2.0	2.0	--
DEC FUSE for ULTRIX (RISC only)	1.0	1.0	1.1
DEC FUSE EnCase ULTRIX (RISC only)	--	--	1.1
DEC FUSE C++ ULTRIX (RISC only)	--	-	1.1
DEC GKS-3D Development Kit	1.1	1.2	--
DEC GKS-3D Runtime Kit	1.1	1.2	--
DECimage Application Servers	2.0	2.0	--

Table 5-1: (continued)

Product Name	Version 4.2	Version 4.2A	Version 4.3
DECimage Scan Software	2.0	2.0	--
DECinspect for ULTRIX	--	--	2.2
DECmcc Management Station	1.0	1.1	--
DECndu (Network Device Upgrade)	1.1	1.1	--
DECnsr for ULTRIX Server (RISC only)	--	--	1.0, 1.1
DECnsr for ULTRIX Client (RISC only)	--	--	1.0, 1.1
DECnet/SNA ULTRIX 3270 Data Stream Programming Interface	1.0	1.0	--
DECnet/SNA ULTRIX Terminal Emulator	1.1	1.1	--
DECnet-ULTRIX	4.2	4.2	4.2
DEC PHIGS (RISC only)	2.2A	2.3	--
DECpresent (RISC only)	1.0	1.0	--
DEC Printserver Support Host Software	4.0	4.	4.0
DEC Realtime Test Integrator Runtime Kit (RISC only)	1.0	1.1	--
DEC Realtime Test Integrator Developer's Kit (RISC only)	1.0	1.1	--
DECserver 200	3.1	3.1	--
DECserver 250	2.0	2.0	--
DECserver 500	2.1	2.1	--

Table 5-1: (continued)

Product Name	Version 4.2	Version 4.2A	Version 4.3
DECset for ULTRIX (RISC only)	--	--	1.0
DEC Trellis Object System	1.0	1.0	1.1
DECtpu for ULTRIX (RISC only)	--	--	3.0
DECwindows Developer's Kit (for OSF/Motif)	1.1	1.1	--
DECwindows DECnet/SNA ULTRIX 3270 Terminal Emulator	1.0	1.0	--
DECwrite	1.1	1.1	--
DSM for ULTRIX (RISC only)	--	--	1.0a
Interleaf 5	--	--	5.3
License Usage Monitor for ULTRIX (RISC only)	--	--	1.0
LOTUS 1-2-3	--	--	1.1
PATHWORKS for ULTRIX	--	--	1.2
SOFTPC	--	--	3.0a
Storageserver 100 for ULTRIX (RISC only)	--	--	1.0
DEC VUIT for ULTRIX (RISC only)	--	--	2.0b
MUXserver 300 Terminal	1.2	1.2	--
VAXeln Window Server	1.0	1.1	--
Pascal for RISC	1.1	1.1	--
X.25 Native Mode for ULTRIX	--	1.0	--

Table 5-1: (continued)

Product Name	Version 4.2	Version 4.2A	Version 4.3
X.25 Gateway Client for ULTRIX	--	1.0	--
DECnet/OSI for ULTRIX	5.0	5.0	5.1
DEC WAN Device Drivers for ULTRIX	--	2.0	2.1a
DECfonts Typeface Collection	1.1	1.2	1.1
DEC Application Control Architecture Services for ULTRIX	--	2.0	--
ULTRIX Disk Shadowing	--	1.0	1.1
XMedia Tools (RISC only)	--	--	1.1

5.3 Enhanced DEC C Compliance of Library Header Files

DEC C shipped on ULTRIX Version 4.1 with some ANSI support in the library header files. The ANSI compliance of the library header files was enhanced in Version 4.2. In particular, all of the system-supplied library header files now use the ANSI C format for predefined macro names.

The following header files are now ANSI compliant (in addition to the header files listed in the DEC C release notes):

```
<fatal.h>                    <sys/errlog_aq.h>
<sym.h>                      <sys/int64.h>
<cmplrs/gptable.h>           <sys/vmmac.h>
<cmplrs/stsupport.h>         <sys/wait.h>
<sys/acct.h>                 <sys/workstation.h>
```

If you used DEC C on ULTRIX Version 4.1, and you created private copies of header files to fix their non-ANSI-format predefined macro names, you can delete those header file copies and use the Digital-supplied header files.

5.4 DEC FDDIconcentrator, DEC FDDIcontroller and DECbridge

The DEC FDDIconcentrator, DEC FDDIcontroller, and DECbridge are dependent on the DECndu (Network Device Upgrade), Version 1.1 for this version of the ULTRIX operating system.

5.5 DECnet-ULTRIX

DECnet-ULTRIX Version 4.2 supports ULTRIX Version 4.2 and higher.

5.6 DEC PHIGS

The following sections discuss specific features and restrictions concerning the use of PEX workstation types through DEC PHIGS on DECstation/DECsystem 5000 series processors.

5.6.1 Using Save-unders with PHIGS Applications

Due to inadequacies in the backing store code in the MIT sample server, it is not possible to use backing store or save-unders with DEC PHIGS/PEX-based 3D graphics applications.

The DEC PHIGS library automatically disables backing store on windows that are accessed. Save-unders should not be enabled, because this may result in garbage in windows when they are moved or resized. Further, running an application with save-unders enabled can induce server crashes or hangs.

Therefore, disable save-unders when using PEX and PHIGS. You can disable save-unders two ways. First, you can place the line `*saveUnder:FALSE` in your `.Xdefaults` file. Second, you can include the `-su` option on the server command line when you invoke the server.

5.6.2 Anti-Aliasing Modes

There are three available settings for anti-aliasing, each with a separate set of functionality and restrictions. The three modes are:

Mode 0

No anti-aliasing is performed. No restrictions due to anti-aliasing exist when mode 0 is used.

Mode 1

Lines are 2.5 pixels wide. Anti-aliased lines are blended from the line color to the anti-aliasing background color (color table entry 0).

Pixels are written only when the computed pixel value is greater than the value of the existing pixel. Overlapping lines are distorted when arbitrary colors are used because the numerical value of a pixel is not necessarily a good indication of pixel brightness (for example, the values of pixels within a `PseudoColor` visual are indices into a colormap, rather than color values themselves). It is the client's responsibility to set up the colormap appropriately.

Anti-aliasing can be enabled only when Z buffering is off. To enable Z buffering, you can either set the HLHSR mode to `Z-buffer` or set the HLHSR mode to `NONE` and place an HLHSR identifier with a value of "Z-buffer" in a structure.

Mode 2

Lines are 2.5 pixels wide. Anti-aliased lines are blended from the line color to the anti-aliasing background color (color table entry 0).

All pixels of a line are written regardless of existing pixel values.

Anti-aliasing can be enabled only when Z buffering is off. To enable Z buffering, you can either set the HLHSR mode to `Z-buffer` or set the HLHSR mode to `NONE` and place an HLHSR identifier with a value of "Z-buffer" in a structure.

5.6.3 Clipped Objects

Objects located on the near clipping plane in modeling space are sometimes rendered, and sometimes clipped.

5.6.4 Polygons with Nonlinear Vertex Data

Smooth-shaded polygons with nonlinear vertex color and vertex normal data are rendered with inconsistent shading.

5.6.5 Z-Buffering and Edges

When Z-buffering is disabled, edges are not correctly rendered.

5.6.6 Trailing Pixels of Lines

A solid line's trailing pixel is not rendered.

5.6.7 Mapping a Pattern to a Line

Patterns mapped to lines shorter than the pattern itself may cause pixel dropout when the line is rendered.

5.6.8 Graphics Primitive Clipping

Line and polygon vertices that are clipped (either by near or far clipping planes or by viewing volume boundaries) may result in pixel dropout.

5.6.9 Unimplemented PHIGS Primitives

Complex QuadMesh objects and picking of annotation pixmaps are not yet implemented.

5.6.10 Recursive Structures in PHIGS

The current implementation of PHIGS does not check application code for recursive structure networks. Any structure network that references itself is a recursive structure network.

Applications containing recursive structure networks may create self-referencing loops. Executing a self-referencing loop usually results in an error similar to the following:

```
XIO: fatal IO error 32 Broken pipe on X server
```

The error occurs immediately after the application containing the recursive structure calls the execute structure function.

5.6.11 Backfacing Surfaces and NURBS

Under some circumstances, surfaces drawn with NURBS will seem to have edges of backfacing surfaces poking through the frontfacing surfaces. This is caused by the hardware attempting to render polygons that are very small and lie along a silhouette edge. In such cases, it is impossible to compute a surface normal with enough precision to consistently determine whether or not a polygon is frontfacing.

To avoid this problem, use an approximation tolerance that avoids producing very small polygons.

5.6.12 Weighting Control Points for NURBS

Only positive weights are allowed on rational control points. Nonpositive weights will generate a `BadValue` error.

5.6.13 Pixel Dropout in Polygons and NURBS

NURBS and polygons containing lines shorter than one pixel are subject to pixel dropout.

5.6.14 Knot Vectors in a NURBS

Knot vectors for surfaces, curves, and trim curves are subject to the following two restrictions (violations will cause a protocol error):

- Knot multiplicities at either end of the knot vector must not be greater than the order of the curve or surface.
- Knot multiplicities not at the ends of the knot vector must not be greater than the order minus one.

Consider, for example, a curve of order three with five control points. The protocol requires that there be exactly eight knots in the knot vector.

The following vectors violate the first restriction:

[0, 0, 0, 0, 1, 2, 2, 2] (too many instances of "0")
[0, 0, 1, 1, 2, 2, 2, 2] (too many instances of "2")

The following vectors violate the second restriction:

[0, 0, 0, 1, 1, 1, 2, 2] (too many instances of "1")
[0, 1, 2, 2, 2, 2, 3, 4] (too many instances of "2")

5.6.15 Supported Color Approximation Types

The following table lists the color approximation types available using different visual types on 8- and 24-plane DECstation/DECsystem 5000 Model 200 series processors. A legend follows the table.

Visual Type	24-Plane Display	8-Plane Display
DirectColor	CS & CR	n/a
TrueColor	CS & CR-Limited	CS-Flat & CR-Limited
PseudoColor	n/a	CS-Flat & CR-100
StaticColor	n/a	CS-Flat & CR-100 (Predef)
GreyScale	n/a	CS-Flat & CR-100
StaticGrey	n/a	CS-Flat & CR-100 (Predef)

n/a	Visual type is not available with the specified display depth.
CS	ColorSpace is supported with no special caveats.
CS-Flat	ColorSpace with flat shading is supported. Interpolating between colors in the color space results in undefined values.
CR	ColorRange is supported with no special caveats.
CR-Limited	ColorRange is supported in a limited way. A TrueColor visual allows only 3 possible predefined color ranges, one along each color axis. TrueColor will work only if the display depth permits an equal number of samples for each color component. For example, 256 samples for 8/8/8 (a 24-plane display) is acceptable, but 8 and 4 for 3/3/2 (an 8-plane display) is not.
CR-100	ColorRange is supported, but the three multipliers (mults) used to pack the color components into a single 32-bit pixel are forced to (1,0,0) to conform to PHIGS Version 4.0P semantics. In PHIGS Version 5.0, the client is responsible for setting the mults to useful values.
Predef	A reasonable ColorSpace or ColorRange encoding must already exist in a predefined color map.

5.6.16 Using a ColorRange

Due to round-off algorithms, the first and last entries in a multi-entry ColorRange are not consistently generated. For a continuously changing ColorRange, this is not noticeable. For a static ColorRange that has discontinuities at the start or finish of the range, at least one entry at each end should be used to ''pad'' the ColorRange and thus ensure its integrity.

5.6.17 Structure Storage Limit

It is not possible to determine exactly how many structures, structure elements, or both can be created in the PHIGS central structure store. This is because the maximum size of the central structure store is dependent upon the following:

- The memory available to the server process (note that the memory allocated to the server and other processes cannot exceed the swap space for the workstation).
- The type of data which is being stored.

The memory requirements for structures and structure elements are given in the following table. The memory requirements are given in bytes and can be used to obtain an upper bound for the number of structures and structure elements that can be created.

Data Type	Memory Requirement in Bytes
Structure	85
Label element	16 + `sizeof(PHIGSLabelInfo)`
Execute element	32 + `sizeof(PHIGSExecuteStructure)`
Primitive element	32 + 2 times `sizeof(PHIGSXXX)`
Other elements	8 + `sizeof(PHIGSXXX)`

5.7 DEC Realtime Test Integrator

The `ek` device driver subset is not supported on this version of ULTRIX. All other functions are compatible with the ULTRIX operating system.

5.8 DECserver 500

A patch is needed to ensure support of the DECserver 500. For information on obtaining the patch, contact a customer representative.

5.9 DECwrite

The following sections discuss the DECwrite product.

5.9.1 Editing a File with DECwrite

If you edit a file with DECwrite, you may get a dialog box with a message similar to the following:

```
This document's style file, Cannot read message
STYLE_FILE_DIR_ENGLISHwr_man.doc_style, is no longer accessible.
```

Remove the following line from your `.Xdefaults` file:

```
*xnlLanguage:        DPSdecwrite
```

5.9.2 Using DECwrite on Multiple Display Units

DECwrite runs only on screen 0 of multiscreen systems. When trying to run DECwrite on a screen other than screen 0, the DECwrite window is displayed, but nothing is displayed in the graphics region. You are allowed to exit DECwrite in the normal fashion.

5.10 Installing the CDA Converter Library

The CDA converter library is packaged with several layered products. Installing the CDA converter library on ULTRIX and UWS Version 4.2 (or higher) may produce an error stating that `/usr/doc` was not found. This directory exists as part of the unsupported subset, UDXDOC430 (for RISC) or ULXDOC430 (for VAX); if the correct subset is not installed, the directory does not exist. This directory is used by many layered products such as DECwrite to store release notes.

If you have not installed the CDRBASE110 (RISC) or CDABASE110 (VAX) subset for the converter library, execute the following command before installing the

products:

```
# mkdir /usr/doc
```

Ignore any error messages that state the directory exists.

If you have installed the converter library, you can correct the problem by creating the directory as described previously, and then executing the following commands for your processor:

- On RISC processors:

```
# cd /usr/doc
# ln -s ../kits/CDR110/convert_message.mem
```

- On VAX processors:

```
# cd /usr/doc
# ln -s ../kits/CDA110/convert_message.mem
```

Documentation Notes 6

The notes in this chapter apply to ULTRIX operating system, the ULTRIX Worksystem Software, and layered products documentation. For information about new and changed documentation, refer to Chapter 7.

6.1 ULTRIX Documentation

The following notes apply to the ULTRIX documentation set.

6.1.1 Guide to Sharing Software

Note these corrections to the *Guide to Sharing Software on a Local Area Network*:

- Figures 2-1 and 2-2 are no longer accurate. For ULTRIX and UWS Version 4.3, both VAX and RISC servers can serve either VAX or RISC clients. The clients must be Version 4.0 or higher.
- Table 2-1 is outdated. Refer to the Chapter 1 of these release notes for a list of media types and media labels for Version 4.3.
- The RA60 is no longer a distribution device for the product. Please ignore references to it.

6.1.2 The param.c File

The `/param.c` file contains the default values for a number of system parameters. The file is located in the `/sys/conf/vax` directory for VAX machines and the `/sys/conf/mips` directory for RISC machines.

When you build a new kernel, the `config` utility calls in the `param.c` file to provide the values for those parameters. Many of the parameters have equivalent configuration file keywords. If a system parameter is defined in the system's configuration file, then that value overrides the default value in the `param.c` file.

You can edit the `param.c` file to alter a parameter's value, but only when an equivalent configuration file keyword for the parameter does not exist (for example, the `max_nofile` parameter which specifies the maximum number of file descriptors). Whenever possible, alter a parameter's value in the configuration file instead of in the `param.c` file.

6.1.3 POSIX Conformance Document

To establish a strictly-conforming POSIX.1 environment, install the the ULTRIX software without the Enhanced Security Features subset. The Enhanced Security Features subset is ULTSEC430.

If trusted path is configured on your system, break characters entered on any serial line result in the line being disconnected. Trusted path handling supersedes any

special break character handling. This results in unexpected behavior in POSIX environments and on systems that support dial-up lines.

In a POSIX environment, for example, the input mode IGNBRK will not function correctly. This is due to the fact that trusted path handling is performed in response to a break character instead of ignoring the break.

On systems supporting modems, trusted path handling interferes with communications between the system and the modem, as well as with applications using the modem. For example, `uucico` uses the break character in its protocol for communicating between systems; once a break character is received by a system supporting trusted path, the line the `uucico` process is communicating over is disconnected.

To work around these problems, do not configure trusted path on systems that support modems and/or run in a POSIX environment. To remove trusted path from the configuration, follow these steps:

1. Remove the following lines from the system configuration file:

```
options   SYS_TPATH
pseudo-device   sys_tpath
```

2. Rebuild the kernel.
3. Reboot the system using the new kernel.

For steps 2 and 3, refer to the *ULTRIX Guide to Configuration File Maintenance.*

6.1.4 Guide to Software Licensing

There is no reference to the `lmfsetup` utility in the *Guide to Software Licensing.*

The `lmfsetup` utility is a script that allows you to register data supplied by a Product Authorization Key (PAK). The `lmfsetup` script prompts you for the data associated with each of the fields on a PAK. When all the data has been entered, the License Management Facility (LMF) ensures that there are entries against all the mandatory fields, and that the checksum validates all the license data. If the data has been entered correctly, the PAK is registered in the License Database. If the data has been entered incorrectly, the appropriate error message is displayed and you are given the opportunity to reenter the data.

See the `lmfsetup`(8) reference page for more information.

6.1.5 Kernel Messages

There are several panics which can be generated from DECsystem 5900 processors. The following panics can be generated from either file `/sys/machine/mips/kn03.c` or `/machine/mips/kn03.c`:

```
CPU read but timeout
CPU write timeout
DMA overrun
multibit ECC error reported on nonexistent memory module
multibit memory ECC error
multibit memory ECC error in shared page
Unknown bus timeout
Unknown memory error interrupt
```

All of these panics are documented in the *Kernel Messages Reference Manual.* The actions described there apply to the DECsystem 5900 panics.

6.1.6 Software Development

The following sections discuss the software development documentation.

6.1.6.1 Guide to Writing and Porting VMEbus and TURBOchannel Device Drivers

The *Guide to Writing and Porting VMEbus and TURBOchannel Device Drivers* was new with the Version 4.2 release. This manual provides general information on device driver structures, data structures used by device drivers, kernel support routines, and specific information on writing drivers for the VMEbus or TURBOchannel.

The device driver examples listed in this guide are available in source form in the directory, `/usr/examples/devdrivers`.

6.1.6.2 Creating Copies of Sparse Dump Files

A sparse file utility is documented in the *Guide to System Crash Recovery.* This section describes how to create a permanent copy of crash dump files on tape.

To create a permanent copy of the dump files on tape, use the `tar` command to extract the dump files. Compress the dump files before copying them to tape because the `vmcore` file created by savecore is a sparse file, which will expand when you copy it to tape. To compress the dump files, use the `compress` command. To copy the dump files to tape, type the following command sequence:

```
# compress path/vmunix.n path/vmcore.n

# tar c path/vmunix.n.Z path/vmcore.n.Z
```

The path is the directory pathname specified in the `/etc/rc.local` file such as `/usr/adm/crash`. The *n* specifies the number of the crash. Each time a system crash occurs, *n* is incremented by 1. For example, if path is `/usr/adm/crash` and *n* is 1, type the following command sequence:

```
# compress /usr/adm/crash/vmunix.1 /usr/adm/crash/vmcore.1
# tar c /usr/adm/crash/vmunix.1.Z /usr/adm/crash/vmcore.1.Z
```

After you specify the `tar` command, use the `rm` command to remove the dump files and to conserve space on the specified file system. The following example shows how to remove the dump files. In this example the dump files are located in `/usr/adm/crash` and *n* is 1.

```
# rm /usr/adm/crash/vmunix.1.Z /usr/adm/crash/vmcore.1.Z
```

To decompress the dump files when extracting them from a tape, use the `cat`, `uncompress`, and `dd` commands. The `dd` command has an option to create sparse output files. Remember that the `vmcore` file created by `savecore` is a sparse file. To extract and decompress the dump files from tape, type the following command sequence:

```
# tar x
x path/vmcore.1.Z, n bytes, n blocks
# cat path/vmunix.1.Z | uncompress | dd conv=sparse of=path/vmunix.1
# cat path/vmcore.1.Z | uncompress | dd conv=sparse of=path/vmcore.1
# rm path/vmunix.1.Z path/vmcore.1.Z
```

To create a copy of the dump files on another system or in another directory, use the `dd` command to copy the files. Remember that the `vmcore` file created by `savecore` is a sparse file. If you simply copy this file, using the `cp` command for instance, it will expand and use up much file system space. Thus, to reserve file system space, you can copy the sparse files using the `dd` command. To copy the

dump files to another directory using the dd command, log in as root or become superuser and enter the following two commands using this syntax:

dd conv=sparse if=*path*/vmunix.1 of=*newpath*/vmunix.1
dd conv=sparse if=*path*/vmcore.1 of=*newpath*/vmcore.1

The *path* is the pathname to the directory specified in the /etc/rc.local file such as /usr/adm/crash. The *newpath* is the directory pathname where you want to copy the dump files.

To copy the dump files to another system using the dd command, type the following command sequence:

```
# compress path/vmunix.1 path/vmcore.1
# rcp path/vmunix.1.Z path/vmcore.1.Z system:/usr/savecrash
# rlogin system
# cd /usr/savecrash
# cat vmunix.1.Z | uncompress | dd conv=sparse of=vmunix.1
# cat vmcore.1.Z | uncompress | dd conv=sparse of=vmcore.1
# rm vmunix.1.Z vmcore.1.Z
```

6.1.6.3 Correction to the Guide to Backup and Restore

In the *Guide to Backup and Restore,* on page 3-9, the newfs command line is incorrect. The correct command line follows:

```
# /etc/newfs /dev/rrz0g rz55
```

6.1.7 Networking and Communications

This following sections contain documentation notes about networking and communications.

6.1.7.1 Corrections to the Guide to Kerberos

The following changes affect the *Guide to Kerberos:*

- The krb_push script is now available online in the /usr/dss/kerberos/dbase directory. You no longer need to type it in as documented in the *Guide to Kerberos.* The online version supersedes the version documented.
- The *Guide to Kerberos* states, on page 4-13, that the bindsetup command adds the following lines to /etc/rc.local:

```
# %BINDSTART - BIND daemon
[ -f /usr/etc/named ] && {
/usr/etc/named -s -a kerberos one -b /var/dss/namedb/named.boot;
   echo -n ' named' >/dev/console
}
```

This is incorrect. The bindsetup command actually adds the following lines:

```
# %BINDSTART - BIND daemon
[ -f /usr/etc/named ] && {
  /usr/etc/named -n -a kerberos.one -b /var/dss/namedb/named.boot;
   echo -n ' named' >/dev/console
}
```

- In Section 4.4, step 9, the following sentence is incorrect:

The following shows the lines to add to `/etc/rc.local` to start kprop:

The term, **kprop** should be **kropd.**

- In Section 4.4, step 10, the file name in the `touch` command line is `principal.pag` not `principal.dir`.
- In Section 4.6, step 3 in the numbered procedure should be placed after step 5.
- In Section 4.5, the following line is incorrect:

```
/usr/etc/named -s -a kerberos one -b /var/dss/namedb/named.boot;
```

This line should read as follows:

```
/usr/etc/named -n -a kerberos.one -b /var/dss/namedb/named.boot;
```

6.1.7.2 Correction to Root Name Server Reference

The *Guide to the BIND/Hesiod Service* contains an incorrect reference to the host name and address of the root name server.

As of April 1990, the root name server, nic.ddn.mil with IP address 26.0.0.73, is on nic.ddn.mil with an IP address 192.112.36.5.

6.1.7.3 Corrections to the Guide to the Network File System

Section 2.4.7 of the *Guide to the Network File System* describes the pattern matching capability of `automount`; however, the asterisk (*) pattern matching capability is described incorrectly.

Use the asterisk (*) to substitute for lines in indirect maps that are all formatted similarly. The `automount` program interprets the asterisk as a catch-all entry, matching any key (directory name) not previously listed in the map. For example:

```
# indirect map auto.home mounted on /home
#
#key       mount-options  location
smith   -rw, nosuid    little:/usr/staff/&
*       -rw, nosuid    big:/usr/staff/&
```

Here, if a user accesses `/home/jones`, the `automount` daemon will mount `big:/usr/staff/jones`.

Note that the `automount` daemon ignores any entry in an indirect map that follows an asterisk.

6.1.7.4 Guide to the X/Open Transport Interface

The four programming examples that are documented in the *Guide to the X/Open Transport Interface* are available online. These examples are located at `/usr/examples/xti`.

The examples are in the following files:

```
clts_client.c       connectionless client
clts_server.c       connectionless server
cots_client.c       connection client
cots_server.c       connection server
```

6.1.7.5 Documentation for DEMNA XNA Interface

The *Guide to the Data Link Interface* does not include the DEMNA XNA interface in the list of supported hardware.

6.1.7.6 Corrections to the Guide to Preparing Software for Distribution on ULTRIX Systems and the kitcap(5) Reference Page

There is incorrect syntax for `/etc/kitcap` file entries and for `gentapes` and `genra` command lines in the `kitcap` reference page and in the *Guide to Preparing Software for Distribution on ULTRIX Systems*.

The corrections follow.

6.1.7.6.1 Section 5.8, Building /etc/kitcap – The syntax and examples in Section 5.8 are incorrect. The correct syntax and examples follow.

The format of an `/etc/kitcap` entry for tape media is as follows:

```
<product code><media code> | [product description]:<directory1>[:directory2]:\
SPACE:SPACE:SPACE:INSTCRTL:<subset1>[:<subset2>]
```

The subsets must be listed so that any subset on which other subsets depend is listed before its dependent subsets.

The example that follows shows an entry for TK50 tape media is as follows:

```
UWS400TK | ULTRIX Worksystem Software:/sys/dist/:\
SPACE:SPACE:SPACE::INSTCTRL:UWSXRT400:UWSMH400
```

The format of an `/etc/kitcap` entry for disk media is:

```
<product_code><media_code>:partition:dd=<destination_directory>:\
[product_description]:<directory1>[:directory2]:instctrl:<subset1>[:<subset2>]
```

The example that follows shows an entry for RA60 disk media:

```
UWS400RA:c:dd=product:ULTRIX_Worksystem_Software:/sys/dist/:\
instctrl:UWSXRT400:UWSMH400
```

The underscore character (_) is required to connect words in a product description for disk media. The subsets must be listed so that any subset on which other subsets depend is listed before its dependent subsets.

6.1.7.6.2 Section 5.9.1, Making Tape Media – The syntax and example in Section 5.9.1 are incorrect. The correct syntax and examples follow.

Use the `gentapes` utility to make tape media. The command line syntax is as follows:

gentapes [**-wv**] [*hostname:<product_code> <special>*]

The `-w` option indicates write-only; the `-v` option indicates verify only. If neither option is specified, the utility writes, rewinds the tape, then verifies.

If you specify a node, the `gentapes` utility looks for the output directory on the

node you specify. For example:

```
# gentapes mysystem:UWS400 /dev/nrmt0h
```

You can use the Network File System (NFS) to remotely mount the kit on a machine with the correct drive.

The `gentapes` utility appends either TK or MT to the *product_code* after finding the entry in the `/etc/kitcap` file. For example, if you type the following command and the `/etc/kitcap` entry specifies TK50 tape, the `gentapes` utility appends TK to UWS400:

```
# gentapes UWS400 /dev/nrmt0h
```

6.1.7.6.3 Section 5.9.2, Making RA60 Disk Media – The syntax and example in Section 5.9.2 are incorrect. The correct syntax and an example follow.

Use the `genra` utility to make RA60 disk media. The command line syntax is as follows:

genra [**-wv**] [*hostname:<product_code> <special>*]

The following example specifies a `hostname`:

```
# genra mysystem:UWS400 /mnt
```

The `genra` utility appends RA to the *product_code* automatically after finding the entry in the `/etc/kitcap` file.

6.1.8 Security

The following sections discuss the security documentation.

6.1.8.1 Controlling Network Access to Workstation Displays

Chapter 6 of the *Security Guide for Users* incorrectly states that if there are differences between the system access control list and a workstation access control list, the system access control list prevails. This is not true.

At system startup, the X server initializes the server access control list by reading the `/etc/X*.hosts` file. This privileged file names the hosts on a network that can access a workstation display.

When the Session Manager, `dxsession`, is started, it updates this server access control list to match the session access control list. The session access control list is the list of hosts that users specify using the Security... option from the Customize menu in the Session Manager window. The Session Manager stores this list of hosts in the `.Xdefaults` file in the user's home directory, using the resources `sm.host_list` and `sm.num_hosts`.

The session access control list, if it exists, overrides the server access control list. For example, if the server access control list includes hosts `orion` and `myrtle`, and the session access control list includes only host `myrtle`, the Session Manager requests that the server remove `orion` from the server access control list, and only authorized users on `myrtle` can access the workstation.

If a user does not add a host to the session access control list using the Security... option from the Customize menu, or if the user does not save the changes made during the current session, the Session Manager does not create a list of entries for

the sm.host_list and sm.num_hosts resources in the .Xdefaults file. Thus, no session access control list exists. If no session access control list exists, the only hosts allowed access to the workstation display are those listed in the server access control list before the session is initiated. These are the hosts listed in the /etc/X*.hosts file.

6.1.9 Reference Pages

The following sections discuss the ULTRIX reference pages.

6.1.9.1 Reference Pages Available Only Online

For ULTRIX and UWS Version 4.3 the new and changed reference pages are available only online.

If you need a hardcopy of any of the reference pages, you can process the source file for the reference page and print the formatted page.

The source files for all the reference pages are stored in subdirectories of the /usr/man directory after you install your ULTRIX system. For example, the /usr/man/man4 directory contains source files for Section 4 reference pages. Each source file in that directory contains one reference page. The names of the source files indicate which reference page they contain. For example, the rz(4) reference page is stored in the rz.4 file. The SCSI(4) reference page is stored in the scsi.4 file, and so on.

To process a reference page source file, use the man command. You can print the output from the man command on a line printer. For example, to process the rz.4 reference page, enter the following command:

```
% man 4 rz | lpr &
```

6.1.9.2 The cp Command

The cp command with the -p option does not work as stated in the reference pages. The command with the -p option preserves only the modification time, access time, and file mode. The user id (UID) and group id (GID) are not preserved. If necessary, the UID and GID can be updated by root.

6.1.9.3 The cut Command

The cut(1) reference page should include the following diagnostics.

A value of 1 is returned if the cut completed successfully, A value of 2 indicates an error was returned. The errors are:

```
cut: cannot use -c in combination with any other option
```

Meaning: The -c option cannot be specified on the same command line as the -s, -f, or -d options.

```
cut: write error: <reason>
cut: close error: <reason>
```

Meaning: An error occurred during redirection which caused the file to not be written or closed properly. An example of this is when the write occurs over an NFS

file system and the error is not detected until the file is closed.

```
cut: cannot open <file>: <reason>
```

Meaning: The input file could not be open for the reason given.

6.1.9.4 The dd Command

The following notes apply to the dd(1) reference page.

On SCSI tape devices when reading a multivolume tape set, the dd command will exit normally upon hitting EOT on the first (and subsequent) volume(s) as though this were a single volume tape set. The user receives no indication of the truncation. Where the user is aware that a multivolume tape set is being read, the user must use the mv=n option to specify the number of volumes in order to permit dd to read the next volume(s).

On non-SCSI tape devices, if data is written beyond the point where the device warns of End-of-Media (for example, foil on 1/2" tapes), dd assumes that there is another volume of data and prompts accordingly, unless the user has specified the number of volumes with the mv=n option. In this case, the user restricts the number of volumes read by the command.

Note

Unlike tar and dump, the dd command does not insert file identification headers or trailers when writing to tape. If you use dd repeatedly to write more than one file to a non-rewinding tape then upon reading the tape, the last file on the media will be assumed to continue to the next volume (if there is another). This allows the command to work correctly when data is piped between it and the tar or dump commands.

Also, unlike tar and dump commands, dd doesn't read or write beyond the device End-of-Media warning. Therefore, the command can only claim to read correctly to the end of tapes that have been written directly by dd itself.

6.1.9.5 The printcap(5) Reference Page

The defaults for PostScript printers contains a mistake. The following entry is not correct:

```
Nu      num     not set         Number up               -N
```

The following is the correct entry:

```
Nu      str     not set         Number up               -N
```

6.1.9.6 The pty(4) Reference Page

If you open the slave side of the pty before opening the master side, the open will fail and the error EIO will be returned.

6.1.9.7 The tar Command

During an extract, the `tar` command uses the original permissions when creating directories. This can create a problem when the original directory does not have write permission at the time it was archived. If `tar` extracts a directory that does not have write permission, it will be unable to create files under that directory during the extraction.

In running the X/Open Verification Suite Release 3.203 in POSIX mode, the following exceptions were found in the `tar` command:

- Prefix usage and file names of 100 to 256 characters
- Permissions
- Multiple volumes

Note that the `pxtar` command is fully XPG3 compliant.

6.1.9.7.1 Prefix Usage and File Names of 100 to 256 Characters – According to Section 10.1.1 of the IEEE Standard Portable Operating System Interface for Computer Environments (New York, NY: The Institute of Electrical and Electronics Engineers, Inc, 1988), "The *name* and the *prefix* fields produce the pathname of the file. The hierarchical relationship of the file is retained by specifying the pathname as a path prefix, a slash character and filename as the suffix. If the *prefix* contains non-null characters, *prefix*, a slash character, and *name* are concatenated without modification or addition of new characters to produce a new pathname."

The calculation used by ULTRIX `tar` is *prefix* and *name* concatenated; ULTRIX does not use the slash (/). In ULTRIX, if the *prefix* is null, the pathname is *name* as in POSIX.1.

The *prefix* field has a backward overflow for file names greater than 100 characters. The calculation used when an overflow occurs is as follows:

```
filename: /a01/a02/a03/ ... /a99
```

POSIX:

```
name:   a74/a75/ ... /a99
prefix: /a01/a02/ ... /a73
```

ULTRIX:

```
name:   a99
prefix: /a01/a02/ ... /a98/
```

To work around this problem, use file names with fewer than 100 characters or use the `pxtar` command.

6.1.9.7.2 Permissions – According to Section 10.1.1 of IEEE Std 1003.1-1988, a process with appropriate privileges restores the ownership and permissions exactly as recorded on the medium, except that symbolic user and group IDs are used for the `tar` format. If only the uppercase `-P` option to the `tar` command is used, ULTRIX does not restore permissions as they were preserved on the media. The lowercase `-p` option to the `tar` command allows the modes to be preserved, and also allows non-permitting processes to preserve the modes.

Barring the restrictions previously noted, you can use either `tar` with both the uppercase `-P` and lowercase `-p` options, `tar -Pp` , or `pxtar` to ensure that permissions are preserved.

6.1.9.7.3 Multiple Volumes – The `tar` command does not conform to Section 10.1.3 of IEEE Std 1003.1-1988 in regard to multiple volumes. The `tar` command supports multiple volumes in non-POSIX mode with no restrictions. However, when used in POSIX mode, any file that spans two media is corrupted. All other files are preserved.

To work around the problem, insure that all files are contained fully in a single medium or use the `pxtar` command.

6.2 ULTRIX Worksystem Software Documentation

The following sections discuss the ULTRIX Worksystem Software documentation.

6.2.1 Addition to DECwindows User's Guide

In Chapter 3, Keyboard Type on page 3-8, add the following to the second paragraph:

For example, if your keyboard model is LK401 AG and you are using an Austrian German keyboard, select the Austrian German LK401 AG keyboard type in the Session Manager's Customize Keyboard menu. This selection also applies if your keyboard model is an LK501 AG, as the LK401 AG and LK501 keyboard types are the same.

6.2.2 The DECwindows Toolkit Documentation

The following sections provide corrections to errors made in the DECwindows Toolkit documentation.

6.2.2.1 DwtGetNextSegment Function

The parameter `text_return` is incorrectly stated as being of type `char *`. The correct type is type `char **`.

6.2.2.2 Invoking the UIL Compiler

The *ULTRIX Worksystem Software Guide to the XUI User Interface Language Compiler* states that the uil compiler is invoked with the `uil` command. In fact, the command to invoke the uil compiler is `dxuil`.

6.2.3 UWS Reference Pages

The following sections highlight changes to the reference pages.

6.2.3.1 X Window System, Release 4

This release provides reference pages from MIT which document the following X Window System components:

- Xlib routines (Section 3X11)
- X Toolkit routines (Section 3Xt)
- X clients (available in the unsupported subset)

6.2.3.2 X Server Reference Pages

The X Server reference pages in Section 8X are as follows:

- `Xws`(8X) - RISC version for DECstation/DECsystem 3100 single screen monochrome and color processors, DECstation/DECsystem 5000 Model 200CX single screen monochrome and color processors, and DECstation/DECsystem 5000 Model 200CX multiscreen monochrome and color processors. In additon, `Xws` supports the Personal DECstation 5000 series and the DS3000 Model 200 with MX, HX, PX, or CX options.
- `Xqvsm`(8X) - VAX version with cross-references from `Xqdsg`.
- `Xwst3d`(8X) - RISC version for DECstation/DECsystem 5000 Model 200 PXG+ and PXG Turbo + processors.
- `Xgb`(8X) - VAX X server.

6.2.3.3 Xgb Reference Page

The `Xgb` reference page was inadvertently removed from the reference page inventory. If you need reference information on the `Xgb` server, see the ULTRIX/UWS Release 4.1 `Xqvsm` reference page.

6.2.3.4 XSetLineAttribute Reference Page

The `XSetLineAttributes` function must be referenced by the `XSetLineAttribute` reference page.

6.2.3.5 dxmail Reference Page

The `dxmail` Print option attempts to print to a printer named `lp` if no other printer has been defined. If no printer with the name `lp` exists, no print output results. One way to specify a printer for `dxmail` is to include a PrintCommand resource in the `.Xdefaults` file, as in the following example, which assumes that a printer of *printername* exists:

```
Mail*PrintCommand:      lpr -Pprintername >/dev/null 2>/dev/null
```

New Features and Changes 7

The following sections discuss changes and new features for the ULTRIX and UWS software and documentation components.

7.1 ULTRIX New Features and Changes

The following sections discuss changes and new features for the ULTRIX components.

7.1.1 System Upgrade Added to Installation Options

A new mechanism for installing ULTRIX and UWS allows owners of Version 4.2 (or higher) systems to upgrade to Version 4.3 without having to destroy the existing environment and system customizations.

This upgrade procedure is detailed in the *Guide to Installing ULTRIX*.

7.1.2 ULTRIX/SQL License No Longer Available

Beginning with ULTRIX Version 4.3, the license for ULTRIX/SQL is no longer available on ULTRIX systems. Existing customers continue to be supported under terms of their contracts.

7.1.3 DECthreads Component

The DECthreads component has been added to the product. This `pthread` interface library is based on Draft 4 of the IEEE standard for multithreaded programming, POSIX 1003.4a.

Documentation consists of the *Guide to DECthreads* manual and online reference pages for the `3thr` library routines. For more information, refer to the online `intro`(3thr) reference page. The routines that have been incorporated are described in these online reference pages:

```
atfork(3thr)
intro(3thr)
pthread_attr_create(3thr)
pthread_attr_delete(3thr)
pthread_attr_getinheritsched(3thr)
pthread_attr_getprio(3thr)
pthread_attr_getsched(3thr)
pthread_attr_getstacksize(3thr)
pthread_attr_setinheritsched(3thr)
pthread_attr_setprio(3th)
pthread_attr_setsched(3thr)
pthread_attr_setstacksize(3thr)
pthread_cancel(3thr)
pthread_cleanup_pop(3thr)
pthread_cleanup_push(3thr)
```

```
pthread_cond_broadcast(3thr)
pthread_cond_destroy(3thr)
pthread_cond_init(3thr)
pthread_cond_signal(3thr)
pthread_cond_timedwait(3thr)
pthread_cond_wait(3thr)
pthread_condattr_create(3thr)
pthread_condattr_delete(3thr)
pthread_create(3thr)
pthread_delay_np(3thr)
pthread_detach(3thr)
pthread_exit(3thr)
pthread_get_expiration_np(3thr)
pthread_getprio(3thr)
pthread_getscheduler(3thr)
pthread_getspecific(3thr)
pthread_join(3thr)
pthread_keycreate(3thr)
pthread_lock_global_np(3thr)
pthread_mutex_destroy(3thr)
pthread_mutex_init(3thr)
pthread_mutex_lock(3thr)
pthread_mutex_trylock(3thr)
pthread_mutex_unlock(3thr)
pthread_mutexattr_create(3thr)
pthread_mutexattr_delete(3thr)
pthread_mutexattr_getkind_np(3thr)
pthread_mutexattr_setkind_np(3thr)
pthread_once(3thr)
pthread_self(3thr)
pthread_setasynccancel(3thr)
pthread_setcancel(3thr)
pthread_setprio(3thr)
pthread_setscheduler(3thr)
pthread_setspecific(3thr)
pthread_testcancel(3thr)
pthread_unlock_global_np(3thr)
pthread_yield(3thr)
```

7.1.4 Audit Data Recovery

The `secsetup` command is used to configure security features. The command modifies the `/etc/savecore` statement in the `/etc/rc.local` file.

The `savecore` command has a new switch, `-a`, which allows the audit data still in kernel buffers to be saved along with a system core dump. Previously, there was no way to recover audit data remaining in kernel buffers.

The audit data is stored in a fashion similar to the corefiles and can be analyzed with the `audit_tool` command.

Audit data is recovered only from kernel buffers. By default, audit data is not deleted from the kernel buffers until after the audit daemon has read and written that data out to either the file system or the network. Thus, data held by the audit daemon at the time of a system crash can be recovered from the kernel buffers.

You can increase the size of the buffer used by the audit daemon, but you do so at the expense of having all data duplicated in the kernel buffers.

Note that it is possible for the recovered data to not align on an audit record boundary, so the first record may appear corrupted. Note that it is also possible to have data duplicated in both the system `auditlog` and the recovered audit data file.

7.1.5 Modified Version of DEC C++ Linker on Kit

Although the new Version (2.0) of DEC C++ is not yet supported on ULTRIX, this kit includes modified versions of the `ld` and `*crt0*` files in a separate directory. (These files are duplicates of files found in the DEC C++ kit.) These files enable users to link C code with DEC C++ code from third-party vendors, without installing the DEC C++ kit. These files are only of interest to such users; other users will continue to use the regular `ld` and files. The C++ files are located in the `/usr/lib/cmplrs/cxx` directory.

7.1.6 The named(8) Command

The `named` daemon has a new option, `-nh`.

This option runs `named` in network safe Hesiod mode. All HS class queries sent over the network are authenticated. Nonauthenticated queries for HS information from hosts other than local hosts are not answered. Zone transfers of non-HS information are not authenticated.

7.1.7 The ndbm(3) Routine

The `ndbm` routine has a new function, `dbm_setpblksiz`. The function sets the block size for large databases.

7.1.8 VAX Math Library Performance

The sine, cosine, and arctangent routines in the VAX math library used polynomial instructions, which some VAX processors emulate (and are thereby slow). These instructions have been replaced by a series of multiply and add instructions. As a result, most VAX processors realize significant performance gains in these routines, while a few VAX processors realize slight negative effects. The accuracy of the routines is not greatly affected.

7.1.9 CursesX Library

A new module and a new header file have been added to the CursesX interface:

- `_ctoc.c` module
- `extraflg.h` header file

7.1.10 sendmail

Version 5.65 of sendmail is now included in the product. Included in Version 5.65 is `sendmail` support for international message files. In addition, the `sendmail.cf` file has been modified to be compatible with Version 5.65.

7.1.11 Device Drivers

Third-party device drivers can now write to mass storage devices of greater than two gigabytes capacity.

7.1.12 I/O Performance Enhancements

The I/O enhancements improve UFS (local) and NFS file system performance. You must enable the UFS enhancements to obtain the increase in file system performance.

UFS sequential I/O performance is improved as follows:

- Reads by 20 to 300 percent
- Writes by 100 to 300 percent

NFS server performance is improved as follows:

- Write throughput up to 100 percent
- With Prestoserve running, up to 100 percent
- Reduces CPU overhead due to write I/O

While you do not have to change any applications, there are some system configuration changes that you may need to make to obtain the optimal performance benefits.

You should make sure that the following configuration parameters are set:

- In the `param.c` file:

  `delay_wbuffers` is set to zero (0).

  `cache_bufcache` is set to 1.

- In the configuration file:

  `bufcache` is set to a percentage of physical memory, usually 8 mb or more on a server.

If the previous parameters are not set as specified, you may not obtain the optimal performance benefits. To modify your configuration parameters, perform the following steps:

1. Make the `param.c` file changes, if necessary.
2. Make the configuration file changes, if necessary.
3. Rebuild the kernel.
4. Copy the new kernel to the `root (/)` file system.
5. Reboot the system.

7.1.12.1 UFS I/O Enhancements

The UFS I/O enhancements are based on the following:

- UFS block clustering causes the file system buffer cache to combine small I/O operations into larger disk operations. For example, file system blocks are combined into larger write requests, and multiple blocks are read in advance of single read requests.
- Performance is based on hardware speed, which depends on the disk and controller. For sequential access of large files, the read/write speed is up to 95 percent or more of the raw disk subsystem performance. Read/write requests are reduced by an average of 50 percent.

To enable the UFS I/O enhancements, some file system parameters must be set to specific values that are determined by your processor, controller, and disk types. You can obtain the parameter values that will result in the best increase in performance by writing a file that is more than 1 megabyte in size to the file system, unmounting the file system, mounting the file system, and then reading the file while you vary the parameter combinations. The fastest time for reading and writing the file determines the values you should use.

Use the `dumpfs` command to display file system information. Use either the `tunefs` command or the `newfs` command to set the file system parameters.

The following file system parameters can be modified:

- Rotational delay between contiguous blocks (*`rotdelay`*)

  If you change the value for the rotational delay between contiguous blocks, you may have to pick up and then put down the files to obtain the performance benefits. New files will then be laid out to get the best performance benefits.

- Maximum number of contiguous blocks (*`maxcontig`*)

  The value for the maximum contiguous blocks can be in the range from 1 to 8. This value determines the maximum number of file system blocks to combine into a cluster. If you specify 8 for *`maxcontig`* on an 8 kilobyte/1 kilobyte file system, the system will attempt to combine reads and writes into 64 kilobyte requests. If you specify 1 for *`maxcontig`*, the system does not attempt to combine any reads and writes. The maximum cluster size is 64 kilobytes, so specifying a value greater than 8 for *`maxcontig`* has no effect on performance.

- Maximum number of blocks per file in a cylinder group (*`maxbpg`*)

  You can specify the value that is specified in the `bpg` field obtained from the `dumpfs` command output.

For example, to change the rotational delay of a file system to zero, first umount the file system and then use the following command syntax:

tunefs -d 0 [*char device*]

To change the number of blocks that make up a cluster to 8, umount the file system and then use the following command syntax:

tunefs -a 8 [*char device*]

The maximum cluster size is currently 64 kilobytes (with maximum contiguous blocks equal to 8 for an 8 kilobyte/1 kilobyte file system)

To set the maximum blocks per file in a cylinder group, use the following command syntax:

tunefs -e *number* [*char device*]

For the *`number`* variable, specify the value of the `bpg` field obtained from the `dumpfs` command output.

If the original rotational delay was not set to zero, you must lay out the files on the disk again to obtain the optimal performance benefits. You can do this by using the `newfs` command and then copying the files to the new file system. You must be in single-user mode to lay out the files.

The following steps provide an example of how to lay out files:

1. Create a new file system. For example:

   ```
   # newfs /dev/rrz1c
   ```

2. Tune the new file system to set the rotational delay to zero, the maximum number of logically contiguous blocks to 8, and the maximum blocks per file in a cylinder group to 1024. For example:

   ```
   # tunefs -d 0 -a 8 -e 1024 /dev/rrz1c
   ```

   You may want to specify command line values that are different from these values.

3. Unmount the original file system whose files you want to lay out again. For example:

   ```
   # umount /usr/users
   ```

4. Mount the new file system on the `/mnt` mount point. For example:

   ```
   # mount /dev/rrz1c /mnt
   ```

5. Use the `dump` command to copy the files from the original file system to the new file system. For example:

   ```
   # dump 0f - /dev/rrz3c | (cd /mnt; restore xf -)
   ```

6. Unmount the new file system. For example:

   ```
   # umount /mnt
   ```

7. Edit the `/etc/fstab` file to replace the original file system specification with the new file system specification.
8. Mount the new file system on the original mount point. For example:

   ```
   # mount /dev/rrz1c /usr/users
   ```

The following tuning information may be needed to use the UFS enhancements. The default values usually are adequate in most situations, and changing the values will result in a change in performance.

- UFS clustering combines small reads and writes into larger cluster operations. Each cluster I/O operation has a cluster buffer associated with it. The system attempts to allocate 100 cluster buffers at startup time. To display the number of cluster buffers that are allocated on your system, use the following command syntax if you have a VAX machine:

  adb -k /vmunix *ncluster_buf$d*

  Use the following command syntax if you have a RISC machine:

  kvar -rl -s *ncluster_buf* **/vmunix**

  If you have less than 100 cluster buffers, you may want to increase the `physmem` value specified in the configuration file. If you make changes to the configuration file, you must rebuild the kernel.

- Logically contiguous sequential writes (data portion) to file systems with a *maxcontig* value greater than 1 are combined into clusters whose size is obtained from the following calculation:

  *fs_bsize * maxcontig*

  Many writes of this size can be outstanding to the disk.

- Logically contiguous read-ahead blocks are combined into clusters. The size of a cluster gradually increases from 2 file system blocks to the size limit obtained from the following calculation:

  *fs_bsize * maxcontig*

  When the size of the read-ahead cluster reaches the size limit, multiple clusters of read-ahead data are scheduled until a maximum number of *cluster_max_read_ahead* clusters are scheduled.

  The value of the *cluster_max_read_ahead* variable is specified with the `adb` command (for a VAX system) or the `kvar` command (for a RISC system) and must be in the range from 1 to *ncluster_buf*. The value 8 is the default. The value of the *cluster_max_read_ahead* variable determines the maximum number of clusters of read-ahead data the kernel will schedule per file. The kernel determines the amount of read-ahead data to schedule by tracking the number of sequential read requests. The more sequential read requests are made to a file, the larger the read-ahead request.

  For example, an 8 kilobyte/1 kilobyte file system with a *maxcontig* value of 8 will attempt to schedule read-ahead data in 8 kilobyte increments until it reaches 64 kilobytes. Then, the file system will schedule multiple 64 kilobyte clusters of read-ahead until *cluster_max_read_ahead* clusters are scheduled.

- The new `crash` utility `wg` command provides NFS write gathering and UFS clustering statistics.

7.1.12.2 NFS Enhancements

The following NFS I/O enhancements are provided:

- NFS write gathering is available with no violation of NFS protocol.

  Multiple write requests to a file that already exists on the server are combined into fewer disk operations. Replies are not sent until all data and associated metadata are safely stored on the disk.

- Depending on your server load, you may be able to run fewer `nfsd` daemons without decreasing the server performance.

- Clients can run more than 4 `biod` daemons to obtain better write performance, but depending on your network or server conditions, running more than 4 `biod` daemons can result in a net performance loss.

7.1.13 ULTRIX SCSI/CAM Architecture Driver Component

For RISC customers, a SCSI/CAM component has been added as a bundled component. This component is included with Version 4.3 on its own distribution media and with its own documentation.

Install the component after you have installed or upgraded to Version 4.3.

The documentation for the SCSI/CAM component consists of these manuals:

- *ULTRIX SCSI/CAM Architecture Release Notes and Installation Guide*
- *ULTRIX SCSI/CAM Architecture Guide to Writing Device Drivers for the ULTRIX SCSI/CAM Architecture Interfaces*

7.2 ULTRIX Worksystem Software New Features and Changes

The following sections discuss changes and new features for the ULTRIX Worksystem Software (UWS) components.

7.2.1 OSF/Motif Version 1.1.3

For workstation customers, both VAX and RISC, the Version 1.1.3 release of OSF/Motif has been added as a bundled component. This component is included with Version 4.3 on its own distribution media and with its own documentation.

Install the component after you have installed or upgraded to Version 4.3.

The documentation for the OSF/Motif component consists of these manuals:

- *ULTRIX DECwindows for OSF/Motif Version 1.1.3 Release Notes and Installation Instructions*
  Order Number AA-PPJ2B-TE
- *OSF/Motif User's Guide*
  Version 1.0, Order Number AA-PC7QA-TE
- *OSF/Motif Programmer's Guide*
  Version 1.1, Order Number AA-PC7MB-TE
- *OSF/Motif Programmer's Reference Manual*
  Version 1.1, Order Number AA-PC7NB-TE
- *OSF/Motif Style Guide*
  Version 1.1, Order Number AA-PC7PB-TE
- *Application Environment Specification (AES) User Environment Volume, Rev.B*
  Order Number AA-PC7RB-TE
- *DECwindows Companion to the OSF/Motif Style Guide*
  Order Number AA-PGZ9A-TE
- *DECwindows Motif Guide to Application Programming*
  Order Number AA-PJN6A-TE
- *DECwindows Extensions to Motif, UNIX Systems Edition*
  Order Number AA-PJN7A-TE

In addition, the OSF/Motif component has online reference pages and help files for its applications.

7.2.2 X Authorization Protocol and Workstation Security

The X protocol has a potential vulnerability: a client that has access to a workstation display can view all of the input events going to any window on that display.

One way to prevent this potential vulnerability is to use DECterm's Secure Keyboard feature when entering passwords and other sensitive information. When you enable Secure Keyboard, you prevent any other client from receiving DECterm's input events. See `dxterm`(1X) for more information.

Another way to prevent this potential vulnerability is to restrict access to your workstation. The X Window System Version 11 Release 4 (X11R4) provides two mechanisms for restricting access to a display:

- The server's host access control list: This mechanism, available through the Session Manager's Customize Security window, restricts access to certain hosts. All clients on those hosts can access your workstation; all others cannot access your workstation.
- The MIT-MAGIC-COOKIE-1 authorization scheme: This mechanism requires that a client supply a password to the server before it can access the display, thereby restricting access to your workstation to those users who have the password.

The following sections discuss MIT-MAGIC-COOKIE-1 support in ULTRIX and UWS Version 4.3.

7.2.2.1 MIT-MAGIC-COOKIE-1 Support

ULTRIX and UWS support the MIT-MAGIC-COOKIE-1 authorization scheme through the `-auth` command line option.

In ULTRIX and UWS Version 4.3, support for the MIT-MAGIC-COOKIE-1 authorization scheme has been added to Digital's Xlib. All of the applications included in Version 4.3 have been linked against this version of Xlib and so include this support. Any Digital layered products that link against Xlib during their installation will also pick up this support. Any layered products that are supplied as prelinked images may not have this support yet and so cannot be used to display to servers with MIT-MAGIC-COOKIE-1 authorization turned on. The same is true for products from vendors other than Digital.

To incorporate the MIT-MAGIC-COOKIE-1 support into your applications, relink them against the Version 4.3 Xlib.

7.2.2.2 How to Use the MIT-MAGIC-COOKIE-1 Authorization Scheme

You can turn on MIT-MAGIC-COOKIE-1 authorization by using the X Display Manager, `xdm`, to manage logins on your workstation in place of the `Xprompter` utility. You can find `xdm` in the Unsupported X11R4 Components subset (UDXUNMIT430 for RISC, ULXUNMIT430 for VAX). To use `xdm`, comment out the line that begins with :0 your `/etc/ttys` file, and add the `xdm` command to the end of your `/etc/rc.local` file. To turn on MIT-MAGIC-COOKIE-1 authorization in `xdm`, set the DisplayManager*authorize resource to true. Each time you log in, `xdm` will automatically generate a new password, give it to the server, and put it in your authority file (`~/.Xauthority`). See `xdm`(1X) for more information.

If you do not wish to use `xdm`, you can turn on MIT-MAGIC-COOKIE-1 authorization as follows:

- Use the X authority file, `xauth` in the Unsupported X11R4 Components subset) to create your authority file (`~/.Xauthority`) and add an authorization entry

for your display to it. For example:

```
% /usr/bin/X11/xauth add mynode:0 . 12345678909876543210
```

- Copy your authority file to the root directory and set the protection so that only root can access the file. For example:

  ```
  % su# cp ~myname/.Xauthority /# chmod 600 /.Xauthority
  ```

- Edit the `/etc/ttys` file to include the following option on the server command line in the line beginning with :0:

  ```
  -auth /.Xauthority
  ```

  For example:

  ```
  :0 "/usr/bin/login -P /usr/bin/Xprompter -C
  /usr/bin/dxsession -e" none on secure
  window="/usr/bin/Xws bc -auth /.Xauthority"
  ```

Whether you are using `xdm` or not, you can use `xauth` to display your authority file or to extract the password to give to another user. See `xauth`(1X) for more information.

When using MIT-MAGIC-COOKIE-1 authorization, clients on any host listed in the Session Manager's Customize Security window (the server's host access control list) can access your workstation even if they do not have the correct password. Therefore, you should remove all hosts from the Session Manager's Customize Security window when using MIT-MAGIC-COOKIE-1 authorization.

7.2.3 New Xws Server Memory Allocation Algorithm

The memory allocation algorithm used previously in the `Xws` server favored clients that created multiple pixmaps of the same size. If a client created large pixmaps of different sizes, large amounts of memory could be preallocated by the server but never used.

In Version 4.3, the `Xws` server uses a new memory allocation algorithm which avoids this problem.

7.2.4 ULTRIX Compatibility Kit No Longer Supported

In ULTRIX and UWS Version 4.2A, a separately-installed Compatibility Kit component was shipped on separate media, which customers ordered through their Digital representative. This kit is not supported in ULTRIX and UWS Version 4.3 and no longer ships with the product.

7.2.5 Bookreader Version 3.1

A new version of the Bookreader, Version 3.1, ships with the product. Bookreader Version 3.1 includes numerous bug fixes and enhancements to the support for the DECwindows Motif Help System (informally known as "HyperHelp").

7.3 New and Changed Documentation

The SCSI/CAM and OSF/Motif documentation listed in the previous sections now ships with the product. In addition, the following documentation has changed:

- The *Release Notes* have been updated with information about Version 4.3, while retaining the information still valid from previous releases.
- The *Guide to Installing ULTRIX* has been revised to describe how to install and to upgrade to Version 4.3.

No other printed documents have been changed.

The following Section 1 and Section 8 online reference pages have changed:

```
ar(1) RISC
cc(1) VAX
cc(1) RISC
chfn(1)
chsh(1)
crash(8)
dbx(1) VAX
dbx(1) RISC
dd(1)
ex(1)
expr(1)
grep(1) VAX
grep(1) RISC
ksh(1)
make(1)
pstat(8)
sh(1)
size(1) VAX
size(1) RISC
talk(1)
vcc(1)
vi(1)
```

The following Section 3 and Section 4 online reference pages have changed:

```
curses(3x)
execl(3)
nl_scanf(3int)
nbuf(4)
ndbm(3)
pty(4)
rz(4)
scanf(3int)
scanf(3s)
tz(4)
```

7.4 ULTRIX and UWS Printed Documentation

The following two tables show the titles and order numbers for the documentation shipping with the ULTRIX and UWS products in Version 4.3.

Table 7-1: ULTRIX Version 4.3 Manuals and Order Numbers

Title	Order Number
Application Environment Specification (AES) User Environment Volume, Rev.B [a]	AA-PC7RB-TE
Application Sources Mailer	AV-LY29B-TE
The Big Gray Book: The Next Step with ULTRIX	AA-PBKNA-TE
CDA Reference Manual Volume 1	AI-PBD3A-TE

Table 7-1: (continued)

Title	Order Number
CDA Reference Manual Volume 2	AI-PC02A-TE
DEC ULTRIX to DEC OSF/1 Migration [a]	AA-PJUAA-TE
DECrpc Programming Guide	AA-PBKYA-TE
DECwindows Companion to the OSF/Motif Style Guide [a]	AA-PGZ9A-TE
DECwindows Motif Guide to Application Programming [a]	AA-PJN6A-TE
DECwindows Extensions to Motif, UNIX Systems Edition [a]	AA-PJN7A-TE
Guide to Backup and Restore	AA-ME92B-TE
Guide to the BIND/Hesiod Service	AA-LY21B-TE
Guide to Configuration File Maintenance	AA-ME90D-TE
Guide to Creating Compound Documents with the CDA Toolkit	AA-PBOKA-TK
Guide to the Data Link Interface	AA-PBKZA-TE
Guide to DECthreads [a]	AA-PJNEA-TK
Guide to Developing International Software	AA-LY26B-TE
Guide to Disk Maintenance	AA-ME93C-TE
Guide to the Error Logger	AA-ME95B-TE
Guide to Ethernet Communications Servers	AA-ME98B-TE
Guide to IBM Terminal Emulation for VAX Processors	AA-MF04B-TE
Guide to Installing ULTRIX [a]	AA-PBL0E-TE
Guide to Kerberos	AA-PBKVA-TE
Guide to Languages and Programming	AA-ML94C-TE
Guide to the Location Broker	AA-PBKSA-TE
Guide to the nawk Utility	AA-PBKPA-TE
Guide to the Network File System	AA-ME99B-TE
Guide to Network Programming	AA-PBKWA-TE
Guide to Preparing Software for Distribution on ULTRIX Systems	AA-MG62B-TE
Guide to PrintServer Clients	AA-PC31B-TE
Guide to SCAMP	AA-PELFA-TE
Guide to Sharing Software on a Local Area Network	AA-PKDUA-TE
Guide to Software Licensing	AA-PBKRA-TE
Guide to the Source Code Control System	AA-ME84B-TE
Guide to System Crash Recovery	AA-ME94B-TE
Guide to System Exercisers	AA-ME96B-TE
Guide to System and Network Setup	AA-ME88C-TE
Guide to System Shutdown and Startup	AA-ME91C-TE
Guide to the User Account Manager	AA-PGL4A-TE
Guide to the uucp Utility	AA-MF03B-TE
Guide to VAX C for ULTRIX	AA-ME83B-TE
Guide to Writing and Porting VMEbus and TURBOchannel Device Drivers	AA-PGL5A-TE
Guide to X/Open curses Screen Handling	AA-LY27B-TE

Table 7-1: (continued)

Title	Order Number
Guide to the X/Open Transport Interface	AA-PBKXB-TE
Guide to the Yellow Pages Service	AA-ME00B-TE
Introduction to the CDA Services	AA-PB0JA-TK
Introduction to Networking and Distributed System Services	AA-ME97B-TE
Kernel Messages Reference Manual	AA-PBKUB-TE
The Little Gray Book: An ULTRIX Primer	AA-MG64B-TE
MH User's Guide	AA-PFTEA-TE
OSF/Motif User's Guide (V1.0) [a]	AA-PC7QA-TE
OSF/Motif Programmer's Guide (V1.1) [a]	AA-PC7MB-TE
OSF/Motif Programmer's Reference Manual (V1.1) [a]	AA-PC7NB-TE
OSF/Motif Style Guide (V1.1) [a]	AA-PC7PB-TE
The Packet Filter: An Efficient Mechanism for User-Level Network Code	AA-PBM2A-TE
Performance Management Guide	AA-PKDVA-TE
POSIX Conformance Document	AA-LY25C-TE
Reader's Guide and Master Index	AA-ME82B-TE
Reference Pages Section 1: Commands (A-L)	AI-PC0WA-TE
Reference Pages Section 1: Commands (M-Z)	AD-PC0WA-T1
Reference Pages Section 2: System Calls	AA-LY15B-TE
Reference Pages Section 3: Library Routines	AA-LY16B-TE
Reference Pages Section 4: Special Files	AA-LY17B-TE
Reference Pages Section 5: File Formats	AA-LY18B-TE
Reference Pages Section 7: Macro Packages and Conventions	AA-LY19B-TE
Reference Pages Section 8: Maintenance	AA-LY20B-TE
Reference Pages for Unsupported Software	AA-MF05B-TE
Security Guide for Administrators	AA-PBKTA-TE
Security Guide for Users	AA-PBKQA-TE
Supplementary Documents, Volume 1: General User	AA-MF06A-TE
Supplementary Documents, Volume 2: Programmer	AA-MF07A-TE
Supplementary Documents, Volume 3: System Manager	AA-MF08A-TE
Technical Summary	AA-MG63B-TE
ULTRIX SCSI/CAM Architecture: Guide to Writing Device Drivers for the ULTRIX SCSI/CAM Architecture Interfaces [a]	AA-PN5HA-TE
ULTRIX and UWS Version 4.3 Release Notes [a]	AA-ME85H-TE
ULTRIX Version 4.3 Encryption Kit Installation Instructions [a]	AV-MF16D-RE
ULTRIX Version 4.3 DECwindows for OSF/Motif V1.1.3 Release Notes and Installation Instructions [a]	AA-PPJ2B-TE
ULTRIX Version 4.3 SCSI/CAM Architecture Release Notes and Installation Instructions [a]	AA-PP85B-TE
vi Beginner's Reference Card	AV-MF10A-TE

Table 7-1: (continued)

Table note:

a. New or revised for ULTRIX Version 4.3.

Table 7-2: ULTRIX Worksystem Software Version 4.3 Manuals and Order Numbers

Title	Order Number
Introduction to the ULTRIX Worksystem Software Environment	AA-MA86D-TE
DECwindows User's Guide	AA-MA87B-TE
DECwindows Desktop Applications Guide	AA-MA88B-TE
Guide to the dxdiff Visual Differences Program	AA-MA89B-TE
XUI Style Guide	AA-MG20A-TE
XUI Programming Overview	AA-MA90A-TE
Guide to Writing Applications Using the XUI Toolkit Widgets	AA-MA91B-TE
Guide to the XUI User Interface Language Compiler	AA-MA94B-TE
Guide to the dxdb Debugger	AA-MA93B-TE
Guide to the XUI Toolkit: C Language Binding	AA-MA95B-TE
Guide to the X Toolkit Widgets: C Language Binding	AA-MF09A-TE
Guide to the XUI Toolkit Intrinsics: C Language Binding	AA-MA96A-TE
X Window System	AA-PQMYA-TE
X Window System Toolkit	AA-PHF4A-TE
Guide to Developing Applications for the Display PostScript System	AA-PAJUA-TE
Display PostScript Perspective for Software Developers	AA-PBKAA-TK
Display PostScript Client Library Reference Manual	AA-PBKEA-TK
Extensions for the Display PostScript System	AA-PBKCA-TK
Color Extensions for the Display PostScript System	AA-PBKDA-TK
Display PostScript pswrap Reference Manual	AA-PBKBA-TK
Reference Pages, Section 1X and 8X	AA-MA85C-TE

Problems Resolved Since Last Release A

This appendix discusses software problems that have been resolved since the last release. These "bug fixes" include problems outstanding from previous releases of the products and problems reported during field tests of Version 4.3.

A.1 ULTRIX Problems Resolved in Version 4.3

This section discusses problems in the ULTRIX operating system that have been resolved since the release of ULTRIX Version 4.2A. A series of tables lists the topic or the name of the component, a description of the problem, and, when applicable, the tables include a reference to a Software Performance Report (SPR) or Critical Level Distribution (CLD) which has been generated by a customer reporting the problem.

Table A-1: Many Commands: Disk Quota Exceeded Problems

Component	Problem Resolved	CLD/SPR
apropos(1) cat(1) cp(1) cut(1) dd(1) diff(1) ed(1) file(1) from(1) id(1) ln(1) man(1) mv(1) pack(1) pcat(1) ps(1) pstat(8) red(1) sort(1) split(1) tee(1) unpack(1) whatis(1)	Now, these commands explicitly close the output file and check for an error on the close. Previously, these commands did not explicitly close the output file and test for errors on the close. When account quotas were in use and an account had reached its quota limit, if NFS writes were done to the account a quota error occurred. Since the error was not trapped, a zero-length output file was produced and, in some cases, the input file was deleted.	IPO-4235

Table A-2: Commands(A-D)

Component	Problem Resolved	CLD/SPR
adduser(8)	The command now verifies that all the characters specified in the UID field are numeric. Previously, it was possible to enter a UID with a numeric first character and non-numeric remaining characters.	
	The script now accepts a negative response to the prompt to create a parent directory. Previously, the script would reject negative responses.	
	The script will now add new users to any new group you create while using the script. Previously, the new group could be created but the new users were not added to the group.	
apropos(1)	Now, the command is executed when its full path name is specified. Previously, specifying the full path to execute the command would execute the man(1) command instead.	
	Now, the command output can be piped to another command.	
ar(1)	VAX only. Now, the command outputs usage error messages to stderr. Previously, the command wrote some error messages to stdout.	
audit_tool(8)	Now the -f option does not result in excessive cpu time.	IPO-5773
awk(1)	Now, the program allows as many output files as the maximum number of open files allowed per process. Previously, the limit was 10 output files.	SMU-2501
	A problem in the program's substr function is fixed. Previously, substr was executed for every other line of input that contained a single string.	ICA-35747
batch(1)	The command now verifies that the filename argument is a valid filename, and displays the message No such file or directory if the file does not exist.	
cc(1)	RISC only. Several cc compiler problems have been fixed: Now, cgram.y detects syntax errors it previously ignored. Now, common.c writes out dashes all at once, thereby speeding emacs compilations. Now, ugen.c logic contains syntax checks for use of comma operator in cases when the rightmost element in a list is to be the value of the entire expression.	MUH-1995

Table A-2: (continued)

Component	Problem Resolved	CLD/SPR
	Now, `ugen.c` build functions check the status of parameters in registers to determine when it is possible to keep a parameter in the register or to save it in a local stack.	UVO-0948
`ccr(8)`	Internal timers have been adjusted so `ccr` works reliably over extended LANs. In addition, four new options have been added: `-8`, use only 802.3 format packets `-e`, use only Ethernet packets `-v`, use verbose mode `-c`, use the specified circuit	ICA-28554
`chfn(1)`	A restriction has been removed: If the passwd entry is distributed from another host, `chfn` can modify it.	IPO-5172
`chgrp(1)`	The command now accepts numeric group names. Previously, `chgrp` assumed a numeric argument was a GID, and did not recognize numeric group names.	
	The command can now be used to allow root to recursively change group using `chgrp -R`, when a subdirectory owner is not the same as the parent directory owner.	
`chown(8)`	The command now accepts numeric user and group names. Previously, `chown` assumed a numeric argument was a UID or GID.	
`chsh(1)`	A restriction has been removed: If the passwd entry is distributed from another host, `chsh` can modify it.	IPO-5172
`col(1)`	The command now converts spaces to tabs wherever possible and preserves tabs wherever possible, as documented in the reference pages.	
`cpio(1)`	The command no longer fails with the error `Attempt to pass file to self!`. Previously, this failure occurred when the target directory's inode matched the file counter in the archive file header. Originally, the header contained the file's on-disk inode number, but this is no longer its purpose.	
	The command now writes tapemarks when writing multivolume `cpio` tapesets. The lack of tapemarks had caused failures to read to the correct end of volume.	
`cpp(1)`	The preprocessor now contains an increased `symsize` limit (16000). Previously, programs with large numbers of defines could not precompile because of the `too many defines` error.	
	The preprocessor now concatenates multiline defines into one line to prevent `dbx` session line numbers from being wrong.	IPO-5120

Table A-2: (continued)

Component	Problem Resolved	CLD/SPR
csh(1)	The shell no longer causes a segmentation fault when a null pointer is passed to the subword routine.	
	The shell no longer hangs waiting for the child process to exit when it is called from the system library call in a System V environment.	IPO-5539
	Repeated file name completions no longer produce the message Too many open files.	
	The shell now performs command substitution and output redirection from the first session created by the dxsession session manager.	IPO-5171
cut(1)	The command now correctly handles the tab character.	ICA-22403
	The command no longer allows you to specify the -c option in combination with any other option. Also, if you use the -c option and the last character written on a line is the tab character, the tab character will no longer be truncated from the line.	
dbx(1)	Many dbx problems relating to FORTRAN and DEC FUSE have been resolved. Among them are:	
	The dbx trace command now works properly.	
	The dbx command no longer causes a segmentation fault when a FORTRAN routine is called or printed.	
	The dbx command now recognizes function names it reports.	
	The dbx command's funct routine can now resolve local variables when the file base is part of the file name.	
	The stop command now works correctly.	
	The dbx command now prints floating-point variables correctly.	
	The search routine no longer fails when it follows another dbx command in the same line.	
	The dbx command no longer dumps core when the user enters the caret and backslash characters.	
	The dbx program no longer causes a segmentation fault when the illegal command printf var is issued.	
	The dbx call now works properly with FORTRAN programs.	

Table A-2: (continued)

Component	Problem Resolved	CLD/SPR
	The `dbx` program now correctly sets breakpoints in C programs that declare local variables within curly brace blocks.	IPO-5795
	The `dbx` program now correctly displays floating point arrays and shows correct values when tracing float arguments.	IPO 5732
`dc(1)`	The command now functions with zero as an exponent.	
`dd(1)`	The command now reports write errors that occur in the first pass through its buffers. Previously, the command could fail to report write errors in small files.	
	The command now works correctly when the `files=n` option is used. Previously, when no `bs`, `obs`, or `ibs` was specified, the command would process the first file correctly but would report the error `ENOBUFS` on the second file.	
`diff(1)`	The `-D` option now creates ANSI-compliant differences.	
	The command now handles lines that do not end in a newline character and reports differences on them.	
`dircmp(1)`	The command now writes all errors to stderr. In previous versions, it wrote some errors to stdout.	
`dump(8)`	The program can now be invoked through `crontab`.	

Table A-3: Commands(E-L)

Component	Problem Resolved	CLD/SPR
elcsd(1)	A looping problem has been resolved.	
expr(1)	The command accepts negative numbers as arguments. Previously, an argument of a unary minus followed by digits caused the command to fail with the non-numeric argument error message.	
	The command now evaluates expressions preceded by a parenthesis as strings. Previously, the command would ignore any characters preceded by a parenthesis.	
file(1)	The command now displays an error when it encounters spaces used as field delimiters in the magic file. Only tabs are permitted as field delimiters. Previously, the command would cause a segmentation fault under these circumstances.	
fitset(8)	The utility now adds the alternate path to inventory entries when the -D function is used in the setld command.	IPO-5908
from(1)	The command now uses the default mbox file, <HOME>/mbox, if the -f option is specified without a filename argument. Previously, the command would cause a segmentation fault if -f was used with no argument.	
	The command now displays a usage error when the -s option is specified without a sender argument.	
	The command now allows the default mbox file to contain a pathname that is as long as the number of characters defined as PATH_MAX in the limits.h file. Previously, the limit was 100 characters.	
ftp(1c)	The utility no longer ignores the account file. Previously, doing so resulted in the user being prompted for data during an otherwise automatic procedure.	AKO-2026
ftpd(1c)	A security problem has been fixed.	
gencat(1int)	Several problems have been fixed.	
id(1)	The command now displays an error if a user's GID is not in /etc/group, or if a user's UID is not in /etc/passwd. Previously, the command would produce a segmentation fault in these cases.	ICA-30348, ICA-30611
inetd(8c)	The daemon now issues a setpgrp(0,0) to release any controlling tty. Previously, rlogin with either csh or ksh as the shell, terminated with a connection closed error.	

Table A-3: (continued)

Component	Problem Resolved	CLD/SPR
ksh(1)	The restricted Korn shell, rksh, is now available as documented.	
	The shell now correctly sets $PWD in scripts. Previously, the shell would dump core when the cd command was used.	
ld(1)	RISC only. When the -A switch was used to prepare a symbol table from an image containing a reference to the special symbol _procedure_table and no object module was included in the load, the loader terminated with a segment violation. This problem is fixed.	
ln03of(8)	The set lines/page function now works on LN03 printers in landscape mode. Previously, the LN03 would print 65 lines per page in landscape mode instead of 66 lines per page.	AKO-2096
ln03rof(8)	The routine spgetchar no longer loses the first character it gets when the character is 8-bit.	
	The filter no longer produces extra blank sheets when print jobs in landscape mode finish at the top of a new page.	UVO-0857
	The filter now prints the last line in a print job when that line is the first line on a page and the job ends without a terminating newline.	
	The command is 8-bit clean. Previously, the command would strip 8-bit characters from PostScript files.	IPO-5686
lookbib(1)	RISC only. The command now traps usage errors when no argument is specified. Previously on RISC systems, the errors were not trapped, resulting in segmentation faults.	
ln(1)	Now, the -f option does not cause the command to overwrite dangling symbolic links and display the error File exists.	
lpf(8)	The filter now passes 8-bit characters.	
	The filter now prints a form-feed character whenever the page count is changed. Previously, the filter printed at 66 lines per page even after the -z option was used to change the page length.	IPO-5300
lpd(8)	Now, the daemon does not exit prematurely.	
	Now, the daemon sends a Ctrl/D before testing serial PostScript printer status.	IPO-5983
	Now, the daemon retries reads when it fails with the EINTR error while reading an archive file.	

Table A-3: (continued)

Component	Problem Resolved	CLD/SPR
	The daemon no longer connects to a remote printer with a closed file descriptor. Previously, `lpd` would cause a hang when it closed the file descriptor after an unsuccessful connection to a remote printer, then attempted to connect using that same file descriptor.	
	The daemon now shares LAT printers with VAX/VMS systems. Previously, the daemon never closed the printer tty once it was opened, thereby preventing other systems from gaining access to the printer until the local print queue was empty (which, on a busy system, might never happen).	IPO-5553
`lprm(1)`	The command now waits for the current print daemon to exit before starting a new daemon. Previously, killing a print job could result in a race condition when a new print daemon started up before an old daemon exited.	IPO-5983
	The command no longer fails to remove a remote print job when the user's host name is uppercase.	AKO-2051
`lprsetup(8)`	The utility now outputs the `Nu` value as a type string instead of type number.	
`ls(1)`	The command now returns the inode number of the referenced file in all cases. Previously, when the argument on the command line was a directory, the command would return the inode number found in the directory, which was wrong for symbolic links and mount points.	
`ltf(1)`	The utility now strips padding from files extracted from foreign tapes.	IPO-5006

Table A-4: Commands(M-R)

Component	Problem Resolved	CLD/SPR
mail(1)	The utility no longer dumps core when a user replies to a message sent from a DECnet Phase V node.	UVO-1089
man(1)	The command no longer produces a segmentation fault when an invalid option is given.	ICA-25858
	The command automatically searches section thr for DECthreads help files.	
mh(1mh)	The mail handler no longer dies when sendmail is not running.	IPO-4795
	The mail handler no longer produces the message inc: no mail to incorporate when it is using the -silent option.	
mkcatdefs (1int)	The command can now handle long file names. Also, if more than one space separates the symbol from the message text, an extra space is no longer inserted in the message.	
mount(1)	Now, the command works properly when you attempt to access a write-locked disk and forget to include the -r option. Previously, the command would fail when attempting to access write-locked RX26 floppy disks.	
mountd(8nfs)	Now, the daemon correctly updates /etc/rmtab when a client is unable to mount a file system.	
mt(1)	The command now returns a device offline error when an attempt is made to rewind a TK50 tape.	ICA-25700
named(8)	A new option exists for named running in netsafe mode to allow or disallow non-Hesiod updates from systems outside of its zone. Use the -nh option to allow named to run in netsafe mode and receive non-Hesiod updates outside of the systems zone. Use the -n option to disallow any update outside of the systems zone.	IPO-5373
nawk(1)	The command now performs comand line variable assignments of the form var=value prior to processing any BEGIN pattern.	IPO-5129
	The command now outputs the audible bell as documented when the escape sequence backslash a is printed. Previously, the command would output the character a instead of ringing the bell.	

Table A-4: (continued)

Component	Problem Resolved	CLD/SPR
	The command now scopes temporary nodes. Previously, a node could be freed by a lower-level routine before the node which had created the routine was finished with it.	
	The functions `toupper` and `tolower` now calculate the size of the string to be converted. Previously, the functions could append extraneous data to the end of the output string.	
	White space dependencies in the `yylex` component have been removed. Previously, the dependencies caused unaligned access problems on RISC machines and segmentation faults on VAX machines.	
`ncp(8dn)`	The kernel `nsp` component of DECnet-ULTRIX has been corrected. Previously, attempts to change the DECnet executor outgoing timer parameter were ineffectual. The command would succeed but would have no effect. Previously, the command was unable to send zero-length packets. Also, performance has been improved during large data transfers.	
`od(1)`	The command now returns an exit status of zero upon successful completion.	
`opser(8)`	The program has been changed to work with the ENHANCED security level in addition to BSD and UPGRADE levels. Previously, ENHANCED security restricted the root and operator accounts to the same password. Now, they can be different. Note that the operator account's uid should be changed to 5 from 0 to effect this change.	IPO-5903
	The program no longer strips the upper digit from `shutdowntime` as the system is shut down. Previously, this problem caused 98 minutes to be treated as 8 minutes.	

	The program no longer produces the following incorrect message when a system is restarted:
	`Command invalid unless time sharing stopped;`
	`restart can only be run from the console device`
		IPO-4653

`passwd(1)`	The command now allows you to change an expired BIND/Hesiod distributed password by using the UWS `Xprompter` window.	IPO-5481, IPO-5509
`pg(1)`	The command now checks the return code from a `read` system call.	

Table A-4: (continued)

Component	Problem Resolved	CLD/SPR
	The command now traps errors that occur when the terminal is powered off and the command is running. Previously, if the command received input from a file, it exited when the terminal was powered off. Now, if the command gets input from `stdin`, it does not exit when the terminal is powered off and will not consume any CPU time until the terminal is powered on.	
`presto(8)`	The command no longer fails on an NFS client when the `automount` daemon is used.	
`pstat(8)`	Now displays only vector totals when the `-T` option is specified.	
	RISC only. The `-u` option now works as documented.	ICA-34149
	The `-t` option now works as documented.	
	The `-f` option now displays ports and pipes in addition to gnodes and sockets.	ICA-23805
	The usage error displays and defaults for options have been improved.	
`rarpd(8c)`	The daemon can now receive an address that is not in `/etc/ethers`.	AKO-1839
	The daemon now properly closes the `/etc/ethers` file and the `-f filename` file every 10 minutes. Previously, the check was made every 100 minutes.	
	The daemon now processes the `-f filename` option correctly.	
`rpc(1c)`	The command now expands wildcards correctly when copying files from one remote host to another through a local host. Previously, the copy would fail with the message `No such file or directory`.	UVO-1133
`rdump(8c)`	The command now transitions to the second tape of a multivolume tape set. Previously, the command failed when the end of the first tape was reached.	
`removeuser(8)`	The script now displays the home directory path when asking whether you want to remove the user's home directory, all subdirectories, and files. Previously, a misreading of the instructions could result in the user removing `root`.	
`restore(8)`	The command now restores all files present in a dump. Previously, an incremental restore failed to supercede in several situations with regular files, special files, and symbolic links.	IPO-4720
`ris(8)`	Now, more than 9 products can be in a RIS area and products can be deleted from a RIS area.	

Table A-4: (continued)

Component	Problem Resolved	CLD/SPR
`rksh`	The restricted Korn shell is now available as documented.	
`rlogin(1c)`	The command now handles `^s` and `^q` sequences the same way for both 7- and 8-bit encodings.	IPO-4939
	Now, command line edits no longer hang.	
`rm(1)`	The command now outputs all errors to stderr. Previously, the command wrote some errors to stdout.	ICA-33250

Table A-5: Commands (S-Z)

Component	Problem Resolved	CLD/SPR
sendmail(8)	Version 5.65 is incorporated into the product.	
	The mail handler no longer loses text when passing mail to mail11.	
	RISC only. The load average threshold now works for RISC machines.	
setld(8)	Now, the fitset utility adds the alternative path to all the alternative entries when the -D function is used.	IPO-5989
	Now, subset names can be mixed uppercase and lowercase characters.	
sh(1)	Now, when allocating memory, the shell determines whether the address it needs is beyond the last block allocated. If it is, the shell allocates memory intil is within the shell's address space. Previously, the shell relied on the SIGSEGV signal to aid in memory allocation.	
	The shell's LOGNAME environment variable is no longer corrupted when a built-in function such as pwd or echo is performed.	UVO-1099
spell(1)	Now, the command outputs all errors to stderr and sets status to 1 when exiting on an error. Previously, the command wrote some errors to stdout, and the return value in these cases was 0 (success).	
tapex(8)	RISC only. The -s test no longer fails. Now, the returned byte count correctly reflects the amount of valid data returned.	
	The -q test no longer fails when the utility attempts to skip past a file which had not been written to the tape.	
tar(1)	The command now checks that all n-buffered I/O writes are successful. Previously, a failed write could go undetected in very small tar sets.	
	The command no longer fails intermittently during writing of multivolume tar sets. Previously, the command would report Error 0 and prompt for rewrite.	IPO-4800
	The command no longer produces intermittent directory checksum errors while reading a multivolume tar set on the TA90 tape drive.	UVO-0938
tbl(1)	The preprocessor now correctly accesses macro files using the -mm and -ms options. Previously, attempts to do so resulted in an open error on the macro files.	

Table A-5: (continued)

Component	Problem Resolved	CLD/SPR
	The preprocessor no longer produces a segmentation fault when it processes a file that contains a table with one line of data whose first field is NULL.	
telnet(1c)	The daemon is now 8-bit clean and a security problem has been resolved.	VBO-0687
	The daemon now sends carriage return characters as CR in binary mode. Previously, it sent carriage returns as CR Null.	
	The daemon now prints error messages when gethostbyname fails and does not set the h_errno variable.	
	The daemon now correctly toggles CRMOD on the pty in response to the commands DO ECHO and DON'T ECHO.	IPO-4674
	Now, the daemon can log connections, including the IP address, to a syslog file.	IPO-5203
tftp(1c)	The tftp protocol now handles up to 65534 data blocks. Previously, the limit was 32767 blocks.	
tic(1)	The command now gives the correct usage error message. Also, the command no longer dumps core when no options are given.	
tip(1c)	The utility no longer dumps core when sending a file through the ~> command.	IPO-5148
	The command now turns off the EXCLUSIVE bit when it exits. Previously, the command, by leaving the bit on, prevented others from using the tty.	IPO-4674
touch(1)	The command now processes file names that contain only numbers. Previously, the command would ignore the file or issue the bad date conversion message.	
true(1)	Now, /bin/true always returns zero. Previously, if there was an error return defined in the .cshrc file, executing a csh script would return that error as the value of /bin/true.	
uname(1)	The mode of file has been changed from 4755 to 755. The command did not need to be at mode setuid.	
uucp(1c)	RISC only. Segmentation faults no longer occur when trying to use uucp on RISC systems.	
	The command now turns off the EXCLUSIVE bit when it exits. Previously, the command, by leaving the bit on, prevented others from using the tty.	IPO-4674, IPO-4580

Table A-5: (continued)

Component	Problem Resolved	CLD/SPR
uustat(1c)	The uucp.night script no longer truncates the R_stat file when encountering an entry that is more than $DTIME hours old. Previously, any entries later than $DTIME were lost and the R_stat could become zero-length.	IPO-05984
uux(1c)	The command no longer produces a segmentation fault when it is executed from a terminal not in /etc/utmp, for example when the command is issued from a script.	
vi(1)	The editor can now open a file with very large buffer sizes from an NFS-mounted file system.	IPO-4592
	The editor now gives an error when a single line is too big to fit on a screen while in insertion mode. Previously, the editor would either dump core or hang the terminal.	CXO-7437
view(1)	The editor now gives an error when a single line is too big to fit on a screen while in insertion mode. Previously, the editor would either dump core or hang the terminal.	CXO-7437
vmstat(1)	The command now prints correct values for per-second fields such as interrupts, syscalls, and context switches, whenever an interval and a count are specified.	ICA-30388, ICA-33171
xargs(1)	The -i option now works.	ICA-25636, ICA-30922
xstr(1)	The command now correctly parses the double quote character when the character is preceded by a backslash. In addition, the command now parses escaped new lines.	
yes(1)	The command now traps errors and exits instead of looping.	

Table A-6: System Calls

Component	Problem Resolved	CLD/SPR
`creat(2)`	The call is now POSIX compliant and works as documented.	IPO-5008
`semget(2)`	RISC only. Now, `sem_otime` is properly set to zero as documented.	
`seteuid(2)`	The call has been changed to comply with the IEEE STD 1003.1 (1990) for the `setuid(2)` call. Now, if `seteuid(2)` has moved a process to a nonprivileged state (ID != 0), the application can move the process back to a privileged state.	
`shmop(2)`	The shared memory routine `shmdt` now correctly recomputes the size of a shared memory region.	AKO-1938
`sigsetmask(2)`	Users can now call `sigsetmask` with the high order bit of the mask set.	

Table A-7: System Components

Component	Problem Resolved	CLD/SPR
vm	VAX only. A routine has been added to the vm_swap.c code to prevent buffers greater than 16Kbytes from reaching the driver. Previously, the driver would fail when reading a VAX SCSI tape with a record size greater than 16Kbytes.	
	Large jobs selected to run no longer get swapped out before then can begin.	IPO-5207
	Now, the routine asynchsel() checks for null pointers to asynchronous buffers. Previously, the failure to check for null pointers when they were present caused the system to crash.	
dli	Down-line loads over FDDI links no longer fail with the message too long error. Previously, the maximum data length allowed limit was calculated incorrectly, resulting in the error being produced for valid messages.	CXO-8403
gfs	The panic smp_lock_long: invalid lock type no longer occurs when the user enters CTRL/C on an NFS locking process.	IPO-5645
	The spec_rwgp routine now uses the ioflag parameter for the FNBUF bit. Previously, reads from the device would produce the ENOBUFS error when a character device was open for asynchronous I/O by one process and for synchronous I/O by another process.	IPO-5506
	The exit routine now closes all open files when a process exits. Previously, locks on the highest file descriptor were not being released when a process exited.	IPO-4895
lockd	Now, NFS locking recovers when a server reboots.	MUH-1791
	Now, more than one lock can be set on files exported from Sun systems. Previously, once a lock was set on a remote file, attempts to lock other files would block until the lock on the first file was removed.	IPO-5215
sys	VAX only. Now, the VAX interrupt stack size can be configered in the system configuration file. Previously, the interrupt stack overflowed in system using DECnet Phase V software.	
	RISC only. Now, performance is improved when ta client and server are on a single host.	IPO-5883
	The tcp_input.c routine has been recoded to comply with the protocol for tcp input.	CXO-7887

Table A-7: (continued)

Component	Problem Resolved	CLD/SPR
	The file descriptor reference count is corrected so a `sendmsg()` call to a socket which failed can be completed.	
	The wakeup channel is now zeroed after the wakeup call. Previously, awakening select calls were not being processed.	IPO-5112
	RISC only. Now, the `desfree`, `lotsfree`, and `minfree` fields of the `maxabuf` global definition in the `param.c` file are the same for RISC as for VAX systems.	
	DECnet Phase V DTSS now correctly changes the `boottime` value if it sets the time value with an infinite inaccuracy.	
	Now, the system no longer panics because the kernel is attempting to unlock the same page twice. Previously, the system would panic, with the message `MUNLOCK: dup page unlock`.	
	Now, the NBIB panic `interlock sequence error` no longer occurs on 8800 systems.	IPO-4635
	Now, the OSI Transport layer can transmit infinite TSDUs, an X.400 requirement for DECnet OSI.	
	The Gateway-to-Gateway protocol identifier constant value has been changed from 2 to 3 to conform to RFC 1060.	ICA-32882
	Now, you can force crash dumps from the console on DECstation 5000 processors. Previously, all systems using an SCC terminal chip that used a terminal for a console hung, rather than crash dumped when you tried to force a dump from the console.	
	Now, the `sys/h/resource.h` file is safe from multiple includes. Previously, programs that included `resource.h` and `buf.h` could not compile because of multiple definitions of structures in `resource.h`.	
	Now, the shared memory routine `checkpage(vm_page.o)` correctly checks for shared memory segments attached to exiting processes. Previously, the failure to do so led to system panics, such as `flushpte: lisams` and `bad c page`.	IPO-5351
	The system no longer panics with `tblmiss on invalid kernel page` or `kpteseg miss outside utlbmiss` messages when a user is logged in to a non-ULTRIX system and presses Ctrl/C while running `cat` on a large file.	IPO-5145

Table A-7: (continued)

Component	Problem Resolved	CLD/SPR
	VAX only. Now, the shared memory routine `obreak()` calls the `dmexpand()` function when a process has a shared memory segment attached and is expanding or shrinking VAS. Previously, allocated swap space was not freed. Ultimately, the system would run out of swap space.	IPO-5657
	The `shmat` function no longer returns an `invalid argument` error when a null address is specified.	
	Now, a server using an `AF_INET` socket can call `listen(2)` with a backlog of zero and get a return. Previously, this client connection blocked in `connect()` and never returned.	
	Now, the system does not panic with the `smp_lock_long: not owner` message when exiting from `dbx` while the process being traced was blocked waiting for a file lock.	IPO-5261
	The shared memory detach routine `shmdt` now works properly. Previously, various panics were issued because the routine did not account for the gap between data and the start of the P0 page tables for processes swapped out and then swapped back in.	AKO-01938
	Connect requests pending on a socket are now returned in FIFO order. Previously, they were returned in FILO order, which caused connections to time out.	
	Now, the UNIX domain uses a `sockaddr_un` structure for addresses, instead of using a generic `sockaddr` structure. Previously, use of the generic structure caused the from socket address to be truncated to 16 bytes, thereby preventing full pathnames from being used.	
	The `buf.h` file has been changed.	UVO-0850
	The XTI socket layer no longer panics when it processes a datagram that does not have an address associated with it.	UVO-0801
	Several timing problems related to lat `ttys` have been fixed.	IPO-4243
	VAX only. Processes using the vector processor no longer find garbage in vector processor registers when executing on a VAX 9000.	
	RISC only. Floating point source registers are now preserved across exceptions.	IPO-5015

Table A-7: (continued)

Component	Problem Resolved	CLD/SPR
	RISC only. Errors detected in floating point instructions no longer fail to set the branch delay bit in the cause register that is passed to the signal handler. Therefore, FORTRAN signal handlers no longer fail on floating point underflows.	IPO-5329
`ufs`	The system no longer panics with the message `ufs_rwgp: messed up gp,xp` when `gp->textp` is a pointer to the mount functions.	
	The routine `ufs_rename()` no longer loops when renaming directories to new parents.	IPO-5389

Table A-8: Library Routines

Component	Problem Resolved	CLD/SPR
`catopen(3int)`	The routine now causes much less memory to be used when opening a catalog with sparse messages defined.	
`ctype(3)`	The `isdigit()` function now returns zero for values greater than 0177. Previously, the function had returned unpredictable results for characters greater than octal 177.	
`cursesX`	The `(overlay)` and `(overwrite)` functions now work correctly. Previously, `overwrite` did not properly calculate window coordinates and `overlay` could fail depending on the window coordinates.	UVO-00887
	The `smkx` and `rmkx` capabilities in the terminfo database have been corrected to not affect whether the keypad is in application or numeric mode. Previously, using the `keypad` function in a `cursesX` application to change the cursor key mode from normal to application would automatically change the keypad to application mode also.	IPO-5529
	The routine `getch` no longer hangs indefinitely waiting for the next character, after a user has entered the escape character as input.	IPO-5286
	The `pnoutrefresh` routine boundary conditions have been fixed, thereby preventing unaligned data errors. Previously, `pnoutrefresh` would attempt to place the cursor off the screen.	VBO-0825
	The down arrow key is no longer lost when toggling between multiple curses programs. Previously, the down arrow key would return escape (033) instead of down arrow (042).	IPO-5678
	In keypad mode, the keypad no longer returns three bytes (the 3-character escape sequence) instead of a single byte (the right arrow key).	IPO-5284
	The cursesX library no longer fails to restore terminal settings while tracking terminal attributes through the `tty` terminal interface.	IPO-5285
	The file `term.h` is now ANSI compliant.	
	The file `cursesX.h` is now ANSI compliant.	
	The library routines `mvscanw(3cur)` and `mvwscanw(3cur)` no longer cause programs to fail with the loader message `Undefined: _sscans` because they now call function `__sscans`, which is defined. [Note the double underscore in `__sscans`.]	

Table A-8: (continued)

Component	Problem Resolved	CLD/SPR
gethostent(3n)	The routine now initializes the h_aliases element in the hostent structure to NULL to prevent wrong aliases resulting from BIND queries.	
libmalloc(3)	Now, the routine allocates 3 bytes when zero bytes are requested, making its behavior identical to that of malloc.	
malloc(3)	VAX only. Now, the routine returns an area that is page-aligned. Previously, memory would not be freed when the return pointer was not already on a page boundary.	IPO-5711
	Now, both malloc and libmalloc return the minimum packet size; both return a NULL pointer if there is not enough available memory to honor a memory request.	
math(3m)	RISC only. Now, math function types are defined in math.h, and users do not have to explicitly declare the function types in programs.	
ndbm(3)	The routines dbm_open, dbm_close, dbm_fetch, and dbm_store have been modified to use less overhead in large data bases.	
openlog(3)	Previously, issuing multiple calls to openlog without calling closelog resulted in syslogs being lost.	
	The code also has been changed so openlog is now idempotent.	
setuid(3) seteuid(3) getuid(3) geteuid(3)	Now, these routines check for a lower bound as well as an upper bound on the uid. Previously, there was no check for negative values for UID_MAX.	

Table A-9: File Formats, Special Files, and Macro Conventions

Component	Problem Resolved	CLD/SPR
`disktab(5)`	These disk table entries for the RZ57 disk have been corrected: nc=1854; pc# 1954050; pf# 524546; ph# 1122562.	
	The RZ58 entry has been corrected. Previously, it referenced parts of the disk that did not exist.	
`dhu(4)`	DHU11 ports no longer hang if framing errors are received on systems configured with the `sys_tpath` option enabled.	AKO-1784
`dmb(4)`	The adapter no longer produces the `interlock sequence error` panics.	IPO-4635
`dmz(4)`	Now, the `getty` is awakened when a connection on a modem line is established.	CXO-7313, IPO-5139
`ln(4)`	The driver no longer generates the `unaligned access` panic when copying Ethernet packets.	
	The command now correctly increments the number of initially-deferred blocks sent. Previously, the command always showed the number of initially-deferred blocks sent as zero, regardless of the actual number.	IPO-5580
`lqf(4)`	The filter now passes 8-bit characters.	
`lta(4)`	Now, the `lat_iframe()` function is called with the correct set of arguments.	
	Now, the driver does not block on an open call if the user has specified a nonblocking open.	
	The cpu table in `lat_sc11.c` has been updated.	
`magic(5)`	The file now recognizes PostScript text files which contain a shortened version of the identification string, thereby allowing the `file` command to recognize files created by translators that do not output the full identification string.	
`mtio(4)`	The TA90 `mtioctl` now executes a rewind/offline operation without reporting an erroneous error when another TA90 drive happens to be active on the same TA90 controller.	IPO-5232
`ne(4)`	The network driver now properly refrains from incrementing the collision counter for multiple collisions when single collisions occur and when the collision counter has reached its maximum value.	

Table A-9: (continued)

Component	Problem Resolved	CLD/SPR
nfs(5nfs)	Now, users who have permission to read an NFS file can do so, even after root attempts to read the file without permission to do so. Previously, after root attempted to read a file and was denied access, other users could not gain access, even though they had permission.	UVO-850
	Now, the vop_setattr() routine correctly uses the uio_offset offset field.	IPO-5566
	The routine cdfs_getdirent no longer assumes that the user buffer is a multiple of 2048 bytes. The new assumption is that the user buffer will be a multiple of 512 bytes.	MUH-2014
	A cast operator problem in the utility has been fixed.	
ni(4)	Now, the driver correctly formats minimum-size Ethernet packets. Previously, packets would get dropped by some drivers because they were not correctly formatted.	
nl_types(5int)	The nh_types.h file can now be included twice in a module without producing compilation errors.	
packetfilter(4)	Now, incoming packets behave as described in the documentation. Previously, there was a restriction requiring incoming packets to be encapsulated Ethernet packets.	
	Now the EIOCDEVP ioctl returns the device type and the MTU. Previously, the ioctl returned the device type as ENDT_10MB and the MTU as 1500.	
	Now, the packetfilter allows a user application to receive packets that are sent to the local host, if no other protocol in the kernel wants to use the packet.	
	The driver can now write packets with a total length of more than 1500 bytes.	IPO-5250
	The packet filter no longer returns EINVAL for read requests. Previously, after running the packet filter for a long time, the file system stopped checking for negative file offsets, which caused the invalid return.	UVO-0804
printcap(5)	Data is no longer lost when ULTRIX and VMS systems share a LAT PostScript printer. To accomplish this however, the serial PostScript printer must have the read/write entry set to rw and the new ps option set to ps=PS.	CXO-07527
pty(4)	Now, if you open the slave side of the pty before opening the master side, the error EIO is returned. Previously, this case resulted in the process never being awakened when the master side was opened, making the pty unusable.	UVO-1019

Table A-9: (continued)

Component	Problem Resolved	CLD/SPR
	The panic `smp_unlock_long: not owner` no longer occurs.	IPO-5468
`tcp(4p)`	Improved performance when both client and server are run on the same host.	IPO-5883
`ra(4)`	Now, timeout code in the MSCP driver prevents i/o hangs during shadowing operations on MSCP devices.	
`terminfo(5)`	The database entry for the VT200 terminal has been updated with an alternate character set.	
`xna(4)`	The driver now calls the packetfilter when the physical address is set. Previously, the code would fail when the EIOCDEVP ioctl was used by the packetfilter to get the address before it was set.	
	A `printf` statement with the incorrect number of arguments has been corrected.	

Table A-10: Miscellaneous Components

Component	Problem Resolved	CLD/SPR
`dti`	Now, the up-arrow key no longer fails to keyclick on Personal DECstation systems.	
`lg31of`	Now, customized settings are retained across jobs by preventing printer reset at start of jobs.	IPO-5306
`lk201.c`	Output to a keyboard while device interrupts are disabled no longer result in key repeats.	IPO-5159
`lpdfilters`	Print filters escape sequence problems have been resolved: <esc>c on lg31 printers <esc>[4m for filters: ln03of, la750f, lg31of, and lg02of	AKO-1870

A.2 ULTRIX Worksystem Software Problems Resolved in Version 4.3

This section discusses problems in the ULTRIX Worksystem Software that have been resolved since the last release. The table lists the topic or the name of the component, a description of the problem, and when applicable, a reference to a Software Performance Report (SPR) or Critical Level Distribution (CLD).

Table A-11: ULTRIX Worksystem Software Problems Resolved

Component	Problem Resolved	CLD/SPR
`dxnotepad(1X)`	Now, users are notified when disk quota is exceeded while saving a file.	IPO-4235
`dxsession(1X)`	Keymap files for the LK401 are now provided.	MUH-1953
	Exiting from a DECterm window when your login shell is `sh` or `sh5` no longer causes the message `p_write() : write failed, errno = 5` to be printed repeatedly in the Session Manager Message window.	IPO-5801
	The `rootPasswd` resource now works as documented.	ICA-35224
	A security problem has been resolved.	IPO-5296
	If a DECnet node name is entered in the Session Manager's Customize Security window and the name is not a recognized node name, the Session Manager no longer attempts to look up the node name as an Internet node. Previously, this problem resulted in very slow logins on workstations in DECnet Phase V networks when the host database was distributed using BIND.	IPO-5456
`dxterm(1X)`	Using the `-e` option no longer causes DECterm to crash.	MUH-1770
`dxuam(8X)`	The User Account Manager no longer fails with the error `X Error: BadName - font or color name does not exist` when the display is directed to a VT1200 or VT1300 X terminal or to a workstation running VAX/VMS.	UVO-1091
`dxue(1X)`	Now, the User Executive correctly displays files with a protection mode of 710.	IPO-5094
	The User Executive no longer aborts when you use the `-d` option on the command line to display on another node, as previously happened when the hosts database was served by Yellow Pages and a large number of menus and menu items were in the system profile.	IPO-5535

Table A-11: (continued)

Component	Problem Resolved	CLD/SPR
	The User Executive no longer aborts when you select the owner field in the Customize File Properties window and the owner of the file used in the sample line (the last of the currently displayed files) is not in the password database.	
	The fields of the file display are now aligned in columns similar to the output of the `ls -l` command.	
`fonts`	Many font file problems have been fixed.	
`Xlib`	The callback function from `XSelectAsyncInput` is no longer inadvertently canceled by other Xlib routines.	
	The Xlib header files now can be compiled without errors or warnings from ANSI C compilers.	
	Xlib problems with the use of the shared memory transport have been resolved.	
	The `XPeekEvent` function now handles errors correctly.	
	The `XFillArcs` routine now can be called with no arcs specified without causing an error.	
	The `XOpenDisplay` routine now copies the defaults database from screen 0 instead of from the default screen.	
	The `XLoadQueryFont` routine now correctly wraps the number of requests at 0xffff.	UVO-0803
`Xprompter(1X)`	The utility now correctly updates the BIND/Hesiod distributed password database when the user is asked to enter a new password during the login procedure.	IPO-5481,IPO-5509
`Xserver`	Some `Xv` problems on multiscreen systems have been resolved.	
	A memory leak has been corrected.	UVO-1171
	An XCopyPlane problem on systems with the MX option has been fixed.	IPO-5620
	Several `Xwst3d` problems have been corrected.	IPO-5873
	A problem drawing certain graphics on systems with the HX option has been corrected.	UVO-1164
	A problem performing zoom functions on systems with the HX option has been corrected.	IPO-5790
	A problem drawing zero-length line segments on systems with the HX option has been corrected. (Also mentioned in UVO-1305.)	IPO-6036, UVO-1280
	The server no longer crashes while drawing lines in windows with negative origins.	
	The `XCopyPlane` function no longer causes a display corruption when copying a pixmap.	UVO-0966

Table A-11: (continued)

Component	Problem Resolved	CLD/SPR
	The `GCOps` table is now consistent across MX operations.	
	Now, the cursor can approach the lower HotBox boundary. Previously, the cursor could not approach the boundary closer than the `xhot` and `yhot` distances.	
	The setting of the TX duty cycle register has been reset for the `Xv` function.	

FDDI Support in ULTRIX B

This appendix discusses how FDDI is supported in ULTRIX.

B.1 Support of the FDDI Network Interface for RIS and DMS

Both the Remote Installation Services (RIS) and Diskless Management Services (DMS) support the Ethernet network interface and the Fiber Distributed Data Interface (FDDI).

B.2 Fiber Distributed Data Interface Overview

The Fiber Distributed Data Interface (FDDI), as specified in the ANSI X3T9.5 standard, uses optical fiber as the transmission medium to provide a high-performance general purpose interconnection among computer and peripheral equipment.

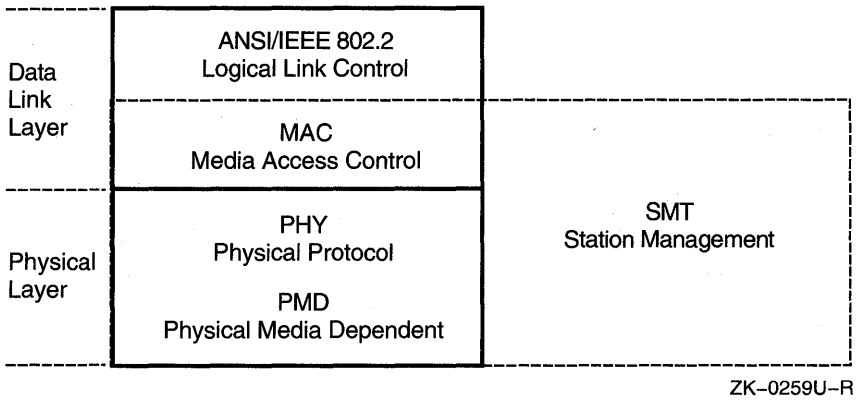
FDDI networks have a ring topology. A station on the ring gains the right to transmit data by receiving a token that is generated by its Upstream neighbor. The station must, in turn, issue a new token to its downstream neighbor following the completion of its data transmission cycle to allow other stations access to the ring.

The FDDI operates at the physical and data link layers. The primary characteristics of the physical layer are as follows:

- Data rate of 100 megabits per second
- Maximum of 200 kilometers of total fiber path length
- Maximum of 500 hosts per fiber optic segment (100 physical connections)
- Maximum transmit frame size of 4500 octets
- Optical fiber in a dual, counter-rotating ring configuration

The physical and data link layers can be separated functionally as described in Figure B-1.

Figure B-1: Physical and Data Link Layers



The physical layer (PL) contains two sublayers:

- Physical Protocol (PHY) entity--the upper layer, which defines encode/decode and clock requirements of data between the PMD sublayer and the data link layer of FDDI.
- Physical Media Dependent (PMD) entity--the lower layer, which provides the medium, connectors, optical bypassing and driver/receiver requirements. Each station may provides a multimode fiber (MMF) or a single mode fiber (SMF) PMD connection to the media.

The data link layer also contains two sublayers:

- Media Access Control (MAC) entity--controls the accessing of the medium, address recognition, generation and verification of frame check sequences, and delivery of frames.

  Logical Link Control (LLC) entity--provides a common protocol between the MAC and the network layer. The LLC layer for the FDDI station is specified in ANSI/IEEE Standard 802.2.

The Station Management entity (SMT), which uses both the physical and data link layers, provides the control necessary for the stations to work cooperatively on a ring.

The multimode fiber and single-mode fiber PMD connection refers to the PMD entity attachment. The multimode fiber provides dual attachment to the ring, while the single-mode fiber provides a single attachment. The dual attach wiring concentrators (DEFCN) can support up to eight single attach ports (such as DEFZA) to connect to the FDDI rings.

Ethernet local area networks (LANs) and FDDI LANs can be interconnected using the FDDI-Ethernet (DEFEB) bridge. The bridge forwards traffic from hosts on the FDDI to hosts on the Ethernet segment, while filtering traffic between nodes that are connected to the same side of the bridge. The bridge can respond to the Maintenance Operation Protocol (MOP) and FDDI station management (SMT) directives.

B.2.1 Managing the FDDI Network Interface with the Simple Network Management Protocol

The Management Information Base (MIB) variables of the FDDI network interface can be accessed through the SNMP protocol from any SNMP client. The MIB interface speed and interface type variables have to be added to the SNMP daemon configuration file, `/etc/snmpd.conf`.

The following are the two entries that you must add:

```
interface speed fza0 104857600

interface type fza0 15
```

Note that you must restart the SNMP daemon after you modify the `/etc/snmpd.conf` file.

Refer to the `snmpd.conf`(5n) and `snmpd`(8n) reference pages for more information.

B.2.2 Managing the FDDI Interface

You can use the `netstat -I fza0 -s` command to obtain a listing of FDDI counters, status, and characteristics.

The following is sample system output from the `netstat` command for an FDDI interface:

```
fza0 FDDI counters at Thu Nov 29 10:49:54 1990

       21217 seconds since last zeroed
  4294967295 ANSI MAC frames count
           0 ANSI MAC frame errors
           0 ANSI MAC frames lost
   287547013 bytes received
   106856927 bytes sent
      678437 data blocks received
      166707 data blocks sent
    38493013 multicast bytes received
      390967 multicast blocks received
      489271 multicast bytes sent
        5328 multicast blocks sent
           0 transmit underrun errors
           0 send failures
           0 FCS check failures
           0 frame status errors
           0 frame alignment errors
           0 frame length errors
       10850 unrecognized frames
           0 unrecognized multicast frames
           0 receive data overruns
           0 system buffers unavailable
           0 user buffers unavailable
           0 ring reinitialization received
          69 ring reinitialization initiated
           1 ring beacon process initiated
           0 duplicate tokens detected
           0 duplicate address test failures
           0 ring purge errors
           0 bridge strip errors
           0 traces initiated
           0 traces received
           0 LEM reject count
           0 LEM events count
```

```
          0 LCT reject count
          0 TNE expired reject count
          2 Completed Connection count
          0 Elasticity Buffer Errors
fza0 FDDI status

Adapter State:                      Running
LED State:                          Green
Link State:                         On Ring Running
Duplicate Address Condition:        Absent
Ring Purge State:                   Non Purger
Negotiated TRT:                     7.987  ms
Upstream Neighbor Address:          08-00-2B-18-BA-8F
UNA Time Out:                       False
Claim Token Yield:                  False
Frame Strip Mode:                   Source Address Match
Ring Error Reason:                  Ring Init Received
Last Direct Beacon SA:              00-00-00-00-00-00
Physical Port State:                In Use
Neighbor Physical Port Type:        Master
Reject Reason:                      No Reason
Physical Link Error Estimate:       15

fza0 FDDI characteristics

Link Address:                       08-00-2B-18-B3-D7
Firmware Revision:                  0.2F
ROM Revision:                       1.0
SMT Version ID:                     1
Requested TRT:                      8.000  ms
Maximum TRT:                        173.015 ms
Valid Transmission Time:            2.621  ms
LEM Threshold:                      8
PMD Type                            Multimode
```

Table B-1 describes the FDDI counter descriptions.

Table B-1: FDDI Counter Descriptions

Counter Name	Description
seconds since last zeroed	The time at which the link entity was created. This value indicates when the associated counter attributes were set to zero.
ANSI MAC frames count	The total number of frames (other than the token frame) seen by this link.
ANSI MAC frame errors	The total number of times the MAC changed the E indicator in a frame from R to S.
ANSI MAC frames lost	The total number of times a frame (other than the token frame) was improperly terminated.
bytes received	The number of bytes successfully received.
bytes sent	The number of bytes successfully transmitted.
data blocks received	The number of frames successfully received.
data blocks sent	The number of frames successfully transmitted.
multicast bytes received	The number of bytes successfully received in multicast frames.
multicast blocks received	The number of frames successfully received in multicast frames.
multicast bytes sent	The number of bytes successfully transmitted in multicast frames.
multicast blocks sent	The number of frames successfully transmitted in multicast frames.

Table B-1: (continued)

Counter Name	Description
transmit underrun errors	The number of times a transmit underrun error occurred. This indicates that the transmit FIFO became empty during frame transmission. This does not include errors in transmitting MAC type frames.
send failures	The number of times a transmit error (other than transmit underrun) occurred.
FCS check failures	The number of times a received frame failed the FCS check.
frame status errors	The number of times a received frame had the E indicator in error but the CRC was correct.
frame alignment errors	The number of times a received frame had an alignment error.
frame length errors	The number of times a received frame had an invalid length, either too long or too short.
unrecognized frames	The number of times a received individually addressed LLC frame was discarded because there was no data link port.
unrecognized multicast frames	The number of times a received LLC frame addressed to a multicast address was discarded because there was no data link port.
receive data overruns	The number of times a frame was discarded because no receive buffer was available.
system buffers unavailable	The number of times a frame was discarded because no link buffer was available.
user buffers unavailable	The number of times a frame was discarded because no user buffer was available.
ring reinitialization received	The number of times a ring reinitialization was initiated by this link.
ring reinitialization initiated	The number of times a ring reinitialization was initiated by some other link.
ring beacon process initiated	The number of times a ring beacon process was initiated was initiated by this link.
duplicate tokens detected	The number of times the MAC detected a duplicate token.
duplicate address test failures	The number of times the duplicate address test failed.
ring purge errors	The number of times the ring purger received a token while purging the ring.
bridge strip errors	The number of times a frame content independent strip operation was terminated by receipt of a token.
traces initiated	The number of times the PC-Trace process was initiated by this link.
traces received	The number of times the PC-Trace process was initiated by some other link.
LEM reject count	The number of times an active connection on this PHY port was disconnected due to rejection by the link error monitor at this end of the physical connection.
LEM events count	The number of raw errors that has been detected by the link error monitor on the physical layer.

Table B-1: (continued)

Counter Name	Description
LCT reject count	The number of times a connection on this PHY port was rejected due to failure of the link confidence test at either end of the physical connection.
TNE expired reject count	The number of times an active connection on this PHY port was disconnected due to rejection by expiration of the noise timer (TNE).
Completed Connection count	The number of times the PHY port entered the "In Use" state, having completed the initialization process.
Elasticity Buffer Errors	The number of times the elasticity buffer function in the PHY had an overflow or underflow.

Table B-2 is an alphabetic (A through N) list of the FDDI status descriptions. Table B-3 contains the remainder of the the FDDI status descriptions in alphabetic order.

Table B-2: FDDI Status Descriptions (A through N)

FDDI Status	Description
Adapter State	The adapter's current state. **Resetting** - After power up or reset process, the adapter is in this state. **Uninitialized** - The adapter is waiting to be initialized. **Initialized** - The adapter has been initialized and waiting to be enabled. **Running** - In this state, the adapter is fully operational, the FDDI link has been initialized and enabled. **Maintenance** - Similar to the running state, except the adapter can only perform loopback operations. **Halted** - Adapter detected an error while operational.
Claim Token Yield	A flag that when set to True indicates that the FDDI MAC entity will yield unconditionally in the claim token process. The FDDI MAC entity sets this flag to True as a safeguard when it believes that its own line address is a duplicate address. If a station with a duplicate address won the claim token process, the ring might never become operational.
Duplicate Address Condition	The result of the Duplicate Address Test performed by the FDDI link entity of the station: **Unknown** - The FDDI link entity is performing the duplicate address test to determine if any other stations on the ring have the same address as the line. **Absent** - The FDDI link entity determined that there is no duplicate of its own line address on the ring.

Table B-2: (continued)

FDDI Status	Description
	Present - The FDDI link entity determined that a duplicate of its own line address exists on the ring. No data can be transmitted or received on the line until this logical ring fault is resolved.
Frame Strip Mode	The frame strip mode used by the station: **SA Match** - The station strips frames from the ring that contain its own address in the source address field. **Bridge Strip** - The station maintains a Sent count of frames sent since obtaining the token, sends a void frame when the transmission is complete (two void frames if it is serving as ring purger), and strips the returning frames from the ring until its Sent count is decremented to zero. Bridge stripping is used by bridges, because they are sensitive to no-owner frames and frequently send frames that do not contain their own address in the source address field.
Last Direct Beacon SA	The Last Direct Beacon Station Address. It will be updated when a Directed Beacon is received.
LED State	The current state of LED for this adapter. **OFF** - The adapter has not been enabled yet. **RED** - The physical connection can not be completed due to a LCT failure or the adapter is in the halted state. **RED_BLINK** - The physical connection can not be completed due to the illegal topology. **GREEN** - The physical connection has been completed. **GREEN_BLINK** - The physical connection is in progress. If LED remains in this state, it may be caused by bad, disconnected, or twisted cable.
Link State	The operational state of the FDDI link entity of the station: **Off Ring Initializing** - The FDDI link entity is initializing and conducting its self-test sequence. **Off Ready** - The FDDI link entity is ready for operation but not yet connected to the logical ring. **On Ring Initializing** - The FDDI link entity is attempting to connect to the logical ring. **On Ring Running** - The FDDI link entity is connected to the logical ring and fully operational. **Off Fault Recovery** - The FDDI link entity is attempting to recover from a logical ring fault such as a failure of the duplicate address test, a local or remote stuck beaconing condition, or ring operational oscillation.
Negotiated TRT	The negotiated value of the token rotation time. This value is referred to as T_Neg in the ANSI FDDI specifications and is negotiated during the claim token process.
Neighbor Physical Port Type	The type of the neighbor physical port, or Unknown if there is no connection yet:

Table B-2: (continued)

FDDI Status	Description
	A - The physical port on a dual attachment wiring concentrator (DAC) or dual attachment station (DAS) that connects to the incoming primary ring and the outgoing secondary ring of the FDDI dual ring. **B** - The physical port on a dual attachment wiring concentrator (DAC) or dual attachment station (DAS) that connects to the outgoing primary ring and the incoming secondary ring of the FDDI dual ring. **Master** - One of the physical ports on a wiring concentrator that connects to a single attachment station (SAS), such as a DEC FDDI controller 700. **Slave** - The physical port on a single attachment station (SAS) that connects to a wiring concentrator or another SAS. **Unknown** - no connection has been established.

Table B-3: FDDI Status Descriptions (P through U)

FDDI Status	Description
Physical Link Error Estimate	The current link error rate as estimated by the link error monitor (LEM). For a value of *n*, the actual rate is 10 to the negative *n*th.
Physical Port State	The operational state of the physical port: **OFF MAINTENANCE** - The physical port is reserved for diagnostic testing and loopbacks. **BROKEN** - The physical port has failed its diagnostic tests and is nonoperational. **OFF READY** - The physical port was disabled. **WAITING** - The physical port is beginning to establish a connection and waiting for a response from its neighbor physical port. **STARTING** - The physical port has received a response from its neighbor physical port and is now exchanging information and performing the link confidence test (LCT) before completing the connection. **FAILED** - Same as WAITING, except that the physical port has failed at least once, by failing the link confidence test (LCT) during initialization, by exceeding the link error monitor (LEM) threshold during operation, or because it is part of an illegal topology. **WATCHING** - Same as STARTING, except that the physical port has failed at least once, by failing the link confidence test (LCT) during initialization, by exceeding the link error monitor (LEM) threshold during operation, or because it is part of an illegal topology.

Table B-3: (continued)

FDDI Status	Description
	IN USE - The physical port has a connection established and is fully operational.
Reject Reason	The reason that the last connection on the physical port was lost. This field is updated every time the physical port loops through the FAILED and WATCHING states:
	No Reason - The physical port is initializing. This value is cleared when the physical port enters the "In Use" state.
	LCT Local - The link confidence test (LCT) failed on this physical port.
	LCT Remote - The link confidence test (LCT) failed on the neighbor physical port.
	LCT Both - The link confidence test (LCT) failed on both this physical port and the neighbor physical port.
	LEM Failure - The bit error rate on the physical port exceeded the link error monitor (LEM) threshold. The LEM monitors the quality of the link during operation.
	Topology Rules - The neighbor physical port is an illegal match for this physical port; for example, an A and an A or a master and a master.
	TNE Expired - The noise timer expired because a single noise event lasted for more than 1.31072 milliseconds. The noise timer is operational only when the physical port is In Use.
	Remote Reject - The neighbor physical port broke the connection for an unknown reason.
	Trace in Progress - The physical port was initializing when a PC trace occurred. When a PC trace occurs, any physical ports that have not yet established a connection are shut down to prevent the topology from changing.
Ring Error Reason	The reason there is an error condition on the ring, or No Reason if the ring is operating correctly:
	No Reason - The ring is operating correctly.
	Initialization Initiated - The FDDI MAC entity of this station initiated the claim token process because it detected a configuration change or a missing token.
	Initialization Received - Another station initiated the claim token process because it detected a configuration change or a missing token.
	Beaconing Initiated - A station initiated the ring beacon process because its TRT timer expired before the claim token process recovered the ring. The beacon process locates the ring break. The station downstream from the break will be stuck beaconing. (A station is stuck beaconing when its FDDI MAC entity has been beaconing longer than the time defined by the ANSI FDDI parameter T_Stuck.)
	Duplicate Address Detected - A station detected a duplicate of its own address.

Table B-3: (continued)

FDDI Status	Description
	Duplicate Token Detected - A station received a token while holding the token. **Purge Error** - The station serving as the ring purger received a token when it was not expecting one. The station expects two void frames and then the token when it is serving as the ring purger. **Bridge Strip Error** - A station using bridge frame stripping received a token before decrementing its Sent count to zero. In bridge strip mode, the station maintains a Sent count of frames sent since obtaining the token, and decrements the count each time one of its frames returns. **OP Oscillation** - The ring is suffering from ring OP (operational) oscillation, where it repeatedly comes up briefly and then goes back into initialization. This problem is frequently caused by a duplicate address condition. **Directed Beacon Received** - A station that is stuck beaconing has sent a frame to the directed beacon multicast address, indicating the suspected cause of the ring break. (A station is stuck beaconing when its FDDI MAC entity has been beaconing longer than the time defined by the ANSI FDDI parameter T_Stuck.) This is the last recovery procedure before initiating the PC trace. **PC Trace Initiated** - A station that is stuck beaconing has forced its upstream neighbors to perform their self tests. (A station is stuck beaconing when its FDDI MAC entity has been beaconing longer than the time defined by the ANSI FDDI parameter T_Stuck.) PC trace is the most drastic fault recovery procedure.
Ring Purge State	The state of the ring purger algorithm of the station's FDDI link entity: **Purger Off** - The ring purger algorithm is not active because the ring is not operational. **Candidate** - The ring is operational and the FDDI link entity is bidding to become the ring purger by sending Candidate Hello frames to the ring purger multicast address. The station with the highest station ID becomes the ring purger. **Non Purger** - The ring is operational and the FDDI link entity is serving as ring purger, constantly purging the ring of fragments and no-owner frames. The station periodically sends Ring Purger Hello frames to the ring purger multicast address.
UNA Timed Out	A flag that when set to True indicates that the FDDI MAC entity has not received an SMT neighborhood information frame (NIF) from its upstream neighbor for 90 seconds.
Upstream Neighbor Address	The 48-bit hardware address of the station that is on the upstream side of the ring from this station.

Table B-4 describes the `fza0` FDDI characteristics.

Table B-4: fza0 FDDI Characteristics

fza0 FDDI Characteristics	Description
Link Address	The 48-bit hardware address of this FDDI network interface.
Firmware Revision	The revision number of firmware in the FDDI adapter.
ROM Revision	The version number of the software stored in Read Only Memory (ROM).
SMT Version ID	The version number of the FDDI Station Management (SMT) protocol used on the station.
Requested TRT	The ANSI MAC parameter T_req, which is the requested value for the token rotation timer.
Maximum TRT	The maximum token rotation time (the ANSI FDDI parameter T_Max) that the FDDI MAC entity will allow to be negotiated in the claim token process. The default value is 173.015 milliseconds.
Valid Transmission Time	The valid transmission time (TVX) used by the FDDI MAC entity. If the FDDI MAC entity does not receive a valid frame or unrestricted token within the valid transmission time, it initializes the ring. The default value is 2.621 milliseconds.
LEM Threshold	The link error monitor (LEM) threshold set for the physical port. The LEM monitors the bit error rate (BER) on the physical port during normal operation. When the bit error rate rises above the LEM threshold, the station disables the physical port, preventing it from disrupting the ring. The LEM threshold is expressed as the absolute value of the exponent of the bit error rate. The legal range for the threshold is 5 through 8, corresponding to the range of bit error rates, which is 10 to the -5th (0.00001) bit errors per second through 10 to the -8th (0.00000001) bit errors per second.
PMD Type	The type of physical medium to which this physical port is attached: **Multi Mode** - Inexpensive thick core fiber combined with light emitting diode (LED) sources and p-type intrinsic n-type (PIN) detectors. **Single Mode** - Expensive thin core fiber combined with laser diode sources and avalance photo diode (APD) detectors.

B.3 Using the Packetfilter with FDDI

This section contains information about using the packetfilter with FDDI interfaces.

Incoming packets are treated as described in the `packetfilter`(4) reference page, which states: "The packet filter treats the entire packet, including headers, as uninterpreted data."

Outgoing packets are treated as raw FDDI packets; no encapsulation is performed. The system does set the FDDI source address on outgoing packets.

Applications sending FDDI packets via the packet filter must provide a full FDDI header on output buffers. Received FDDI packets will include the full FDDI header. The FDDI header is defined in `<netinet/if_fddi.h>`. Note the use of a three-byte padding field, `fddi_ph`, whose contents are ignored, but which will be present at the front of all received FDDI buffers and which must be present at the front of all transmitted buffers.

In previous versions of ULTRIX, programs such as `tcpdump` and `nfswatch` were able to monitor FDDI networks, due to an incorrect implementation of the packet filter support for FDDI. These programs are unsupported and have not been updated in the current release. Therefore, `tcpdump` and `nfswatch` no longer support FDDI. Only those programs that use EIOCDEVP to distinguish between interface types can support FDDI interfaces. The new interface type ENDT_FDDI is defined in `<net/pfilt.h>`.

Because `tcpdump` and `nfswatch` are public-domain programs, you may obtain FDDI-capable versions for use on ULTRIX Version 4.3 by using anonymous FTP from gatekeeper.dec.com. For more information, obtain the file `/pub/DEC/ultrix-faq.txt` from gatekeeper.dec.com.

How to Order Additional Documentation

Technical Support

If you need help deciding which documentation best meets your needs, call 800-343-4040 before placing your electronic, telephone, or direct mail order.

Electronic Orders

To place an order at the Electronic Store, dial 800-234-1998 using a 1200- or 2400-bps modem from anywhere in the USA, Canada, or Puerto Rico. If you need assistance using the Electronic Store, call 800-DIGITAL (800-344-4825).

Telephone and Direct Mail Orders

Your Location	Call	Contact
Continental USA, Alaska, or Hawaii	800-DIGITAL	Digital Equipment Corporation P.O. Box CS2008 Nashua, New Hampshire 03061
Puerto Rico	809-754-7575	Local Digital subsidiary
Canada	800-267-6215	Digital Equipment of Canada Attn: DECdirect Operations KAO2/2 P.O. Box 13000 100 Herzberg Road Kanata, Ontario, Canada K2K 2A6
International	————	Local Digital subsidiary or approved distributor
Internal[a]	————	SSB Order Processing - WMO/E15 *or* Software Supply Business Digital Equipment Corporation Westminster, Massachusetts 01473

[a] For internal orders, you must submit an Internal Software Order Form (EN-01740-07).

Reader's Comments

ULTRIX and UWS Version 4.3
Release Notes
AA-ME85H-TE

Please use this postage-paid form to comment on this manual. If you require a written reply to a software problem and are eligible to receive one under Software Performance Report (SPR) service, submit your comments on an SPR form.

Thank you for your assistance.

Please rate this manual:	Excellent	Good	Fair	Poor
Accuracy (software works as manual says)	☐	☐	☐	☐
Completeness (enough information)	☐	☐	☐	☐
Clarity (easy to understand)	☐	☐	☐	☐
Organization (structure of subject matter)	☐	☐	☐	☐
Figures (useful)	☐	☐	☐	☐
Examples (useful)	☐	☐	☐	☐
Index (ability to find topic)	☐	☐	☐	☐
Page layout (easy to find information)	☐	☐	☐	☐

What would you like to see more/less of? _____

_____

_____

What do you like best about this manual? _____

_____

_____

What do you like least about this manual? _____

_____

_____

Please list errors you have found in this manual:

Page	Description
_____	_____
_____	_____
_____	_____
_____	_____
_____	_____

Additional comments or suggestions to improve this manual:

_____

_____

_____

_____

What version of the software described by this manual are you using? _____

Name/Title _____ Dept. _____

Company _____ Date _____

Mailing Address _____

_____ Email _____ Phone _____

Do Not Tear – Fold Here and Tape

digital™

NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST–CLASS MAIL PERMIT NO. 33 MAYNARD MA

POSTAGE WILL BE PAID BY ADDRESSEE

DIGITAL EQUIPMENT CORPORATION
OPEN SOFTWARE PUBLICATIONS MANAGER
ZKO3–3/Y32
110 SPIT BROOK ROAD
NASHUA NH 03062–9987

Do Not Tear – Fold Here

Cut
Along
Dotted
Line