

**ULTRIX
Release Notes
for RISC Processors**

Order No. AA-NE12A-TE

ULTRIX-32 Operating System, Version 3.0

ULTRIX Worksystem Software, Version 2.0

Digital Equipment Corporation


Copyright © 1989 Digital Equipment Corporation
All rights reserved.

The information in this document is subject to change without notice and should not be construed as a commitment by Digital Equipment Corporation. Digital Equipment Corporation assumes no responsibility for any errors that may appear in this document.

The software described in this document is furnished under a license and may be used or copied only in accordance with the terms of such license.

No responsibility is assumed for the use or reliability of software on equipment that is not supplied by DIGITAL or its affiliated companies.

The following are trademarks of Digital Equipment Corporation:

DEC	ULTRIX	VMS
DECnet	ULTRIX-11	VT
DECUS	ULTRIX-32	XUI
DECwindows	VAX	ULTRIX Worksystem Software
MicroVAX	VAXstation	

UNIX is a registered trademark of AT&T in the USA and other countries.

PostScript is a registered trademark of Adobe Systems, Inc.

X Window System is a trademark of MIT.

This manual was written and produced by the ULTRIX Documentation Group in Nashua, New Hampshire.

Contents

About This Manual

Audience	ix
Organization	ix
Conventions	x

1 System Setup Notes

1.1 Installation Notes	1-1
1.1.1 Changes to the Advanced Installation	1-1
1.1.1.1 Selecting Swap Partitions	1-2
1.1.1.2 Unsupported Subsets	1-2
1.1.1.3 System Disk Partition Tables	1-2
1.1.2 Structure of Distribution Media	1-3
1.1.3 Making Special Files After an Installation	1-3
1.1.4 ULTRIX Boot Path Restriction	1-3
1.1.5 Failure During Tape Installation	1-3
1.1.6 Correcting Nonexistent Symbolic Links	1-4
1.1.7 Formatting Disks	1-4
1.1.8 Identifying Devices	1-4
1.2 Remote Installation Service Notes	1-4
1.2.1 Changes to the Network Installation	1-4
1.2.2 Duplicate Copies of /etc/mop_mom on Startup	1-5
1.2.3 Registering Clients with BIND Hostnames	1-5

1.3 Diskless Management Services Notes	1-5
1.3.1 Diskless Management Services Installation Notes	1-6
1.3.2 Different DMS Commands	1-6
1.3.3 The /etc/exports File	1-7
1.3.4 DMS Mount Point Change	1-7
1.3.5 Creating Links for the /usr/hosts Directory	1-7
1.3.6 Changes in Setting Up Disk Environments	1-7
1.3.7 Timezone Setting for Diskless Clients	1-8
1.3.8 Diskless Client Directory	1-8
1.3.9 Sufficient Disk Space	1-8
1.3.10 VAX Diskless Clients	1-8
1.3.10.1 Changing the swap Area	1-8
1.3.10.2 Loading the MOP Subset	1-9
1.3.11 Adding and Modifying Clients	1-9
1.3.12 Kernel Build Failure	1-9
1.4 System Configuration Notes	1-9
1.4.1 Increasing Your Swap Space	1-10
1.4.2 Printer Setup	1-10
1.4.3 Spool Directories for Remote Printers	1-11
1.4.4 Shared Modem Lines	1-11
1.4.5 Reactivating Hardwire Terminals	1-11
1.4.6 Unterminated Terminal Lines	1-12
1.4.7 System Exercisers	1-12
1.4.8 Eight-Bit Terminal Driver Support	1-12
1.4.9 PostScript Support Requires Adobe Font Metrics Files	1-13
1.4.10 Relocating awk and sed Binaries	1-13
1.4.11 The /etc/svcorder File for YP and BIND	1-14
1.4.12 Configuring Your Network with YP or BIND	1-14

2 ULTRIX Worksystem Software Notes

2.1 Known Server Problems	2-1
2.2 DECwindows Applications	2-2
2.2.1 The dxmail Mail Handler	2-3
2.2.2 The dxterm Terminal Emulator	2-3

2.2.3	The dxcalendar Program	2-6
2.2.4	The dxcalc Calculator	2-6
2.2.5	The dxdb Debugger	2-6
2.2.6	The dxsession Session Manager	2-7
2.2.7	The dxvdoc Program	2-9
2.2.8	The dxdiff Visual Differences Program	2-9
2.2.9	The dxue User Executive	2-10
2.2.10	The dxnotepad Program	2-11
2.2.11	The dxpaint Program	2-12
2.2.12	The dxpsview PostScript Previewer	2-12
2.2.12.1	Known Problems	2-12
2.2.12.2	Application Resource Names	2-14
2.2.12.3	Differences Between the Previewer and an LPS40 Printer	2-14
2.2.13	The dxwm Window Manager	2-14
2.2.13.1	Naming Windows and Icons	2-15
2.2.13.2	Handling Mapping Requests	2-16
2.3	Demonstration Programs	2-17
2.4	X11 R3 Applications	2-17
2.4.1	The uwm Window Manager	2-18
2.4.2	The xbiff Command	2-18
2.4.3	The xterm Terminal Emulator	2-18
2.4.4	Changes to .Xdefaults from MIT X10	2-20
2.4.5	Command Line Syntax Changes	2-21
2.4.6	Unsupported R3 Clients	2-21
2.5	DECwindows Fonts	2-21
2.5.1	Font File Names	2-22
2.5.2	Access to Font Names	2-22
2.5.3	Understanding Font Names	2-23
2.5.4	Font Properties	2-24
2.5.5	Font Problems	2-25
2.6	The Xlib Library	2-25
2.7	The XUI Toolkit	2-27

3 ULTRIX-32 Operating System Notes

3.1 Differences Between RISC-Based and VAX-Based Systems	3-1
3.1.1 Components Not in RISC-Based Systems	3-1
3.1.2 Components that Differ	3-3
3.1.3 Components on RISC-Based Systems Only	3-9
3.1.4 Unsupported Components on RISC-Based Systems	3-10
3.2 Hints for Porting Software to RISC-Based Systems	3-10
3.3 Compliance with Operating System Standards	3-12
3.4 Network File System	3-12
3.4.1 NFS File and Record Locking	3-12
3.4.2 NFS Server Security Changes	3-12
3.5 Programming Notes	3-13
3.6 Libraries	3-13
3.6.1 C Library	3-13
3.6.2 Math Library	3-14
3.7 Compiler Notes	3-14
3.8 Commands and Utilities	3-16
3.8.1 The ar Command	3-17
3.8.2 The lint Command	3-17
3.8.3 The lprsetup Command	3-18
3.8.4 The lpr Command	3-18
3.8.5 The cp Command	3-18
3.8.6 The showmount Command	3-19
3.8.7 Regular Expression Handling	3-19
3.8.8 The mail Command	3-19
3.8.8.1 Alternate Forwarding File	3-19
3.8.8.2 Daemon Start Order	3-20
3.8.8.3 Delimiters for Aliasing	3-20
3.8.8.4 Invoking mail More Than Once	3-20

3.8.9	The sh Command	3-20
3.8.10	The newcsh Command	3-21
3.8.11	The chfn Command	3-21
3.8.12	The cpio Command	3-21
3.8.13	The csh Command	3-21
3.8.14	The dc Command	3-21
3.8.15	The ltf Command	3-22
3.8.16	The make Command	3-22
3.8.17	The pstat Command	3-23
3.8.18	The rdump Command	3-23
3.8.19	The script Command	3-23
3.8.20	The setld Command	3-23
3.8.20.1	Removing Subsets	3-23
3.8.20.2	Installing Subsets from Two Installation Devices	3-23
3.8.20.3	Installing Subsets	3-24
3.8.21	The size Command	3-24
3.8.22	The talk Command	3-24
3.8.22.1	The talk Command Is Not 8-bit Clean	3-24
3.8.22.2	Changes in the talk Program	3-25
3.8.23	The uucp Command	3-25
3.8.24	The uucpsetup Command	3-25
3.8.25	The vi Command	3-25
3.8.26	The yppasswd Command	3-26
3.9	System Facilities	3-26
3.9.1	Errors in the /etc/passwd File	3-26
3.9.2	The ptrace System Call	3-27
3.9.3	The sendmail Program	3-27
3.9.4	Writing into a Remote a.out File	3-27
3.10	Networking and Communications	3-28
3.10.1	DLI/802 Passes Up Packets That Should Be Dropped	3-28
3.10.2	DLI/Ethernet Drivers	3-28
3.10.3	Incorrect UUCP Checksum Calculation with F Protocol	3-28
3.10.4	Performance Improvement for YP and BIND	3-29
3.10.5	Manual Update of BIND server's Serial Number	3-29

3.10.6 Use of Full Host Names When Running BIND	3-29
3.10.7 NFS Server Security Changes	3-30
3.10.8 Blank Lines in Yellow Pages Map Files	3-30
3.11 Standards and External Specifications	3-30
3.11.1 POSIX Interface Support	3-30
3.11.2 System Calls in the POSIX Environment	3-31
3.12 Miscellaneous Known Problems	3-31

4 ULTRIX Documentation Notes

4.1 Revised and New Documents	4-1
4.2 Reference Pages	4-2
4.3 Undocumented Library Functions	4-3

About This Manual

These notes apply to the ULTRIX Worksystem Software, Version 2.0 (RISC) and ULTRIX-32 Operating System, Version 3.0 (RISC) software kits. The ULTRIX Worksystem Software, Version 2.0 (RISC) kit is intended for RISC-based workstations and contains two components:

- Graphics windowing software
- Base operating system software

The ULTRIX-32 Operating System, Version 3.0 (RISC) kit is intended for RISC-based server systems and contains only the base operating system software.

Audience

This manual should be read by all customers who have purchased an ULTRIX Worksystem Software, Version 2.0 (RISC) or ULTRIX-32 Operating System, Version 3.0 (RISC) system.

Organization

- Chapter 1 System Setup Notes
- Contains the notes that you should read before installing your system, setting it up as a remote installation (ris) or a diskless management services (dms) server, or configuring (building) your system kernel.
- Chapter 2 ULTRIX Worksystem Software Notes
- Contains the notes that pertain to ULTRIX Worksystem Software, which provides the graphics windowing software. If you have an ULTRIX Worksystem Software, Version 2.0 (RISC) system, you should read the notes in this chapter.

- Chapter 3 **ULTRIX-32 Operating System Notes**
Contains the notes that pertain to the ULTRIX-32 Operating System, which provides the base operating system software. If you have an ULTRIX Worksystem Software, Version 2.0 (RISC) or ULTRIX-32 Operating System, Version 3.0 (RISC) system, you should read the notes in this chapter.
- Chapter 4 **ULTRIX Documentation Notes**
Contains the notes that pertain to the ULTRIX documentation that shipped with your ULTRIX Worksystem Software, Version 2.0 (RISC) or ULTRIX-32 Operating System, Version 3.0 (RISC) system.

Conventions

The following conventions are used in this manual:

- special** In text, each mention of a specific command, option, partition, pathname, directory, or file is presented in this type.
- name(n)** References to ULTRIX commands, system calls, or subroutines include the section numbers in the *ULTRIX Reference Pages* or *ULTRIX Worksystem Software Reference Pages* where they are documented.
- italics* In syntax and function definition descriptions, this type indicates terms that are variable.
- UPPERCASE** The ULTRIX system differentiates between lowercase and uppercase characters. In examples, enter uppercase characters only where they are specifically indicated by an example or a syntax line.
- example** In examples, system output is printed in this type.
- example** In examples, user input is printed in this boldface type.

System Setup Notes 1

The notes in this chapter provide information about installing and setting up your ULTRIX Worksystem Software, Version 2.0 (RISC) or ULTRIX-32, Version 3.0 (RISC) system. You should read this chapter in its entirety before you install your software distribution kit. Then, depending on your system, you should read the relevant notes in the remaining chapters.

This chapter discusses:

- Installation notes
- Remote installation service notes
- Diskless management services notes
- System configuration notes

1.1 Installation Notes

The notes in this section pertain to the installation procedure. This section discusses:

- Changes to the advanced installation
- Structure of the distribution media
- Making special files after the installation
- ULTRIX boot path restriction
- Failure during tape installation
- Correcting nonexistent symbolic links
- Formatting disks
- Identifying devices

1.1.1 Changes to the Advanced Installation

The next three notes are specific to the advanced installation procedure:

- Selecting swap partitions
- Unsupported software subsets
- System disk partition tables

1.1.1.1 Selecting Swap Partitions – Avoid selecting the a partition of any disk for use as a swap partition. If a customized partition table has previously been defined for a disk and swapping occurs to the a partitions, the disk partition table, which resides on the a partition, will be destroyed. For information about increasing swap space, see Section 4.1.

1.1.1.2 Unsupported Subsets – To install desired unsupported subsets from tape or CDROM distribution after you have completed the installation, use the `/etc/setld` command. For further information, see `setld(8)` in the *ULTRIX Reference Pages*.

To install subsets from tape, load the tape labeled UNSUPPORTED. To install subsets from your CDROM distribution, mount the c partition of the CDROM and load subsets from `/mnt/RISC_UNSUPPORTED` or `/mnt/VAX_UNSUPPORTED`, as is appropriate.

1.1.1.3 System Disk Partition Tables – If you have modified the partition table of the system disk, the advanced installation issues a prompt asking if you want to use the default partition structuring or the customized partitioning. The basic criteria for allowing a customized partition structure are that the root or a partition must be at least 32768 blocks and that the swap or b partition must be at least 12288 blocks. (This assumes that you plan to add a second paging device.) The installation does not check to guarantee the size of any other partitions on this disk. Therefore, you are responsible for selecting a suitable location and size for the `/usr` file system.

During the advanced installation, no other disks on the system are modified. Whatever partition tables might exist are used; otherwise, default partition structuring is assumed.

During a basic installation, however, target system disks are set to have default partitions. No other disks on the system are modified during a basic installation.

1.1.2 Structure of Distribution Media

The subsets on the distribution media are an ordered set of tar images starting at the fifth file on tape. These images are compressed (with the `compress(1)` utility). One benefit of doing this is that less space will be required to hold the subsets when you set up a RIS area for network installations.

Note that the `setld` product environments on the CDROM distribution are structured hierarchically.

1.1.3 Making Special Files After an Installation

During an installation to disk, only the special files for the disk and tape devices actually used are created. To make special files for other devices, follow this procedure once the installation procedure has completed.

```
% su
Password:
# cd /dev
# MAKEDEV device...
```

The device name specified should be the name of the peripheral as it is displayed during auto-configuration when your system boots.

1.1.4 ULTRIX Boot Path Restriction

If an attempt to boot a kernel image from disk fails because the image does not exist, the `ultrixboot` program prompts for the kernel image name as follows:

```
Enter image name:
```

With this release, you must enter the following path using the same syntax used at the console prompt:

```
rz(0,0,0)vmunix.new
```

The `ultrixboot` program is currently unable to load a kernel image from a disk other than the actual booted device.

1.1.5 Failure During Tape Installation

During a TK50 tape installation, you must wait until the tape completely rewinds before answering any questions presented by the Standalone ULTRIX Environment. This is necessary to prevent the installation software from accessing the tape device while it is still rewinding after the bootstrap operation.

1.1.6 Correcting Nonexistent Symbolic Links

After performing an installation, some symbolic links point to nonexistent files. To correct this, issue the following commands:

```
% su
Password:
# rm /etc/rdump /etc/rrestore /usr/bin/make
# ln -s ../bin/rdump /etc/rdump
# ln -s ../bin/rrestore /etc/rrestore
# ln -s s5make /usr/bin/make
```

Note that correcting this situation is not essential to normal operation.

1.1.7 Formatting Disks

The *Basic Installation Guide for the DECstation 3100* states that you must format and prepare your disks. Because your disks are preformatted and ready for use, you should ignore this step.

1.1.8 Identifying Devices

During an installation, you are instructed to use the `test` command to identify the devices that are connect to your system. To ensure that your system will boot successfully, you should use the following commands to identify your devices:

```
% test -c
% init
```

1.2 Remote Installation Service Notes

The notes in this section pertain to a Remote Installation Service (RIS) installation. This section discusses:

- Changes to the network installation
- Duplicate copies of `/etc/mop_mom` on startup
- Registering clients with BIND hostnames

1.2.1 Changes to the Network Installation

The ULTRIX Worksystem Software, Version 2.0 (RISC) kit installation offers the option of doing an advanced installation procedure from the network. Only those subsets beginning with UDT and UDW in a client's model file will be installed during the system installation. You can install other subsets from the network at a later time by using the `/etc/setld` command.

The basic installation procedure from the network only installs the mandatory UDT and UDW subsets.

1.2.2 Duplicate Copies of /etc/mop_mom on Startup

The `ris` utility attempts to modify the `/etc/rc.local` file to start the `mop_mom` daemon on system startup. It does this by searching for a line in `/etc/rc.local` that begins with `/etc/mop_mom`. If no such line is found, one is created. This can cause duplicate invocations of the daemon if the `/etc/rc.local` has been modified to start the `mop_mom` daemon in a way that is not recognized by `ris`.

If the `/etc/rc.local` file must be modified to start `mop_mom` in a way not understood by this utility, you must add the following string to the `/etc/rc.local` file to prevent duplicate invocations:

```
/etc/mop_mom_risfix
```

This has no effect on system startup and effectively indicates to `ris` that the `mop_mom` daemon has already been set up correctly.

1.2.3 Registering Clients with BIND Hostnames

The `ris` utility does not correctly register clients with full BIND specified hostnames in the mop databases. This will cause a boot failure for these clients when attempting to perform the network installation. If you encounter this problem, make the following change:

```
% su
Password:
# cd /usr/var/dnet/clients
# mv full.hostname.sys full.hostname
```

The `full.hostname` is the fully specified hostname for the client that is to be booted.

1.3 Diskless Management Services Notes

The notes in this section pertain to a diskless environment. This section discusses:

- Diskless management services installation notes
- Different DMS commands
- The `/etc/exports` file
- DMS mount point change
- Creating links for the `/usr/hosts` directory

- Changes in setting up disk environments
- Timezone setting for diskless clients
- Diskless client directory
- Sufficient disk space
- VAX diskless clients
- Adding and modifying clients
- Kernel build failure

1.3.1 Diskless Management Services Installation Notes

Diskless management services (DMS) has undergone a number of changes that affect the way clients are added. For example, only one GENERIC kernel is distributed for use in running clients added to the environment. Kernels are no longer built on the server system. This means that at least one client in the environment must be registered to build a customized kernel. Regardless of the client's swap requirements, a temporary (in the case of choosing a local swap device) network swap file is created to support this activity. After adding this client, you should boot the RISC-based diskless system using the following command:

```
>> setenv bootpath mop()
>> auto
```

When the client system boots, it will automatically build the kernel as described on the server system. When the build completes, you are instructed to set the boot environment at the console and to reboot using commands displayed.

After the first client has built a customized environment, you can add other systems that are to be considered clones of the first by using the "duplicate sub-option". For now, only the /dev directory and the /vmunix and /etc/hosts file and /sys hierarchy are duplicated from the first environment. The rest of the client's root environment is installed as in the past from the selected "dlenv?/root?.mips" environment.

1.3.2 Different DMS Commands

For ULTRIX Worksystem Software, Version 2.0 (RISC) and ULTRIX-32, Version 3.0 (RISC), DMS has been significantly changed to support a non-VAX architecture. As a consequence, the /etc/dms.new command has been provided with the RISC-based server kit to serve other RISC-based clients. You should use the appropriate DMS command when managing your RISC-based and VAX-based clients: /etc/dms.new for RISC-based system and /etc/dms for existing VAX-based system.

1.3.3 The /etc/exports File

Do not change the entries in the /etc/exports file by replacing blanks with tabs or breaking up long lines with a return. The dms utility places its entries in the /etc/exports file in a format that it can recognize when it reaccesses the file. Failure to follow these guidelines can cause problems with the mount daemon and with the dms utility's ability to read the file properly.

1.3.4 DMS Mount Point Change

The mount point for dlenv? and dlclient? have changed. The new mount point are /dlenv? and /dlclient?, not /var/diskless/dlenv? and /var/diskless/dlclient?. The /etc/fstab and /etc/exports files reflect this change. This note pertains to /etc/dms.new only.

1.3.5 Creating Links for the /usr/hosts Directory

On the diskless client, the /usr/hosts directory does not contain any links that will allow you to log in to another system by entering that system's name as a command.

To create the links, type the following command:

```
% /usr/hosts/MAKEHOSTS /dlenv?/root?.mips/usr/hosts
```

1.3.6 Changes in Setting Up Disk Environments

The DMS utility no longer requires dedicated disks for the software or client environments when setting up a diskless area. You are now allowed to use individual partitions for diskless areas. You must take care to set up the partitions of the disk they will be using. That is, you must take care that the partitions chosen are large enough to complete the procedure successfully when installing software or to hold the total number of clients that will be registered. Repartitioning of the disk is no longer performed through the dms utility. You must set up the disk partitions before using them.

These changes offer more installation flexibility by not requiring an entire disk for diskless environments and allow multiple environments on different partitions within the same disk. The new changes also allow you to add additional diskless environments at any time.

Before invoking the dms utility, you should read the *Guide to Diskless Management Services*. In particular, you should read Chapter 3, which discusses the new procedure for selecting disks and partitions.

1.3.7 Timezone Setting for Diskless Clients

When you add a diskless client, its timezone setting is extracted from the server's config file. The `dms` utility depends on the presence of a properly constructed configuration file on the server system as was created during the installation acting of the server.

1.3.8 Diskless Client Directory

A diskless client directory (`/dlclient?`) must exist before you install any software subsets. You must use the appropriate `dms` option to configure a client partition on your server system; otherwise, when you install the second set of subsets, `dms` will fail.

1.3.9 Sufficient Disk Space

It is important that you carefully calculate disk requirements for diskless client areas by using the documentation provided with your server kit. If you do not have enough disk space and you attempt to add a diskless client, the `dms.new` utility will fail to register the client. In addition, the remove client function also will not work. In this case, you will have to remove the `/dlclient?/client_name.root` directory manually.

1.3.10 VAX Diskless Clients

The next two notes are specific to serving VAX diskless clients:

- Changing the swap area
- Loading the subset

1.3.10.1 Changing the swap Area – If you execute the `dms Modify Client Parameters` option to change the swap area from `SERVER` to `LOCAL` or `LOCAL` to `SERVER`, the system will display warning messages. On rebooting VAX diskless clients, you should ignore any of the following messages that may appear:

```
mkdir : /sys/SYSTEMNAME : File Exists
mv: cannot access dev/ttypf
mv: cannot access dev/ptypf
dev/X0 : File Exists
```

1.3.10.2 Loading the MOP Subset – When a RISC machine serves VAX diskless clients, the MOP subset should be loaded on the server and installed into all VAX client areas. Otherwise, the diskless client cannot build a netblk executable file that contains the required network information, and the following message will be displayed:

```
Cannot find SAS/mop.h
```

In addition, if the diskless client swaps on local disk, the kernel will not boot.

1.3.11 Adding and Modifying Clients

When adding clients by using the dms duplicate suboption, you should follow the instructions to boot the client before attempting to modify the same client.

1.3.12 Kernel Build Failure

A difference in the time of day between the server and a client system (more specifically, when the server's date and time lags the client's) can cause the kernel build to fail. This discrepancy causes an obscure problem with makefile dependencies to appear in the form of a "syntax error" when diskless client systems attempt to build their kernel.

A workaround for this problem is as follows:

1. Either temporarily adjust the time on the server so that it is ahead of the client system, or set the date and time on the client system so that it lags the server's date and time.
2. Reboot the client system and allow it to complete the kernel build.

1.4 System Configuration Notes

The notes in this section pertain to your system configuration. This section discusses:

- Increasing your swap space
- Printer setup
- Spool directories for remote printers
- Shared modem lines
- Reactivating hardware terminals
- Unterminated terminal lines
- System exercisers

- Eight-bit terminal driver support
- PostScript printer support
- Relocating awk and sed binaries
- The /etc/svcorder file for YP and BIND
- Configuring your network with YP or BIND

1.4.1 Increasing Your Swap Space

If you increase your swap space beyond the default 60 Mbytes by adding swap partitions or using a larger swap partition and you plan to run processes with very large data segments (that is, greater than 85 Mbytes), you must edit your system configuration file and increase the system `dmmax` parameter accordingly. The syntax of the `dmmax` parameter is:

```
dmmax n
```

`n` Specifies the number of 512-byte disk blocks in a swap fragment. (The default is 4096.)

The following table provides guidelines for making the appropriate change:

Configured Swap	Recommended Value
60 Mbytes (default)	4096 (default)
0 – 85 Mbytes	4096
86 – 170 Mbytes	8192
171 – 340 Mbytes	16384

For further information, see the *Guide to System Configuration File Maintenance for RISC Processors*.

1.4.2 Printer Setup

The default size limit of a file to be printed is 1025024 bytes (local or remote). As superuser, you can modify this default by editing the `/etc/printcap` file on the local system where you issue the `lpr` command and inserting the `mx#0` or `mx#nn` parameter. If `mx#0` is specified, the default file size is unlimited. If `mx#nn` is specified, the default file size is the specified number of 1024-byte blocks.

Once set, this new limit applies both to local print jobs and to those print jobs sent to remote systems. Note that this limit on the local system does not affect print requests sent from remote systems.

As a nonprivileged user, you can workaroud the file size limit by dividing the file into sections less than 1025024 bytes before printing (assuming that the file data format allows for this). Or, you can use the `rcp` command to copy the file to a system with a printer that has a sufficiently large file size limit. Log in on that system and then request local printing.

1.4.3 Spool Directories for Remote Printers

If you specify a `printcap` entry for a remote printer, you must also create a spool directory for that printer, just as you would for a local printer. For example, if you have an LA100 printer with a `printcap` entry of `:sd=/usr/spool/la100`, then you should type the following commands to create its spool directory with the proper owner:

```
% su
Password:
# mkdir /usr/spool/la100
# chown daemon /usr/spool/la100
```

1.4.4 Shared Modem Lines

The use of shared lines requires modem control. Carrier detection must be raised on 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.

1.4.5 Reactivating Hardwire 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 temporarily reactivate the terminal line, follow these steps:

1. Determine the process identification (PID) of the `getty` process associated with the terminal line with the `ps` command.
2. Use the `kill` command to kill that process. If you do not know the number of the hung terminal line, use the `last(1)` command with the user's login name.

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

```

% su
Password:
# last anders
anders  tty01      Mon Nov 18 10:00   still logged in
anders  tty01      Mon Nov 18 08:35 - 09:08   (00:33)
anders  tty01      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 00 I    0:00 - 2 tty00 (getty)
  160 01 I    0:00 - 2 tty01 (getty)

# kill -9 160

```

1.4.6 Unterminated Terminal Lines

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 will be repeated continuously and will cause system performance to degrade. To prevent this, terminals should be left powered on at all times. If the line is not used, it should be specified as `off` in the `/etc/ttys` file to prevent a `getty` process from being started on the line.

1.4.7 System Exercisers

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`.

1.4.8 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. For instructions to enable logins on terminal lines which require eight-bit characters, see `gettytab(5)` in the *ULTRIX Reference Pages*. The `p8` and `pd` flags have been added to `gettytab` to facilitate the use of eight-bit characters.

DEC VT100 series terminals are capable of displaying only the lower half of the DEC Multinational Character Set. Standard 7-bit ASCII characters are included in the lower half of the Multinational Character Set.

DEC VT200-series and VT300-series terminals are capable of displaying the full DEC Multinational Character Set. 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:

- Call up the terminal's setup menu by pressing the Setup (F3) key. Select the **General** menu option.
- Move to the field that allows you to select the terminal mode. Select the option **VT200 mode 7 bit controls**.
- Select the **To Directory** option to return you to the top level.
- Call up the **Comm** menu. Select the **8 bits no parity** option.

The DECwindows terminal emulator, `dxterm(1X)`, can also be set up for use with eight-bit characters. In this case, the terminal mode must be set to VT300 mode.

1.4.9 PostScript Support Requires Adobe Font Metrics Files

A new optional software subset, ULTAFM030, installs the Adobe Font Metrics (AFM) files in the `/usr/lib/font/metrics` directory. The subset contains .57 AFM files and one PostScript procedure, `encodefont.ps`, which requires approximately 885K of disk space.

The AFM files contain the font metrics (character bounding box, width, name, ligature, kerning and font properties) for the PostScript outline fonts used in all Digital PostScript output devices (hardcopy and video). The information contained in these files is essential for the proper composition of text by any application or system that needs to print to these devices.

Almost all the files are present in two forms of character encoding, namely Adobe and ISO Latin 1. The PostScript procedure allows applications to switch between the two forms of character encoding.

1.4.10 Relocating awk and sed Binaries

The `awk` and `sed` binary files have been moved from the root file system to `/usr`. If you use `/opr/backup`, you need these files to be located in `/bin`. To replace the `awk` and `sed` links with actual files in `/bin`, follow this procedure:

1. Login as root (superuser):

```
% su
Password:
```

2. Change directories to /bin:

```
# cd /bin
```

3. Check to see that awk and sed are links:

```
# ls -al awk sed
lrwxr-xr-x 1 root 14 Mar 23 1988 awk@ -> ../usr/bin/awk
lrwxr-xr-x 1 root 14 Mar 23 1988 sed@ -> ../usr/bin/sed
```

4. Remove the links:

```
# rm awk sed
```

5. Copy the binaries for awk and sed to /bin:

```
# cp /usr/bin/awk /usr/bin/sed /bin
```

1.4.11 The /etc/svcorder File for YP and BIND

When you select a host name and address resolution service, such as Yellow Pages (YP) or Berkeley Internet Name Domain (BIND), the setup scripts create a file that is new with ULTRIX-32 Operating System, the /etc/svcorder file.

The gethostent interface checks for the existence of /etc/svcorder. If it does not exist, the local /etc/hosts file is used to resolve host names and addresses. If the file does exist, however, the gethostent interface queries each service sequentially listed in /etc/svcorder to resolve a request.

If you plan on setting up YP or BIND manually, it is important to create the /etc/svcorder file in order to use that name service. The fact that the YP and BIND daemons may be running does not mean that you are using that service.

For further information, see svcorder(5) in the *ULTRIX Reference Pages*.

1.4.12 Configuring Your Network with YP or BIND

It is imperative to booting that you have localhost and the full host name (with domain name if running Berkeley Internet Name Domain (BIND)) and host address defined in the local /etc/hosts file, and the local entry in /etc/svcorder. This file entry is a necessity when booting because the Yellow Pages (YP) and BIND daemons may not been started yet.

The following is an example of a valid `/etc/hosts` file:

```
127.0.0.1 localhost
128.1.1.2 student.harvard.edu student
```

In this example, `student` is the local host name in the BIND domain `harvard.edu`.

The `localhost` and host name entries should also be in the YP and BIND database.

ULTRIX Worksystem Software Notes 2

The notes in this chapter pertain specifically to the graphics windowing software that is part of the ULTRIX Worksystem Software, Version 2.0 (RISC) kit. Note that full DECwindows functionality is not available for ULTRIX Worksystem Software, Version 2.0 (RISC). This chapter discusses:

- Known server problems
- DECwindows applications
- Demonstration programs
- X11 R3 applications
- DECwindows fonts
- The Xlib library
- The XUI Toolkit

2.1 Known Server Problems

- All server messages are logged in the `/usr/adm/X?msgs` file, where `?` indicates the number of your display.
- When you are not running the session manager, console messages are displayed randomly on the screen. This may interfere with graphics displayed by the X server. Therefore, you should redirect console messages to a terminal emulator window. Note that you do not need to do this if you are running the session manager.

To determine the terminal emulator window in which you would like console messages displayed, follow this procedure:

1. Find out the tty number of the terminal emulator window by entering the `tty` command.
2. Become superuser.
3. Enter the following command:

```
# /usr/bin/xcons 1 tty ? xcons &
```

Note that if you are running the `dxsession` session manager, you should specify a terminal other than `ttyv0` or `ttypf`. Console messages should now be directed to that terminal emulator window. However, console messages are redirected for the current session only.

- You cannot set the volume, the keyboard click, or bell to zero.
- On diskless systems, the lack of `/tmp` can inhibit the X server from starting. You should make sure that you have created `/tmp` in your diskless root area.

The following notes apply to all servers:

- A new option has been added for all servers. Specify `-dpi 100` for a screen density of 100 dots per inch (dpi) or `-dpi 75` for a screen density of 75 dpi. The default is 75 dpi.
- Some workstations do not have writable default colormaps; for example, true-color systems have no colormap because they can display any color. For maximum portability, you should not write applications programs that require writable cells in the default colormap.
- If you try to create a very large pixmap (for example, one with 2 Gbytes of data), the server will crash. There is no simple workaround for this problem.
- If you try to create a very large window (for example, one with a height of 64K pixels), the sever will crash. There is no simple workaround for this problem.
- In some cases, `XDrawPoints` does not always draw the points on the specified pixel.

2.2 DECwindows Applications

This section provides notes on the following:

- `dxmail`
- `dxterm`
- `dxcalendar`
- `dxcalc`
- `dxdb`
- `dxsession`
- `dxvdoc`
- `dxdiff`

- dxue
- dxnotepad
- dxpaint
- dxpsview
- dxwm

2.2.1 The dxmail Mail Handler

The following notes apply to dxmail:

- If dxmail should crash after a system installation, you should make sure that the `/usr/lib/X11/app-defaults/Mail` file is present on your system.
- The OR capability in Pick has been disabled due to a problem with the XUI Toolkit attached dialog box widget. The problem causes the dxmail application to go into a tight loop that prevents display updating and consumes most of the CPU resources.

The function was disabled with the following lines in the file:
`/usr/lib/X11/app-defaults/Mail`

```
! The next 2 lines turn off the "or" buttons in the pick window.
! This is temporary, until a bug in adb widgets gets fixed.
*pick*or.sensitive: off
*pick*bigor.sensitive: off
```

- If you create a mail message using dxmail, enter text that wraps across line boundaries, and then try to print that mail message, the lines will be truncated.

To work around this problem, enter carriage return characters when typing long lines in the text widget.

2.2.2 The dxterm Terminal Emulator

The following notes apply to dxterm:

- A security feature has been added to the dxterm terminal emulator. You can set the secure mode by selecting the Secure Keyboard item from the Commands pull-down menu.

The window will appear in reverse video after you specify Secure Keyboard. Also, the toggle button will appear highlighted next to the Secure Keyboard item to indicate that security mode has been set.

All keyboard input will be directed to the secure window, even if you have selected another window for input focus. This is useful for keeping passwords and confidential information secure.

When you iconify a secure window, the security mode is turned off. If you want security to be on, you must turn it on again when you change your icon back to a window.

If you try to create a second secure window, dxterm beeps, which indicates that only one emulator can be in security mode at a time.

- A resource has been added that allows you to block keyboard and mouse events that have been sent from a client. This resource has been added to eliminate character strings from being sent when the terminal is idle. This stops an unauthorized user from putting simulated keyboard input into your dxterm window, and executing commands as you. For example, an unauthorized user could remove a file of yours by sending an rm command while your terminal was idle.

The following resource has been added:

`allowSendEvents` Specifies that clients can send keyboard and mouse buttons events. The default is False.

- The auto-resize terminal option allows the terminal's logical size (rows and columns) to change when the window's size is changed. If you do not set this option, when you resize your window, your window will have the same number of rows but will appear longer or shorter.
- When using some versions of emacs with dxterm, be sure the TERM environment variable is not set to xterm. Set TERM to vt100 to insure that the emulator functions correctly.
- When you use some versions of emacs with dxterm, CTRL/S causes input to no longer be accepted by the emulator. To avoid this situation, set the following resource in the .Xdefaults file:

```
DXterm.main.terminal.controlQSHold:      off
```

- Keyboard:
 - Compose allows you to compose only ISO Latin-1 characters; DIGITAL MCS characters are not supported.
 - The following items are not functioning in Customize/Keyboard:
 - Caps lock/Shift lock
 - Tilde key
 - Comma key
 - <X] key
 - Angle brackets

- The following control keys are not functioning. Use the following replacements:

Key normally used	Replacement
Ctrl/2	Ctrl/space
Ctrl/3	Ctrl/[
Ctrl/4	Ctrl/slash
Ctrl/5	Ctrl/]
Ctrl/6	Ctrl/Shift/~
Ctrl/7	Ctrl/Shift/_
Ctrl/8	<X]
Ctrl/?	Ctrl/Shift/_
Ctrl/~	Ctrl/Shift/~
Ctrl/<X]	Ctrl/X

- User-defined keys (UDKs) cannot be loaded from the host.
- Selecting foreground and background colors in the Session Manager will select the background and foreground colors respectively in `dxterm` if you select "light characters, dark background" (Screen Mode off).
- The font names listed in the `dxterm` manpage used the old font naming convention. The font names should be as follows:

<code>littleFontSetName</code>	Specifies the font used for the "little" font set. The default is <code>"*-.Terminal-*-*-*-*180-*-*-*-*"</code> .
<code>bigFontSetName</code>	Specifies the font used for the "big" font set. The default is <code>"*-.Terminal-*-*-*-*140-*-*-*-*"</code> .
- The Secondary Device Attributes (DA) sequence does not give a properly formatted response. This request is made with the following sequence:


```
ESC > c
```

Usually, this reports the specific device type and version of the terminal. This sequence is rarely used and no known ULTRIX programs use it.
- If a process is running in background on a local `dxterm` and the window is killed, the output from the background process may be seen when the next `rlogin` is performed.
- When you start a `dxterm` window and use the `.X11startup` file, the commands in your `.login` file are not executed.
- ReGIS emulation does not work; `dxterm` crashes if it receives any ReGIS data.
- On a 4-plane color system (for example, a VAXstation II/GPX), `dxterm` creates a private colormap for each window in order to emulate a 4-plane VT340 on a 4-plane workstation. `dxterm` creates

the colormap when any sixel graphics are displayed in the window. The window manager then loads the private colormap when that window has the input focus, and it loads default colormap when the window loses the input focus. To restore a dxterm window to using the default colormap, first clear the window by selecting Clear Display command in the Commands menu and then reset the terminal by selecting Reset Terminal command in the Commands menu.

- There are various problems when displaying sixel graphic while recording lines off the top. In some cases, recorded lines can overprint each other when you use the vertical scroll bar. To avoid these problems, turn off the Record lines off top button in the Customize Window menu when displaying sixel pictures.
- Attempting to resize a dxterm window before you get a prompt in the window may cause the window to disappear.
- The caps lock/shift lock item does not work in Customize/Keyboard. To change between caps lock and shift lock, use the Customize Keyboard display in the Session Manager.

2.2.3 The dxcalendar Program

The data files for dxcalendar on VAX-based and RISC-based systems are incompatible. Any VAX-based dxcalendar database file that is run with dxcalendar will be converted to a new format and cannot be used on a VAX-based system again. In future VAX-based systems, the dxcalendar program will be fixed to understand the new format. If you must use your VAX-based dxcalendar database on your RISC-based system, make a backup copy of it before running the RISC-based dxcalendar program.

2.2.4 The dxcalc Calculator

- You can invoke help only once in a calculator session.
- If you try get a glossary of terms from the Help -> About menu, dxcalc may crash.

2.2.5 The dxdb Debugger

The following notes apply to dxdb:

- The debugger may crash and dump core if one button, such as the Step or Skip button, is clicked on repeatedly. This error occurs when dxdb has not finished updating all the windows and starts another callback before it is ready.

- Occasionally, extra lines will be drawn across the source window in addition to the current execution cursor. By causing an expose event on the source window, you can remove them.
- The execution cursor may move off the bottom of the source window without the source window adjusting itself to keep the cursor in view. You can scroll the window to keep the cursor in view.
- For the present, you should run dxdb from the same directory as the source code and the executable. In addition, the executable should be compiled with the `-g` option, should be owned by the user, and should not be stripped.
- You must remove the file `$HOME/.dbxinit`. The dxdb program works incorrectly if this file exists.
- Use the `dxdb*terminalEmulator` resource to change the version of the terminal emulator used by dxdb. Its default value is `/usr/bin/dxterm`. To use a different terminal emulator, put this resource in your `.Xdefaults` file.
- Terminal emulator (dxterm) windows do not close on their own. You must pull down the Commands menu and choose the Quit menu item. After running a program under dxdb, you should quit from the dxterm session before running dxdb again.
- After quitting from dxdb, you must also quit from the dxterm session before your dxdb windows will close.
- Very long lines in the Examine and Dump windows may cause problems.
- The Source window's Next Func button sometimes positions lines incorrectly.

2.2.6 The dxsession Session Manager

The following notes apply to the dxsession session manager:

- The dialog box for keyboard customization for foreign languages in the Customize menu does not work.
- The message window is not selectable.
- Messages from `.login` files are displayed in caution boxes. (For example, `ioctl` not supported, where are you? ...)

To stop the caution dialog box from being displayed and have the session manager remove the Start Session dialog box, remove the commands that specify terminal settings from your `.login` file or surround the commands with the following conditional statement:

```
if( $term != "none" && $term != "None" ) then
  commands
endif
```

This conditional statement checks to see if the terminal type is `None` (the type the session manager assigns to all terminal emulator windows). If the terminal type is `None`, the terminal settings commands are not invoked. If you are logging in from a real terminal, the commands are executed.

For a sample `.login` file, see `/usr/skel/.login`.

- If you make a change to the session manager resource with the Customize menu and do not save them, selecting "Use Last Saved Settings" during the same session will cause the session manager to do one of two things:
 1. If it is just after installation, the session will terminate and return to Xprompter (`login`).
 2. Any other time, the session manager window will disappear, although the `dxsession` process is still running.

The time it takes for the window to disappear depends on the activity directed to the session manager window (for example, iconification or resize). When this occurs, any other processes running on the display (for example, window manager or terminals) will still function properly, but the `dxsession` process must be killed with the `kill` command to return to the login prompter Xprompter.

The workaround to this problem is to log out and log back in, rather than use the "Use Last Saved Setting" option.

- For the printscreen option, there is no support for grayscale or color output. Black-and-white output from color or grayscale systems is possible.
- For the printscreen option, the aspect ratio qualifier does not work. The aspect ratio setting of "2 to 1" in the Customize Print Screen dialog box (pulled down through Customize in the Session Manager) does not work correctly. You should use only the default "1 to 1" setting.
- The `dxsession` print screen capability uses the print widget.

2.2.7 The dxvdoc Program

The dxvdoc program has the following known restrictions:

- Complex document layout is not supported, including page and galley layout. Some text display attributes are also not processed when you display a document.
- Specifying on the command line a file containing a graphic which displays on the first window causes the Viewer menu bar options and scroll bar arrows not to appear. You can avoid this problem by selecting such files from the Viewer open file selection box. A workaround is to increase the window size using the resize button.
- The Viewer may exit if you open several DDIF files containing graphics in a single session. A workaround is to limit the number of graphics files viewed, and reinvoke the Viewer in order to view more files.
- Graphics displayed on a monochrome screen use a black foreground and white background. If the default user settings include a default dark background, the graphics will probably not be seen until a re-expose; that is, the screen will appear blank because the graphics have been drawn black on black. A workaround is to increase the window size using the resize button.
- If scrolling an image is not possible, it may be represented by an improperly specified DDIF bounding box. A workaround is to increase the window size using the resize button.
- The CDA Toolkit requires that documents referred to by DDIF document external references exist in the current default directory if a full file pathname is not specified in the DDIF file external reference description.

2.2.8 The dxdiff Visual Differences Program

- The dxdiff application will exit and dump core if you use two instances of the "Do Differences in New" menu option.
- Some font spacing information does not work properly. If the highlighted difference areas appear out of register with the normal areas, use a different font.
- Modified resources specified in the .Xdefaults file do not take effect on any display except the first one.

2.2.9 The dxue User Executive

- In the script `/usr/lib/X11/ue/docp`, echo statements on lines 13, 18, and 21 should be deleted or commented out.
- There are limitations to the size of the file you can use `cat` on. If you try to use `cat` on a file that is too large, `dxue` will not display the entire file. If the file is too long, the top of the file will be displayed and there will be a note at the bottom saying that data was lost. As a workaround, put the following script in your path, make it executable, and create a new menu item for it. For further information, see the *DECwindows User's Guide*. It will run more in a terminal emulator. The script is as follows:

```
#!/bin/sh
CMD=/usr/ucb/more
FUNCTION=More
DOING_FILES=0
Quiet=0
for arg in $@
do
  case $DOING_FILES in
    0)
      case $arg in
        -uefiles)
          DOING_FILES=1
          ;;
        -quiet)
          QUIET=1
          ;;
        *)
          CMD="$CMD $arg"
          esac
          ;;
    1)
      FILES="$FILES $arg"
      esac
  done
  case $QUIET in
    0)
      echo $FUNCTION Files:
      read FILES
      case $FILES in
        ?*)
          exec dxterm -e /bin/sh -c "$CMD $FILES ; cat"
          esac
          ;;
    1)
      exec dxterm -e /bin/sh -c "$CMD $FILES ; cat "
      esac
  done
```

- Double-clicking on a file is not implemented. It will be treated as two single clicks.
- In the Processes dialog box, the list box clips the push buttons next to it. The push buttons still work, but they are unreadable. They say, from top to bottom, Kill, Suspend, and Delete.
- There can be only one instance of any named view mapped at a time. If a view is selected from the Views menu and it is already mapped, it will simply be raised.
- Regular expressions typed into the file filter behave like shell expressions with the exception that an empty filter matches all files.
- Sometimes, dxue takes so long to look up the files in a directory that the X server assumes dxue has exited and destroys dxue's window, making it look like dxue crashed. The dxue process is still running and will need to be killed. This can happen while waiting for crashed NFS servers to time out.
- When leaving the Task status box and returning to the UE window after completing a task, dxue will display the files in the selected view and not the startup file view displayed before. The title bar reads Startup but the file view displayed is not the startup file view. To work around this problem, click on the name of the directory again.
- When using the File Properties dialog box to select properties, choosing Owner Selection on a large directory with many owners of the files will cause a significant delay in updating the display.
- Occasionally a pull-down menu may appear to be dislocated from its source in the pull-down. If this happens, move the dxue window slightly.
- Cutting and pasting into dxue from other applications is currently not functional.
- Highlighted text in the directory list and in the list boxes in the Edit Menus dialog box may be a color that is hard to read. Setting the `*boldForeground` resource to another color will fix this problem.

2.2.10 The dxnotepad Program

- The online help file for the dxnotepad application contains a category called Opening an Existing File that does not have the Include File... option listed. This option can be viewed elsewhere in the file by using the index option. The other entries, Open and Open..., appear correctly.

- In some instances, the menu accelerators do not work properly. When this occurs, the XUI Toolkit reports the errors on dxnotepad startup.

2.2.11 The dxpaint Program

When you use dxpaint, the palette's contents are first dumped to a file that is created in the directory where dxpaint was invoked and then are spooled to the printer. After the file is printed, the created file is not removed from the directory in which it was created. You may want to remove this file.

2.2.12 The dxpsview PostScript Previewer

To see if a file created with your text formatter adheres to the Adobe page description commenting conventions, look at the first line of the output file for the following line:

```
%!PS-Adobe-1.0
```

The Previewer also lets you view files whose first two characters are not:
%!

A warning message is displayed asking you to confirm whether the named file is actually a PostScript file.

The previewer's performance when skipping from page to page is faster on structured files. Unfortunately, this works only if the files are structured and commented correctly, and many are not. If you have trouble previewing a file, try toggling Use Comments under the Options menu.

The remainder of this section discusses:

- Known problems
- Application resource names
- Difference between the previewer and an LPS40 printer

2.2.12.1 Known Problems

- The Use Bitmap Widths toggle under the Options menu does not work.
- The error message box is not cleared when a new file is displayed. You can work around this by acknowledging the error message before viewing another file.

- The mark operator is regarded as a composite object by save/restore. This may cause problems when you view Version 2.0 VAX DOCUMENT-generated files. The previewer will bring up an error box with a message saying that restore is undefined. You will find the following in your PostScript file:

```
/svobj save def
mark
n your PostScript file:
svobj restore
cleartomark
```

Change this to:

```
/svobj save def
mark
n your PostScript file:
cleartomark
svobj restore
```

The following are problems with the previewer's PostScript interpreter. You will probably not encounter them with text files that are formatted by programs such as troff. You might see them if you write your own PostScript programs.

- There is a noticeable difference in the size between numerals and capitals in the font Helvetica.
- There is no way to stop the previewer if you give it a file with an infinite loop, short of killing the application. For instance, the following PostScript program causes an infinite loop:

```
%!
{1 pop} loop
```
- The wcheck operator returns false on stdout and stderr even though the streams are writable.
- Lines with the same non-integer widths may appear to have different thicknesses. Lines with butt end caps may have protruding dots at the ends of the lines.
- Character placement is incorrect for rotations of 90.0 and 270.0 at some point sizes.
- Zero-width lines that extend past the sheet boundary may crash the previewer.
- Using the nulldevice operator followed at any time by a showpage will cause a fatal error in the previewer.
- The settransfer operator affects value and setting of currentgray. The settransfer should leave the currentgray value unchanged, but does not.

2.2.12.2 Application Resource Names – The following resources can be applied in your .Xdefaults file.

PSViewer*pixmapCacheSize: integer

The previewer keeps a cache of pages. If the value is greater than 1, the previewer can display previously seen pages more quickly. However, the greater the number, the more memory used in the server. The default is 1.

PSViewer*fileNameSuffix: string

This is the file name suffix used by the Open... item in the File menu. The default is 'ps'.

PSViewer*fileNamePrefix: directory name

This is the directory used by the Open ... item in the File menu. The default is the directory from which the previewer was invoked.

2.2.12.3 Differences Between the Previewer and an LPS40 Printer –

The previewer emulates an LPS40 printer. However, it has several exceptions:

- The resolution on a laser printer is 300 dpi while the screen resolution is 75 or 100 dpi (depending on monitor size). This resolution discrepancy makes text at small point sizes (below 12-point) and lines with non-integer widths appear jagged (have rendering artifacts).
You can eliminate some line artifacts by following the example in Section 9.3 of the *PostScript Language Program Design* from Adobe.
- The operator stack size on the LPS40 printer is 500; the stack size for the previewer is 2048.
- The size of the dictionary stack on the LPS40 printer is 20; the corresponding limit in the previewer is 2048.
- The implementation of the quit operator is different. In the previewer it is effectively a null operator.
- The #copies operator has no effect in the previewer.

2.2.13 The dxwm Window Manager

This section provides notes on the following aspects of the dxwm window manager:

- Naming windows and icons
- Handling mapping requests

2.2.13.1 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 that are 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 provides an example of 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);
.
.
}
```

Notes

1. Do not type the backslashes (`\`); they are used as continuation characters and are displayed for text formatting reasons only.
2. The `XA_WM_NAME` property specifies which property is to be changed, while `XA_STRING` specifies its data type.
3. The 8 argument indicates that the data is in 8-bit format.
4. The `PropModeReplace` argument indicates that the previous associated information is to be discarded. For further

information, see the *Guide to the Xlib Library: C Language Binding*.

5. The specified string “My Window” is the new name for the XA_WM_NAME property, and the 9 argument indicates that the string has nine characters.
6. To change the icon name, substitute XA_WM_ICON_NAME for XA_WM_NAME.

2.2.13.2 Handling Mapping Requests – If you are using the dxwm window manager and are running an application that does not handle exposure events, you may see nothing in the window that you create even if you make calls to flush the output buffer.

This problem occurs because the server does not consider a request to map a window finished as soon as it sends the request to the window manager. The window, however, actually does not get mapped until after the window manager completes its processing. After sending the map request to the window manager, the server goes on to the next request. If the next request is to display graphics on the same window, the following occurs:

- First, the server issues the graphics request to the window. However, because the window has not been mapped by the window manager, the graphics request is discarded.
- Next, the window manager maps the window, thus causing the server to generate an exposure event to the client to redraw the window.
- Finally, because the application is not written to handle exposure events, it does not redraw the window.

The permanent solution to this problem is to have the application handle exposure events. However, a temporary solution to use while debugging is to request exposure events on the window as part of the initial call to XCreateWindow. Then, when the window manager maps the the window, you can wait for the exposure event before issuing the graphics request. That is, you should issue the graphics request only after you have been notified of the exposure event.

Note

You should not attempt to run dxwm and uwm concurrently.

2.3 Demonstration Programs

The following notes apply to the demonstration programs that are included with the kit:

- The `/usr/examples/xuidemo` directory contains a program that demonstrates the XUI widgets. The program `xuidemo` has separate examples of all the XUI widgets and also lets you switch the language of the program's interface back and forth between English and German.

The `xuidemo` executable and `uid` files are not shipped with the kit. To create these files, execute `make` or `make all`. Do not attempt to reinstall these files as root while in the `/usr/examples/xuidemo` directory, with the `make install` rule, or your `Makefile` will be removed.

- The `xsh` command searches for a User Interface Language (UIL) file called `xsh.uid` and passes command strings contained in that file to the shell to be executed. The `xsh.uid` file must be in the directory in which you execute the `xsh` command.

By default, a sample UIL file is supplied in the `/usr/demo/dwt` directory. To create your own UIL file, you can either copy the `/usr/demo/dwt/xsh.uil` file to your current directory and modify it, or create a new `xsh.uil` file. For further information, refer to the `Makefile` in the `/usr/demo/dwt` directory.

- The `debugger` executable and `uid` files are not shipped with the kit. To create these files, execute `make` or `make all`. Do not attempt to reinstall these files as root while in the `/usr/examples/debugger` directory, with the `make install` rule, or your `debugger.uil` will be removed.
- The `/usr/examples` directory contains the following programs:

<code>dxdb</code>	Demonstrates the <code>dxdb</code> debugger.
<code>DDIF_graphics</code>	Contains sample <code>DDIF</code> files.
<code>cards_db</code>	Contains sample <code>dxcardfiler</code> cards.
<code>debugger</code>	Contains a sample <code>RM</code> program.
<code>fonts</code>	Contains an ASCII listing of the fonts that are included with the kit.

2.4 X11 R3 Applications

This section provides notes on the following aspects of the X11 Release 3 applications:

- uwm
- xbiff
- xterm
- Changes to .Xdefaults from MIT X10
- Command line syntax changes
- Unsupported R3 clients

2.4.1 The uwm Window Manager

- When displaying applications while running the uwm window manager, you must specify the position and size of the application window on the application command line, in the .Xdefaults file, or manually with the mouse.
- To specify the position and size of an application window with the mouse, size the window by pressing one of the following mouse buttons:

Left	MB1	Positions the upper left corner of the window at the mouse pointer and sizes the window with a default width and length.
Middle	MB2	Positions the starting point of the window at the mouse pointer and sizes the window to a width and length determined by mouse movement.
Right	MB3	Positions the window at a default screen location and sizes the window with a default width and length.
- You should not attempt to run uwm and dxwm concurrently.

2.4.2 The xbiff Command

xbiff does not handle /usr/spool/mail/\$user being a directory.

2.4.3 The xterm Terminal Emulator

The following notes apply to xterm:

- If a large amount of text is displayed through an xterm window, it can hang until the next expose event is issued. This event can be forced by refreshing the xterm window (by moving, iconifying, restacking it, and so on).
- As shipped on the kit, xterm has incorrect permissions (not set uid root). To correct this problem, log in as root and type the following:

```
% su
password:
# chmod 4755 /usr/bin/xterm
```

- A security feature has been added to the xterm terminal emulator. You can set the secure mode by pressing CTRL/MB1 and selecting the Secure Keyboard item from the X11 xterm pull-down menu. The window will appear in reverse video after you specify Secure Keyboard. Also, a check mark will appear next to the Secure Keyboard entry to indicate that security mode has been set. All keyboard input will be directed to the secure window, even if you have selected another window for input focus. This is useful for keeping passwords and confidential information secure. When you iconify a secure window, the security mode is turned off. If you want security to be on, you must turn it on again when you change your icon back to a window. If you try to create a second secure window, xterm beeps, indicating that only one emulator can be in security mode at a time.
- A resource has been added that allows you to block keyboard and mouse events that have been sent from a client. This resource has been added to eliminate character strings from being sent when the terminal is idle. This stops an unauthorized user from putting simulated keyboard input into your xterm window, and executing commands as you. For example, an unauthorized user could remove a file of yours by sending an rm command while your terminal was idle and the file would be removed. The following resource has been added:
allowSendEvents Specifies that clients can send keyboard and mouse buttons events. The default is False.
- The xterm terminal emulator does not interpret the ESC 8 and ESC 9 sequences properly. Applications that send these sequences (for example, running the man command on some of the reference pages) may appear to hang. If this happens, reset your terminal by selecting "Full Reset" from the Modes pop-up menu that is invoked with CTRL/MB2.
- If you run the MIT R3 xterm, you may encounter problems highlighting text when the xterm window is partially obscured by the right edge of the screen. You should refrain from using the MIT R3 xterm terminal emulator.
- The -C option does not work and prevents xterm from coming up when used as a system login window.

- If a process is running in background on a local xterm and the window is killed, the output from the background process may be seen when the next rlogin is performed.

2.4.4 Changes to .Xdefaults from MIT X10

The new X11 R3 clients (for example, xterm and xmh) will not work with existing .Xdefaults files. Use the /usr/skel/.Xdefaults file supplied in the kit as a guide for modifying your Xdefaults. A brief summary of the types of changes you will need to make follows:

- Resource name wildcarding uses an asterisk (*) instead of a period (.).
- Some minor changes have been made to resource names; use /usr/skel/.Xdefaults as a guide for this.
- Ensure that font names in your .Xdefaults file correspond to font names supplied in the kit. Some of the MIT clients dump core if you give them an invalid font name.
- Use an exclamation point (!) for comments instead of a number sign (#). The session manager is able to deal with both, but xrdb (see below) does not handle the number sign unless a special command line option is specified.
- The .Xdefaults file is preprocessed by the session manager or xrdb before it is downloaded to the server. As a result of this preprocessing, comments are stripped out along with portions of the .Xdefaults file that are taken out by #ifdef statements.
- All translation tables are now specified in the .Xdefaults file. This design makes the .XtActions file obsolete.

All X11 clients (MIT and DECwindows) use only the .Xdefaults file downloaded to the X server (by the session manager at session start or by xrdb). This means that all .Xdefaults files on machines other than your workstation are irrelevant. Any edits to your workstation .Xdefaults file are not effective until they get downloaded to your server. You can force the session manager to download the file by logging out and logging back in or by typing the following:

```
% xrdb .Xdefaults
```

For more information, see xrdb(1X) in the *ULTRIX Reference Pages*.

2.4.5 Command Line Syntax Changes

- While specifying a display variable on the command line, you must precede it with a `-d` or a `-display` option. For example:

```
% /usr/bin/xterm -d dorothy:0
```

- Geometry specifications must be preceded by the `-geometry` option.

2.4.6 Unsupported R3 Clients

Some of the unsupported MIT R3 clients (specifically, `xkill` and `xwd`) run better with the `uwm` window manager than with the standard DECwindows window manager, `dxwm`.

When used with `dxwm`, `xkill` kills `dxwm` instead of the pointed-to window.

When used with `dxwm`, `xwd` saves the entire screen instead of the pointed-to window.

2.5 DECwindows Fonts

This section provides notes on screen font functionality. The software includes the following:

- The full ULTRIX Worksystem Software Version 2.0 font set for both 75-dpi and 100-dpi color/monochrome displays. This library of 464 fonts provides:
 - Custom-designed ULTRIX Worksystem Software Version 2.0 menu and label fonts
 - Special `dxterm` terminal emulator fonts in both large and normal sizes
- A font compiler (`/usr/bin/xfc`) for compiling binary distribution format font files (`.bdf` files) into ULTRIX Worksystem Software Version 2.0 server natural-format font files (`.snf` files).
- 123 75-dpi display fonts from MIT X11 Release 3. The notes below do not apply to the MIT fonts.

The following sections discuss:

- Font file names
- Access to font names
- Understanding font names
- Font properties
- Font problems

2.5.1 Font File Names

The following notes apply to font file names:

- All font file names are in lowercase.
- All font file names have .snf appended to them.
- Previously, the font file names were not called out explicitly but were grouped by font family and style. The font file names are not used by applications or defaults files; the important element is the font name.

2.5.2 Access to Font Names

The font names for the ULTRIX Worksystem Software font library are listed in the /usr/examples/fonts directory, in the files fontnames.75dpi for 75-dpi fontnames and fontnames.100dpi for 100-dpi fonts. If your monitor is a 19-inch monitor, then your screen density is 75 dpi. If your monitor is a 15-inch monitor, then your screen density is 100 dpi.

ULTRIX Worksystem Software Version 2.0 applications and default files can access fonts by using file names or by using font names. File names must end in .snf and must be in a directory that is in the server's search path. The preferred approach, however, is to use font names. A font name consists of a series of parameter values separated by hyphens.

Access by font name is used for both the font name pattern argument (with optional wildcards) and the returned font name list by XFontNames and XFontNamesWithInfo, and as an input argument to XFont to open and load a screen font.

ULTRIX Worksystem Software Version 2.0 applications specifying default fonts in their .Xdefaults files also use this font-naming convention, substituting wildcard characters for fields to be filled in at startup by the application. For example, you may specify FAMILY_NAME, WEIGHT_NAME, and POINT_SIZE for a default font, but not specify RESOLUTION_X, RESOLUTION_Y, and PIXEL_SIZE to get the correct physical font size automatically independent of the resolution of the display. The application using this default would then fill in the resolution and other significant fields and query the server to obtain a list of possible fonts before attempting to open them. For ULTRIX Worksystem Software Version 2.0, a simple font selection algorithm that always chooses the first font match is implemented because servers are guaranteed to supply the font library. In the future, application defaults are planned to be accessed by an options menu scheme.

Servers also support wildcarding the font name argument given directly to an OpenFont request, avoiding the use of ListFonts* to select an available font. Note, however, that the server's algorithm for choosing a font from an ambiguous font name pattern is server dependent.

2.5.3 Understanding Font Names

When using either routine, pass the display identifier and font name. Xlib font names consist of the following fields, in left-to-right order:

1. Foundry that supplied the font, or the font designer
2. Typeface family of the font
3. Weight
4. Style (roman, italic, oblique, reverse italic, or reverse oblique)
5. Width per horizontal unit of the font
6. Additional style font identifier
7. Pixel font size
8. Point size
9. Resolution in pixels/dots per inch
10. Spacing (monospaced, proportional, or character cell)
11. Average width of all characters in the font
12. Set character encoding

The full name of a representative font in `/usr/lib/X11/fonts/decwin/screen100` is as follows:

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

Note that in this and the following examples, you should do not type the backslash character (`\`), which is used here as a continuation character. The font is named "ITC Avant Garde Gothic." The font weight is book, font style is roman, and width between font units is normal.

The pixel size is 14 pixels and the decipoint size is 100.

Horizontal and vertical resolution in dots per inch (dpi) is 100. When the number of dots per inch is 100, 14 pixels are required to be a 10-point font.

The font is proportionally spaced. Average width of characters is 80. Character encoding is ISO8859-1.

The following is an example of the full name of the comparable font designed for a 75-dpi system, in `/usr/lib/X11/fonts/decwin/screen75`:

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


This font requires 10 pixels to be 10 points. This font differs from the previous font only in pixel size, resolution, and character width.

Xlib enables clients to substitute a question mark for a single character and an asterisk for one or more fields in a font name. The following shows how to specify a 10-point ITC Avant Garde Gothic font of book weight, roman style, and normal spacing for display on either 75-dpi or 100-dpi systems:

```
-ADOBE-ITC Avant Garde Gothic-Book-R-Normal--*-100--*-P-*
```

When you use the asterisk, be sure that the substitutions are defined clearly. For example, the following font name specifies two fonts:

```
-ADOBE-ITC Avant Garde Gothic-Book-R-Normal--*-100--*-P-*
```

In this example, the leftmost asterisk substitutes for all fields before the 100. The name defines the following two fonts:

```
-ADOBE-ITC Avant Garde\  
Gothic-Book-R-Normal--11-80-100-100-P-80-IS08859-1  
-ADOBE-ITC Avant Garde\  
Gothic-Book-R-Normal--14-100-100-100-P-80-IS08859-1
```

The first is an 8-point font; the second is a 10-point font.

2.5.4 Font Properties

The font properties generally include the font name fields, plus other useful global font information such as the height of capitals (CAP_HEIGHT), calculated weight and setwidth (WEIGHT and SETWIDTH), and so on. All ULTRIX Worksystem Software fonts (except Terminal) have at least the following properties (more will be added in the future):

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

2.5.5 Font Problems

All terminal fonts contain some incorrectly encoded glyphs, and do not contain the appropriate font name properties.

Not all applications currently handle 100-dpi fonts gracefully.

2.6 The Xlib Library

The following notes apply to the Xlib library functions:

- If you send a `ClientMessage` event, then you must set the `format` member of its `XClientMessageEvent` structure. The valid choices are 8, 16, and 32. If you do not set this member, then the client that receives the event will exit with an internal Xlib error.

A `ClientMessage` event has room for client-supplied data, and the `format` member tells whether this data should be viewed as a list of bytes, short integers, or long integers. You must set the `format` member even if you do not have any client-supplied data.

- When you use the `XWriteBitmapFile` routine, the `#define` macros will include the file extensions instead of removing them as expected. This problem applies to all file extensions. For example, if you specify the file name as `foo.dat` the `#define` statement will look like:

```
#define foo.dat_width n
```

instead of:

```
#define foo_width n
```

- The FORTRAN bindings for the Xlib C language routines solve the conflicts between FORTRAN and C calling conventions when you call Xlib routines from FORTRAN. The FORTRAN bindings provide compatibility with C naming and calling conventions.

When compiling a FORTRAN program including calls to the C entry points, you must include the library and the Xlib libraries in the command as follows:

```
f77 -o program_name object_file_list /usr/lib/libF77.a -lX11 -loldX
```

The `/usr/include/X11/xlibF77.h` header file contains the FORTRAN constants and structures equivalent to the Xlib C definitions. With the exception of a few instances, the names are equivalent to the C naming conventions. They differ whenever a constant name differs with another only by the fact they are of different cases or whenever a structure begins with an underscore. The names have been changed as follows:

- KeySym definitions differing in name only by case have been resolved by appending `_lc` to the name containing the lowercase character. For example, `XK_a` was renamed to `XK_a_lc`.
- Structures beginning with an underscore have an capital X prepended to the name. `_XExtension`, `_XAsyncItem`, and `_XQEvent` were renamed to `X_XExtension`, `X_XAsyncItem`, and `X_XQEvent`, respectively.

When using a FORTRAN function, it must be declared as an `integer*4`. For example, `XOpenDisplay` should be declared as `integer*4 XOpenDisplay`. Functions that return character strings such as `XDisplayName`, `XDisplayString`, `XGetAtomName`, `XKeysymToString`, and `XServerVendor` must be declared as returning a character array of a specific size. For example, `XDisplayName` would variable being assigned the returned string is an array of at least 20 characters.

The X10 compatibility functions, resource manager routines, and the `XSetFontPath` routine are not supported by the Fortran bindings. The calling conventions for the following functions have changed:

- `XFetchName`

```
Status XFetchName(display, w, window_name)
    Display *display; /* Specifies connection to X Server. */
    Window w; /* Specifies window. */
    char *window_name; /* Returns the window name. */
```

- `XGetIconName`

```
Status XGetIconName(display, w, icon_name)
    Display *display; /* Specifies connection to X Server. */
    Window w; /* Specifies window. */
    char *icon_name; /* Returns the icon name. */
```

- `XSetCommand`

```
Status XSetCommand(display, w)
    Display *display; /* Specifies connection to X Server. */
    Window w; /* Specifies the window. */
```

- `XSetStandardProperties`

```
SetStandardProperties(display, w, window_name, icon_name,
    icon_pixmap, hints)
    Display *display; /* Specifies connection to X Server. */
    Window w;
    char *window_name; /* Specifies the window. */
```

(continued on next page)

```

char *icon_name; /* Specifies the window name. */
Pixmap icon_pixmap; /* Specifies the icon_name. */
XSizeHints *hints; /* Specifies a pointer to the size hints */

```

- **XrmGetResource**

```

Bool XrmGetResource(database, str_name, str_class, str_type,
                    str_type_return, str_value_return)
XrmDatabase database; /* Specifies the database */
char *str_name; /* Specifies the value name */
char *str_class; /* Specifies the class name */
char *str_type_return; /* Returns the destination's type */
XrmValue *value_return; /* Returns the value in the database */

```

2.7 The XUI Toolkit

The following notes apply to the XUI Toolkit:

- The XUI Toolkit widget development suggests that programmers not count on the size of the core part record as a fixed size but rather obtain the size at widget initialization time. This is so that users do not have to recompile their programs whenever a new version of DECwindows is released. Thus, indices and part offsets are used to access fields in user's widgets. For further information, see the *Guide to the XUI Toolkit: C Language Binding*.

When developing widgets on a RISC-based system, however, unpredictable results occur and the widget do not operate correctly if the index is greater than 255.

- The XUI Toolkit main include files DECwDwtApplProg.h and DECwDwtWidgetProg.h have been renamed to DwtAppl.h and DwtWidget.h to accommodate file name restrictions of some platforms. Also, the UIL main include file DECwDwtApplProg.uil has been renamed to DwtAppl.uil. The old names may still be used on platforms that support them. Do not use the old names; support will be dropped in a future release.
- An XtSetValues function on the attachments of a child of an attached dialog box can result in an infinite loop. Attachments should be set up when the child is created.

In most cases you can also avoid this problem by performing the XtSetValues while the widget is unmanaged.

- The dxsession print screen capability and the dxcardfiler and dxpaint programs use the DECwindows print widget. The following applies to the use of the print widget in these applications.

The print dialog box is used in many applications for the queueing of files or screens to a printer. By default, all printing devices on the system are displayed in the Printers list box. With this release, however, a new feature allows print queues to be associated with print formats through the definition of environment variables. The environment variable is defined to be a comma-separated or space-separated list of print queues with the first queue being the default choice. For example, if `DECwPrinterFormatANSI2` is defined to be `MY_PRINTER,LOCAL_LN03`, when you select `ANSI2` from the print format listbox, only `MY_PRINTER,LOCAL_LN03`, and `ANOTHER_LN03` will be shown in the printer listbox, with `MY_PRINTER` being the default choice.

The environment variable names and the print formats associated with each are as follows:

Environment Variable	Print Format
<code>DECwPrinterFormatText</code>	Text
<code>DECwPrinterFormatLine</code>	Line printer
<code>DECwPrinterFormatTerm</code>	Terminal
<code>DECwPrinterFormatANSI2</code>	ANSI2
<code>DECwPrinterFormatANSI</code>	ANSI
<code>DECwPrinterFormatPS</code>	PostScript
<code>DECwPrinterFormatReGIS</code>	ReGIS
<code>DECwPrinterFormatTek</code>	Tektronix
<code>DECwPrinterFormatDDFF</code>	DDFF
<code>DECwPrinterFormatDDIF</code>	DDIF

- If text is entered into the DECwindows text widget and wraps line boundaries, carriage return characters are not automatically inserted, and the resulting text will be truncated when printed.
For example, if you create a mail message using `dxmail`, enter text that wraps across line boundaries, and then try to print that mail message, the lines will be truncated.
To work around this problem, enter carriage return characters when typing long lines in the text widget.
- If you would like the text widget to have emacs-like key bindings, add the following text bindings to your `.Xdefaults` file:

```

!  

!DECtoolkit text bindings  

!  

*DwtText.Translations: #override\n\  

  Ctrl<Key>a:      beginning-of-line()\n\  

  Ctrl<Key>b:      backward-character()\n\  

  Ctrl<Key>d:      delete-next-character()\n\  

  Ctrl<Key>e:      end-of-line()\n\  

  Ctrl<Key>f:      forward-character()\n\  

  Ctrl<Key>j:      newline-and-indent()\n\  

  Ctrl<Key>k:      kill-to-end-of-line()\n\  

  Ctrl<Key>n:      next-line()\n\  

  Ctrl<Key>o:      newline-and-backup()\n\  

  Ctrl<Key>p:      previous-line()\n\  

  Ctrl<Key>u:      kill-to-start-of-line()\n\  

  Ctrl<Key>v:      next-page()\n\  

  Ctrl<Key>w:      delete-previous-word()\n\  

  Ctrl<Key>y:      unkill()\n\  

  Ctrl<Key>z:      scroll-one-line-up()\n\  

  Mod1<Key>b:     backward-word()\n\  

  Mod1<Key>d:     delete-next-word()\n\  

  Mod1<Key>f:     forward-word()\n\  

  Mod1<Key>v:     previous-page()\n\  

  Mod1<Key>z:     scroll-one-line-down()\n\  


```

- The help widget does not support XtSetValues of many text resources.
- There is a problem with widgets that pop up spring-loaded widgets directly over themselves. The first widget will not see the LeaveWindow event that is produced as the popped-up widget is mapped into the pointer location. This is due to a problem in the event-dispatching mechanism in the MIT R3 intrinsics.

For example, a widget specifies the following translation syntax:

```

<EnterWindow>: highlight()
<LeaveWindow>: un_highlight()
<Btn2Down>: popup_menu()

```

When the pointer enters the widget's window, the widget is highlighted. When MB2 is pressed, the pop-up menu is displayed. A LeaveWindow event should be dispatched to unhighlight the widget as the pointer is moved into the pop-up menu. This LeaveWindow event is not delivered and the widget is left in the highlighted state when the menu pops down.

- XUI Toolkit dialog boxes perform an XGrabKey on the TAB key so that they can synchronously transfer focus to the next child within the dialog box. If a dialog box receives a TAB key while the Toolkit is filtering events (for example, while another modal dialog box is up), the original dialog box will not see the TAB event and

will never call `XAllowEvents` to unfreeze the keyboard. If this happens, exit from the application and start it again to unfreeze the keyboard.

- For the list box widget, the shift MB1 double-click for the extended confirm callback is not supported.
- To avoid confusion between XUI Toolkit intrinsics header files and X11 R3 intrinsics header files, `mit` is used in the path. The X11 R3 header files are not merged with the XUI Toolkit header files. To avoid changing your source files, update your Makefiles with the following line: `-I/usr/include/mit`.
- The FORTRAN bindings for the XUI Toolkit C language routines solve the conflicts between FORTRAN and C calling conventions when you call XUI Toolkit functions from FORTRAN. The FORTRAN bindings provide compatibility with C naming and calling conventions.

When compiling a FORTRAN program including calls to the XUI Toolkit entry points, you must include the library, the C XUI Toolkit library, and the Xlib library in the command as follows:

```
f77 -o program object_file_list /usr/lib/libdwtF77.a -ldwt -lX11 -loldX
```

The `/usr/include/X11/xlibF77.h` file contains the FORTRAN constants and structures equivalent to the XUI Toolkit C definitions.

When using a FORTRAN function, it must be declared as an `integer*4`. For example, `XtOpenDisplay` should be declared as `integer*4 XtOpenDisplay`.

The interface for the FORTRAN binding for `DwtOpenHierarchy` is as follows to accommodate the `file_names_list`:

```
int DwtOpenHierachy(num_files, name_len, names_list,
                   ancillary_structures_list,
                   hierarchy_id_return)
    DrmCount num_files;    /* the number of files in file
                           names list. */
    int name_len;         /* length of each file name. */
    String *file_names_list; /* array of file names. */
    IDBOSOpenParamPtr *ancillary_structures_list;
    DRMHierarchy *hierarchy_id_return;
                           /* an identifier of the DRM
                           hierarchy. */
```

ULTRIX-32 Operating System Notes 3

The notes in this chapter pertain specifically to the base operating system software that is part of the ULTRIX Worksystem Software, Version 2.0 (RISC) and ULTRIX-32, Version 3.0 (RISC) kits. This chapter discusses:

- Differences between RISC-based and VAX-based systems
- Hints for porting software to RISC-based systems
- Compliance with operating system standards
- Network File System
- Programming notes
- Libraries
- Compiler notes
- Commands and utilities
- System facilities
- Networking and communications
- Standards and external specifications
- Miscellaneous known problems

3.1 Differences Between RISC-Based and VAX-Based Systems

The notes in this section pertain to the differences between RISC-based and VAX-based systems. The remainder of this section discusses:

- Components not in RISC-based systems
- Components that differ
- Components on RISC-based systems only
- Unsupported components on RISC-based systems

3.1.1 Components Not in RISC-Based Systems

The following components are not supplied with a RISC-based system:

- **adb**
The adb absolute debugger has been replaced in functionality by enhanced RISC-based version of the dbx debugger.
- **arff**
This archiver is for RT-11 format devices used to make console media for certain VAX computers.
- **kern_loop**
Both /etc/kern_loop and /etc/kern_unloop are not applicable on RISC-based systems.
- **lk**
This linker is for use with VAX C and VAX FORTRAN only. It has been replaced by the linker that is provided with the RISC compiler system.
- **libg**
This library is not supported on RISC-based systems.
- **makespt**
The /etc/makespt utility has to do with placement of kernel structures and is VAX dependent.
- **mkconsole**
This system maintenance command is not supported on RISC-based systems.
- **profiling**
The /usr/lib/lib*_p.a profiling library, which is linked when a user wishes to profile program performance, is not supported on RISC-based systems. It has been replaced by the profiling package that is provided with the RISC compiler system.
- **radisk**
This command is for maintaining Digital Storage Architecture (DSA) disks, which are not supported on RISC-based systems.
- **rxformat**
This command formats floppy disks, which are currently not supported on RISC-based systems.
- **sizer**
This program, which sizes the VAX hardware, is not supported on RISC-based systems.

- **symorder**
This command reorders the symbol table in the kernel and is not needed.
- **vcc**
The VAX C compiler produces VAX machine code and is replaced with cc, the RISC C compiler.
- **2780/3780**
These commands are part of the RJE 2780/3780 emulator package, and because there is no synchronous port, they are not supported for RISC-based systems

3.1.2 Components that Differ

The following components differ on RISC-based and VAX-based systems:

- **Executable images**
Executable images on RISC-based systems are larger and therefore take up more disk space than their counterparts on VAX-based systems. This is due to the instruction set of the RISC architecture. Typically images on RISC-based systems are 30 to 40 percent larger than on VAX-based systems.
- **Shared memory segments**
The attach points for Shared Memory Segments (SMS) in the virtual address space of a process are different than on VAX-based systems. SMS are attached by means of the shmat system call and, by default, fall between the text segment and the private data segment. This means that the problem of private data segment expansion (by using the sbrk or brk system call) being restricted by an attached SMS does not exist on RISC-based systems. Programs should let the system default the attach address, whenever possible. For more details, see shmop(2) in the *ULTRIX Reference Pages*.

Attach points for Shared Memory Segments (SMS) in the virtual address space of a process must be aligned on 4-Mbyte boundaries. SMS are attached by means of the shmat system call. Processes that permit the system to default the attach points will find that they are properly aligned. Processes that explicitly attach to a given address will find that the attach will succeed only if the given address is 4-Mbyte aligned or if the SHM_RND flag is set. Note that the latter case will cause the address to be round down to a 4-Mbyte boundary. This restriction is imposed by hardware constraints. Programs should let the system default the attach address whenever possible. For more details, see shmop(2) in the *ULTRIX Reference Pages*.

- floating

There is no floating or double in the kernel. The FPU is assigned to a process, and the kernel manipulates scalars only. The `fixpoint.h` header file contains macros to convert an integer to its floating point format. User programs can use this.

- float-fmt

VAX processors typically use `dfloat` floating-point format. RISC processors use IEEE floating-point format (close to `gfloat`). Thus, on RISC-based systems, there will be greater range of floating numbers but less precision (fewer decimal places).

Programs that need the extra precision of `dfloat` will have problems. Programs that need to be cognizant of the low-level format of floating point numbers also will have problems. This will be true in RISC-based systems FORTRAN or any language that does floating-point work.

In addition, there is no equivalent to `dfloat` or `hfloat`.

- page size

The page size on a RISC-based system is 4 Kbytes (4×1024) by contrast to 1 Kbyte on VAX-based system. Programs should use the `getpagesize` system call or include `vmac.h` and use the macros for page size manipulations. For further information, see `getpagesize(2)` in the *ULTRIX Reference Pages*.

- config

The `smmax`, `smmin`, and `smbk` shared memory config parameters on RISC-based systems are in 4 Kbyte pages. On VAX-based systems, they are in 512-byte pages.

The config program has been moved to `sys/mips/mips/config/config` in contrast to `/etc/config`. There is a symbolic link from `/etc/config`. In addition, some new features have been added on RISC-based systems. Because of the 4 Kbyte page size, certain virtual memory sizing parameters are now 4Kbyte pages.

- csh

The RISC version of `csh` is based on different sources from the VAX version, is functionally equivalent to it, and has 8-bit character support.

- **prof**
The RISC version of `prof` is functionally different from the VAX version. For further information, see `prof(1)` in the *ULTRIX Reference Pages*.
- **ranlib**
The `ranlib` command organizes archives of object files to allow faster linking. While it still exists for RISC-based systems, it is a shell script to pass a flag to the `ar` librarian, which then performs the same function.
- **lint**
Some differences in the messages printed and the conditions checked do exist between the RISC and VAX versions of `lint`. To build `lint` libraries, use one of the following:

```
% lint -C libname myprog.c (VAX-based systems)
% lint -c myprog.c (RISC-based systems)
```


Note that the command line on RISC-based systems uses System V syntax.
- **nlist**
Because RISC-based systems use COFF format for object files, this structure is different from that used on VAX-based systems. Programs that had hard-coded initializations that assumed the VAX `nlist` structure will have to be changed.
- **/usr/mdec**
On RISC-based systems, this directory contains bootblocks that are installed to bootable disks.
- **sys**
The `/sys` directory on RISC-based systems has been changed to allow for both VAX and RISC kernel binaries to coexist in the same directory structure in the future; `/sys` has been reorganized as follows:

```
BINARY => b.mips/{BINARY, SYSNAMES}
file systems grouped: fs/{gfs,ufs,nfs,specfs}
net grouped: net/{net,netinet,netdnet,netimp}
RISC specific: mips/{mips, ... mipsif, ...}
                /mips/mips/{config}
sas grouped: sas/mips
```

- **COFF**
RISC-based systems use the COFF format (Common Object File Format) in its object files and load modules. Therefore, the following utilities that make use of the object file format have been replaced by their new versions: nm (has more symbol types), dbx, as, ld, ar, size, and strip.
- **make**
The System V make, which has been made backwards-compatible with Berkeley make, is used on RISC-based systems.
- **ps**
The ps command prints out statistics on system activity. Some output formats are changed slightly.
- **nroff/troff**
RISC-based systems have new versions, which read ASCII tables in contrast to the VAX-based versions, which use binary object file format tables to control output. These new versions provide the same action as System V.
- **emacs**
The /usr/new/gnuemacs command is V18.51 on RISC-based systems and V17.nn on VAX-based systems.
- **disktab**
By default, the root and swap partitions are larger on RISC-based systems than on VAX-based systems (see /etc/disktab).
- **MAKEDEV**
Device numbers on RISC-based systems are different from those on VAX-based systems.
- **a.out.h**
This header file does not include exec.h on RISC-based systems, as it does on VAX-based systems.
- **brk**
On VAX-based systems, the program's virtual address space begins at zero. Text starts at zero and runs to &etext. Data then follows to &edata. The bss segment then follows to &end, and the rest is available for growth.

On RISC-based systems, the virtual address space begins at 0x00400000. Text starts at 0x00400000 and runs to &etext. There is a gap to 0x10000000, where the data begins. Data runs to &edata, is followed by bss to &end, and the rest is available for growth.

The major implication here is the interaction between `brk` and `getrlimit`. On VAX-based systems, doing a `getrlimit` for `RLIMIT_DATA` was a possible approximation for the maximum value that could be passed to `brk`. On RISC-based systems, the correct value is:

```
"the value returned by a getrlimit" +0x10000000;
```

On VAX-based systems, the correct value is:

```
"the value returned by a getrlimit"+&etext.
```

One solution is to use `sbrk`, not `brk`.

• `cc`

The RISC C compiler is a new version of `cc`. The following differences should be noted:

- This version of `cc` does not support the `const` keyword.
- Pointers in RISC-based systems are unsigned, and in VAX-based systems, they are signed.
- On RISC-based systems, you cannot dereference `NULL` pointers; includes `arg` to `strlen`.
- `asm` is not supported on RISC-based systems in any form.
- The RISC C compiler does not allow "old-fashioned initialization." An example of this, which worked on VAX-based systems but gave a warning, and does not work on RISC-based systems is "`int i 0;`".
- `varargs` is different on RISC-based systems. Any program that tries to walk the argument list by taking the address of an argument and incrementing it will not be successful, especially for double precision arguments. Programs using the macros in `varargs.h` will work. Compiling with the `-varargs` option on RISC-based systems will attempt to detect nonportable code.
- The `setjmp/longjmp` buffer is larger on RISC-based systems. Programs with a hard-coded 10 word buffer will fail; programs that correctly include `<setjmp.h>` and declare a `'jmp_buf'` will work correctly.
- The RISC C compiler has boundary alignment rules. User programs should only see this as a performance issue (the kernel does fix-ups). It is better, however, to align double-words, words and half-words on natural boundaries. See `uac(1)` in the *ULTRIX Reference Pages*.

- Pointers cannot be used as the variable on `switch` statements on RISC-based systems.
- The RISC C compiler will not allow the same `.c` or `.o` file to be listed twice. It will generate doubly defined symbol errors; the VAX C compiler (`cc`) allowed this.
- On VAX-based systems, the `cc -L` option on the command line collectively affects `-l` options. On RISC-based systems, the `cc -L` option is seen strictly left to right. Therefore, if a `-L` is supposed to affect a `-l`, the `-L` must come first.
- On RISC-based systems, global symbols do not have an extra leading underscore. This mostly affects assembler programmers.
- The `-R` option (read-only text) is not supported.
- The `-Md` or `-Mg` options are not needed on RISC-based systems; the hardware has only one double precision format.
- RISC-based systems define a macro (for example, `LANGUAGE_C`) for the preprocessor that makes it possible to write multilingual include files.
- For `cpp` predefined symbols, `ultrix`, `unix`, and `bsd4_2` are both RISC-based and VAX-based systems. On RISC-based systems, the equivalent predefined symbol of `vax` is `mips` and `MIPSEL` and `host_mips` are also defined.
- The following RISC C compiler options are not supported on VAX-based system: `-l` (no dirname); `-P` (preprocess, produce `.i`); `-Wphase,opt`; RISC-based systems recognize the environment variables `ROOTDIR` and `TMPDIR`; `-cpp` and `-nocpp` (most useful for languages other than C); `-G` (relevant only to RISC architecture), `-j`, `-k`, and `-ko` (relevant only to RISC-based systems compiler design); `-std` (warn nonstd usage; `vcc` has `-V standard=portable`); `-volatile` and `-varargs` (modify compiler behavior in certain areas); `-V` (print versions); and `-EB` and `-EL` (see the *Guide to Languages and Programming for RISC Processors*).
- Profiling on VAX-based systems has two levels that can be selected with the `-p` and `-pg` options. Profiling on RISC-based systems also has two levels that can be selected with the `-p` option or by running the post-processor `pixie` program. The RISC C compiler is not affected by either option; all work is done in the assembler or loader (or postprocessor).
- One level of optimization exists on VAX-based systems, which is off by default and enabled with the `-O` option. Five levels of optimization exist on RISC-based systems. By default, the 2nd level

is used, which can be disabled with the `-O0` option. The `-O` or `-O2` options invoke something comparable to VAX-level optimization, and there are rather more complex processes that can be used with `-O3` and `-O4` options. RISC-based systems also have the `-Olimit` option that allows optimization to be bypassed with overly complicated code sections.

- On both RISC-based and VAX-based systems, the `-t` and `-B` options specify passes and paths; however, the pass names for `-t` differ (there are more on RISC-based systems), and the semantics of `-B` belong to the `-h` option. The `-B` option is used to specify a command suffix instead. RISC-based systems also have the `-H`, `-K`, and `-#` options that are designed for compiler development work.
- Like optimization, RISC-based systems offer four levels for debugging information (controlled by the `-g` option). VAX-based systems have only two (on and off).
- If a global data item is used as if it were a code location (for example, if a data structure has the same name as a system call), an error message similar to following will be printed at load time:

```
/lib/libc.a(gethostent.o): jump relocation out-of-range, bad object  
file produced, can't jump from 0x4197a0 to 0x10008198 (stat)
```

If this happens, you should change the name of the data structure (in this example, it was named `stat`).

3.1.3 Components on RISC-Based Systems Only

The following components are only supplied on RISC-based systems:

- `btou` and `utob`
These compiler tools convert between `ucode` and binary formats.
- `cord`
This is the code reorganizer for the compiler.
- `dis`
This is the disassembler.
- `odump`
This utility produces formatted dumps of object files.
- `pixie`
This utility provides program profiling.
- `ppu`
This produces pretty print `ucode` (intermediate compiler code).

- **rzdisk**
This utility is used to format, scan, and reassign bad blocks on SCSI disks.
- **showsnf**
This DECwindows command displays the natural format of font files that the server uses.
- **stdump**
This utility dumps the symbol table of object files.
- **uac**
This command is used to control printing of unaligned access messages.
- **uld**
This is the ucode (compiler intermediate) loader.

3.1.4 Unsupported Components on RISC-Based Systems

The following games, commands, and programs are not part of the ULTRIX Worksystem Software, Version 2.0 (RISC) or ULTRIX-32, Version 3.0 (RISC) kits.

- **adventure** (game)
- **apl** (apl interpreter)
- **cpm** (/usr/new/cpm)
- **doctor** (game)
- **f77** (BSD FORTRAN 77 version)
- **hyper** (/usr/new/hyper)
- **icon** (icon programming language)
- **ingres** (university version of INGRES)
- **lisp** (BSD Franz Lisp)
- **modula** (Modula-2 compiler)
- **spms** (/usr/new/spms)

3.2 Hints for Porting Software to RISC-Based Systems

The following are provided as helpful hints for porting software to a RISC-based system:

- **NULL pointers**
On VAX-based systems, a null pointer can be dereferenced because page zero of user process space is mapped and valid. On RISC-based systems, however, you cannot dereference a null pointer without a segmentation violation. The pointer must be tested for non-NULL first.
- **Alignment**
On VAX-based systems, short words (2 bytes) or long words (4 bytes) can be accessed on any byte boundary. On RISC-based systems, however, references must be "naturally aligned." Short words (2 bytes) must be on an even byte boundary. Long words (4 bytes) must be accessed on a boundary evenly divisible by 4.
Unaligned accesses in user programs cause a "trap" into the kernel (the system attempts to "fixup" the unaligned access). If the system is able to accomplish the fixup, a message is printed to the controlling tty (if there is one) stating at what pc the alignment error was encountered. If the system is not able to fixup the unaligned access, the process will be terminated with a SIGBUS (bus error) signal. Doing the fixups for the program does have a performance impact on the program. For further information, see uac(1) in the *ULTRIX Reference Pages*.
- **Signed and unsigned pointers**
On VAX-based systems, if a pointer type function returns a -1 as an error status, the comparison "if (ptr < 0)" can be made, because pointers are signed values.
On RISC-based systems, if a pointer type function returns a -1 as an error status, the comparison "if (ptr < 0)" will never be true, because pointers are unsigned values. (Thus, the compiler removes the code for the comparison.) The error returns are not caught. The comparison could be "if ((int)ptr < 0)" or "if (ptr == (char *) -1)".
- **Variable number of arguments**
On RISC-based systems, a set of macros exists to declare formal parameters and access arguments in a list of a variable number of arguments. For further information, see /usr/include/varargs.h (or /sys/h/varargs.h).

3.3 Compliance with Operating System Standards

This release of the base operating system, contains improvements and modifications to the system calls and library routines to meet or exceed compliance with a number of industry standards. These standards are recognized as the means to make an operating system independent of vendors' hardware. The standards to which the base operating system conforms are the System V Interface Definition (SVID), POSIX, and X/OPEN standards.

The ULTRIX-32 Operating System has been improved by adding support for SVID termio, and by fixing known SVID inconsistencies. In addition, System V terminfo and curses are now supported. SVID support for this release will be based on SVID Issue 2, Volume 1, with some Volume 3 additions.

The ULTRIX-32 Operating System is compatible with the Portable Operating System Interface for Computer Environments (POSIX), IEEE P1003.1 standard.

The ULTRIX-32 Operating System also contains changes to conform to the *X/OPEN Portability Guide*. The X/OPEN group plans to adopt the POSIX standard as its base when this standard is finalized.

3.4 Network File System

The notes in this section pertain to the NFS software. This section discusses:

- NFS file and record locking
- NFS server security changes

3.4.1 NFS File and Record Locking

The Network File System has been extended to support System V and POSIX-compatible file and record locking across the network. The lock manager and status monitor programs are included in this release and provide easy installation, deadlock detection, and the ability to configure out the networked base service for local system performance. This functionality allows an unlimited number of file locks. Previous versions of the ULTRIX operating system were limited by kernel data space for the number of supported file locks.

3.4.2 NFS Server Security Changes

Each NFS request is now checked to ensure that the file system being accessed is currently exported. If the file system is not exported, the operation is rejected and the ESTALE errno value is returned to the client

process. An unexported file system message is also written to the server console and error log.

An NFS server must run the `mountd` daemon. If a server does not run the `mountd` daemon, clients will not be able to access server file systems. Exported file systems are marked by the `mountd` daemon based on information found in the `/etc/exports` file. If this file is modified, the next time `mountd` is asked to process a request (for example, a `showmount` or `remote mount`), it updates the export state on the server.

If an administrator removes an entry from the `exports` file, and follows that by executing a `showmount -e` command, all subsequent NFS access requests for the removed file system or directory are rejected.

NFS server daemons must be started in the correct order, which is `portmap`, `mountd`, and `nfsd`. The `nfsd` daemon now exits if started before either the `portmap` or `mountd` daemons. The `nfssetup` setup script starts these daemons in the correct order.

3.5 Programming Notes

To conform to the POSIX standard and the common use of signal handlers, the definition has been changed from `int handler()` to `void handler()`. Programs compiled with `int handler()` generate a compiler warning message but run correctly.

Additionally, the signal routine definition has been changed from `int (*signal())()` to `void (*signal())()`. If your program includes the header file `<signal.h>` and uses the old syntax to define the signal routine, the program will fail to compile. Programs should not explicitly define the signal function; rather, they should use the definition included in the `<signal.h>` header file.

3.6 Libraries

The notes in this section pertain to the C and math libraries.

3.6.1 C Library

- The `execvp` function will receive a SIGILL error if passed a PATH list that contains names greater than `NAME_MAX` (255) characters long.
- The `atof` and `strtod` functions have the following problems:
 - `errno` is not set to `ERANGE` on overflow
 - `radix` points other than periods are not recognized if the `setlocale` function is called

- no overflow is indicated if the exponent is larger than 19327352832
- The `longjmp` returns zero if called with a second argument of zero instead of returning 1. You should not call `longjmp` with a second argument of zero.

3.6.2 Math Library

Certain math functions in `/usr/lib/libm.a` (for example, `asin`) do not properly set `errno` (to `EDOM`) and return 0.0 when arguments are out of range, as defined.

3.7 Compiler Notes

The following notes pertain to the RISC C compiler:

- The `-O` or `-O2` optimization level produces incorrect code in the following cases:
 - If a common subexpression using floating-point numbers appears on both branches of an `if-else` statement:

```
#include <stdio.h>
main(argc, argv)
int    argc;
char   **argv;
{
    int    start;

    if (argc >= 2) {
        if (sscanf(argv[1], "%d", &start) != 1
            || start >= 50.0) {
            fprintf(stderr, "Ouch.\n");
            exit(1);
        }
    } else {
        while (start >= 50.0)
            exit(2);
    }
}
```

Note that splitting the conditionals up or otherwise simplifying the code avoids the problem.

- If a null, redundant `if` statement occurs at the end of a `while` loop:

```

struct list {
    struct list *next;
};

main() {
    register struct list *ptr;

    for ( ; ptr; ptr = ptr->next) {
        func(ptr);
        if (ptr->next == 0)
            ;
    }
}

```

In this case, the code generated for accessing the ptr variable is incorrect. Deleting the unnecessary if statement solves the problem.

- Incorrect code is generated in some cases when a global variable is used on a switch statement and its value is changed immediately afterward. Instead, the value is changed before the switch is executed. For example:

```

extern choice;
extern global;

func() {
    if (global == 9) choice = 7;
    else global = 5;

    switch (choice) {
        case 3: break;
        case 6: global = 2;
            break;
        case 8: global = 0;
            break;
        case 9: global = 1;
            break;
    }
    choice = 7;
}

```

In this example, the choice variable is incorrectly assigned the value 7 before the switch statement is executed. If you encounter this problem, you should use optimization level 0 or 1.

- The RISC C compiler does not optimize the following program correctly:

```

main()
{
  int x,y;
  unsigned z;
  if (x-y+1 > z) good(); else bad();
}

```

The '+1' gets lost. Use the `-O0` compiler option to disable all optimization, or cast `z` to `int` in the expression. Code that compares signed and unsigned characters in this fashion is problematical and should be avoided.

- The compiler does not convert floating point numbers to unsigned long correctly if the resulting value is too large for a signed number.
- The C compiler does not accept a constant pointer expression in a case statement. For example:

```
(int)(amp((REC *)0)->field)
```

Remove the redundant cast, and it is accepted.

- You cannot pass the address of a floating point argument to another function. Use a temporary variable instead.
- The compiler does not correctly recognize variables declared as pointer to void (`void*`). You should use pointer to char variables (`char*`) instead.
- `cc`

When the `/tmp` file system is full, `cc` reports:

```

ugen: internal : line 0 : build.p,
line 1743 unexpected u-code

```

3.8 Commands and Utilities

The notes in this section pertain to the following ULTRIX-32 Operating System commands and utilities:

- `ar`
- `lint`
- `lprsetup`
- `lpr`
- `cp`
- `showmount`
- `mail`
- `sh`

- newcsh
- chfn
- cpio
- csh
- dc
- ltf
- make
- pstat
- rcp
- rdump
- script
- setld
- size
- talk
- uucp
- uucpsetup
- vi
- yppasswd

In addition, the following sections contain a description of an expression handling problem that occurs with some commands.

3.8.1 The ar Command

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 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.

3.8.2 The lint Command

The -o option on the lint command for your RISC system is similar in meaning to the -C option on VAX-based systems. If you would prefer them to both use -C, edit /usr/bin/lint, which is a shell script, and change the following lines:

From:

o) if ["\$OPT"]

To:

[Co]) if ["\$OPT"]

3.8.3 The lprsetup Command

For known serial printers (with the exception of the LA50 and the LA75), the `lprsetup` command sets the `fs` option in `/etc/printcap` to no parity. The `fs` option for the LA50 and LA75 is set to even parity. Having no parity can cause problems, especially if the printer's dip switches are set to a specific parity, because the terminal driver will set the line to some default parity. This may or may not be the same as the printer's settings.

Therefore, the printer's dip switches should be set to a known parity, and the `fs` option in `/etc/printcap` should be set explicitly to match the printer.

You can do this by editing `/etc/printcap` or by using the `lprsetup` command when creating the printing environment. For further information, see the `sg_flags` field in `ty(4)` in the *ULTRIX Reference Pages*.

3.8.4 The lpr Command

The `lpr` command allows control characters to be printed using the `-l` option. Even when this option is specified, print jobs are piped through the `of` filter or whatever filter is specified in the submitting `lpr` command. Some control characters, for example `^A` and `^Y`, have special meanings for the filters.

If the desired file contains these control characters, it may cause the filters to hang. If this happens, you will have to remove the job from the queue using the `lprm` command. You may also have to change the control characters in the file. If you change the control characters in the file, use the `lpr` command with the `-x` option. This is a transparent filter that allows all data to be passed to the printer unchanged.

3.8.5 The cp Command

Issuing the `cp` command with the `-r` option to two directories with the same name causes continuous recursion until it fills the file system or the user's quota limit is exceeded.

3.8.6 The showmount Command

The `-e` option for the `showmount` command will fail if the `/etc/exports` file on the server is larger than 2-3K. The result is that the request will time out and the system will display:

```
% showmount -e thumper
showmount: rpc timed out
```

The workaround is to log in to the server system (`thumper` in the example) and type:

```
% cat /etc/exports
```

3.8.7 Regular Expression Handling

Some commands have limitations in the way that they handle regular expressions. If the length of the expression exceeds the command's predefined boundaries, the expression may not be correctly processed by the command. This applies to the `ed`, `ex`, and the `grep` type commands.

3.8.8 The mail Command

The notes in this section pertain to the `mail` utility. This section discusses:

- Alternate forwarding files
- Daemon start order
- Delimiters for aliasing
- Invoking mail more than once

3.8.8.1 Alternate Forwarding File – A potential Network File System (NFS) problem can prevent access to the mail `.forward` file if the system serving the user's home directory is not up and the system does not share the `/usr/spool/mail` directory.

When mail is sent from a machine to a user on a system that contains a mail `.forward` file and that file is not accessible, the mail ends up in `/usr/spool/mail` on some other system. The mail appears to be lost and there is no notification to the user that the mail was delivered to the wrong system.

An alternate mail forwarding file solves this problem. The user can create a forwarding file on any system that forwards the user's mail. The file is created in the `/usr/spool/mail` directory and has the following syntax:

```
username.forward
```

Entries in this file are in the same format as the entries in the `.forward` file in the user's home directory.

The alternate mail forwarding file in `/usr/spool/mail` must be owned by the person to whom mail is directed.

If a `.forward` file in the user's home directory file is present and accessible and if there is also a `/usr/spool/mail/username.forward` file (on the same system), `sendmail` chooses the `.forward` file in the user's home directory instead of the one in the `/usr/spool/mail` directory.

3.8.8.2 Daemon Start Order – You must make sure that the `sendmail` daemon start specification in the `/etc/rc.local` file comes after the start of the Yellow Pages (YP) daemons. If the `sendmail` daemon is specified before the YP daemons, users defined in the YP maps for passwords and aliases will not be visible to `sendmail` and mail will get returned with the following error message:

```
unknown user
```

This problem only occurs when the order of the start specifications in the `rc.local` file is incorrect. If the `ypsetup` script is used to implement the YP database, the YP start specifications are automatically placed in the proper place in the file.

3.8.8.3 Delimiters for Aliasing – Alias names in the `.mailrc` file must be separated by commas or spaces, but not both. If you mix commas and spaces for delimiters between alias names, the alias fails. It generates bad address constructs in the To: fields.

3.8.8.4 Invoking mail More Than Once – If you invoke `/usr/ucb/mail` more than once on your current mailbox (`/usr/spool/mail/username`), you may lose mail. The mail program is similar to the `vi` editor in that when you invoke `/usr/ucb/mail`, you cause an edit to your mailbox as you delete mail. Therefore, if you invoke mail twice on your current mailbox, the results are unpredictable.

3.8.9 The `sh` Command

- The Version 7 Bourne Shell, `/bin/sh`, is not 8-bit clean.
- If the Bourne shell, `/bin/sh`, is run from another program (by the system or `exec` system call) whose maximum file descriptor in use is number 10, the prompt string will not be printed. This can happen if a program has eight files open (in addition to the customary `stdin`, `stdout`, and `stderr`) at the time `sh` is called.

The problem does not occur with the System V Bourne shell, sh5, or with the C shell, csh. It also does not happen if file descriptor 11 is in use.

3.8.10 The newcsh Command

The newcsh command has been enhanced with a command line edit feature.

The `/usr/new/csh` command is a modified version of the VAX version of newcsh, which included command completion and filename recognition. The enhanced newcsh contains all of the functionality of the VAX version of `/usr/new/csh`, plus the new command line edit feature.

For further information, see newcsh(1) in the *ULTRIX Reference Pages*.

3.8.11 The chfn Command

When you run chfn as root, you must supply a user name that is listed in the `/etc/passwd` file or the Yellow Pages (YP). If the user name supplied to chfn is unknown, chfn dumps core with a segmentation fault.

3.8.12 The cpio Command

The cpio command should not be used to save and restore symbolic links or directories which contain symbolic links. Corruption of the "linked to" file name can occur if it is used in this manner.

3.8.13 The csh Command

The following alias causes csh(1) to dump core:

```
alias xxx `foreach x ( 1 ) \  
anything`
```

Any use of the alias xxx causes csh to dump core.

To avoid this problem do not use a backslash (\) when aliasing a foreach or while loop. A second way to avoid this problem is to use a C shell script instead of the alias.

3.8.14 The dc Command

The dc command behaves unpredictably under the following conditions:

- With division OR remainder
- The dividend is negative
- The scale is zero

The result depends on whether the number of digits after the decimal is even or odd as the following script shows:

```
csH> dc
_8.88 (# of digits is even, result of -8.88/1 is -9)
1/p
-9
_8.888 (# of digits is odd, result of -8.888/1 is -8)
1/p
-8
_8.8888 (# of digits is even, result of -8.8888/1 is -9 again)
1/p
-8
_8.88 (# of digits is even, result of -8.88%1 is 1%p
9.112
```

You can prevent this problem by avoiding the specific conditions under which it occurs or by setting the scale value equal to the number of digits you want after the decimal place.

3.8.15 The ltf Command

If you try to extract files from a tape created with the ltf command, the utility generates an error message about header errors on the tape.

The workaround is to name individual files on the extraction list, one at a time. Individual files are extracted despite the error message.

These errors are not encountered when using ltf to extract from disk files.

3.8.16 The make Command

- If a makefile contains a dependency on a nonexistent file instead of always executing the rule, the current time is used for comparison. This can cause troubles when using make across an NFS environment if the time on the file server is later than the time on the client.
- If the variable \$* is used in an explicit makefile rule (that is, not a rule like .c.o), it will not have a predictable value. Note that the documentation for make states that \$* is only valid in implicit rules.
- The documentation for s5make suggests installing it as /bin/make to use the System V behavior. This will have no effect because /bin/make is now a symbolic link to s5make. Most System V make behavior is now incorporated into /bin/make; the only major difference is that make always uses /bin/sh, and s5make uses the SHELL environment variable, if present.
- The make command treats \$\$ as the end of a file name in a dependency list, not as a single \$.

3.8.17 The pstat Command

When the `-t` option is specified, the `pstat` command does not give the correct information about the local serial lines or the console.

3.8.18 The rdump Command

If you are using the `rdump` utility to dump from a machine running an earlier version of ULTRIX-32 Operating System, you must specify the `-o` option.

The following example illustrates the correct command line syntax for dumping from a pre-version 3.0 machine to one running version 3.0:

```
rdump Ouof V3.0_machine:/dev/rmt0h /local_pre-V3.0_filesystem
```

3.8.19 The script Command

When exiting the `script` command with a CTRL/D, the command hangs and will not release the terminal until all children exit.

To avoid hanging the process initiated by the `script` command wait until all children have exited or use the `kill` command to terminate all children before exiting.

3.8.20 The setld Command

The following notes pertain to the `setld` command.

3.8.20.1 Removing Subsets – Your ULTRIX-32 operating system comes with several thousand files, which are organized into discrete functional units, called subsets. If you need to remove any of the files that were placed on your system when you installed it, use the `setld` utility to remove them. Failure to use the `setld` command may degrade the usability of the software.

The `setld` utility tracks the files installed on your system by using data that it stores in `/usr/etc/subsets`. Do not remove any of the files in that directory, because you will lose the ability to install and delete system and layered product software.

3.8.20.2 Installing Subsets from Two Installation Devices – The results of attempting to use the `setld` command to install software subsets from two installation devices simultaneously are undefined. Do not attempt to run two instances of `setld` at the same time to install software subsets. Also do not attempt to use `dms`, `ris`, or `setld` at the same time.

3.8.20.3 Installing Subsets - The `-a` option of `setld` enables you to quickly re-install subsets that had been previously installed with the `-l` option.

Attempting to re-install a currently installed subset with `-l` will fail because the currently installed subsets are not offered on the `-l` menu.

Attempting to use `-a` to install subsets from the installation media without having first used `-l` on the media will also fail, because `setld -l` performs some necessary initializations for recognizing the subsets on the product tape. Note that the basic, advanced, and network installations perform these initializations for all media read during the installation.

You can determine whether a subset has been initialized in this way by checking to see if the subset is in the output from the `setld -i` command.

Both the `-a` and `-l` options will install subsets that had been deleted with the `-d` option.

3.8.21 The size Command

The `size` command 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, while more cryptic, essentially have the meaning as the following VAX-based `size` message:

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

3.8.22 The talk Command

The following notes pertain to the talk program.

3.8.22.1 The talk Command Is Not 8-bit Clean - The talk command is not 8-bit clean. Typing in DEC Multinational Characters (DECMCS) causes the characters to echo as a sequence of a caret (^) followed by the character represented with its high bit cleared.

This limitation makes talk unusable if you want to communicate using a language that has DECMCS characters in its alphabet.

3.8.22.2 Changes in the talk Program – To use the talk program with machines on your network that may be running earlier versions of ULTRIX-32 Operating System, you must initiate a session with the command `otalk (/usr/ucb/otalk)` instead of the command `talk`.

You must also respond to a request from a machine running an older version of the talk program with the `otalk` command.

The following example demonstrates how to use the `otalk` command. In this case, `user1`, whose system (`system1`) is running an earlier version of the operating system initiates a session with `user2`, whose system (`system2`) is running the current version. `User1` types the following:

```
system1> talk user2@system2
```

The following message appears on the screen of `user2`:

```
Message from Talk_Daemon@system2 at 12:37 ...
talk: connection requested by user1@system1.
talk: respond with:  otalk user1@system1
```

To establish the connection `user2` follows the instructions from the `Talk_Daemon` and types the following at the system prompt:

```
system2> otalk user1@system1
```

3.8.23 The uucp Command

The `-nrec` option of the `uucp` command is undocumented. This option does the same thing as the `-m` option, except that it does not send mail to the requester (sender) but rather to the recipient (rec) of the file. For further information, see `uucp(1)` in the *ULTRIX Reference Pages*.

3.8.24 The uucpsetup Command

The `uucpsetup` command does not modify the `/etc/remote` file to include the `tty` device names.

3.8.25 The vi Command

- If you set the environment variable `EXINIT` to `open`, `vi` causes a bus error core dump. `EXINIT` allows you to customize the mode of the editor at startup time. Do not initialize `vi` with `EXINIT open`.
- For the `vi` family of editors (`vi`, `ex`, `e`, and `edit`), do not set the `wrapmargin` to a positive value and then `wrapscan`. Doing so can cause data corruption in the edited buffer when performing a yank and put operation.

3.8.26 The yppasswd Command

The yppasswd command changes the protections of your password Yellow Pages related maps to 666 (-rw-rw-rw-), which allows all users read and write privileges.

To protect your YP maps from malicious users, change the modes of your */etc/yp/domain-name* directory to 700 (drwx-----).

3.9 System Facilities

The notes in this section contain information on the kernel, I/O, and related system tools.

3.9.1 Errors in the /etc/passwd File

Errors in the */etc/passwd* file may not be adequately checked by programs that read the file.

Any changes that you make to */etc/passwd* must conform to the file format described in the reference pages for *passwd(5)* and *passwd(5yp)*. In particular, if the user id (UID) or the group id (GID) is omitted or incorrectly inserted, some programs (those which do not use the *getpwent(3)* family of functions) may assume inappropriate defaults for a UID or a GID. This may introduce a security hole on some systems, especially those that allow a null UID to default to a value of 0.

The following are particularly important:

- Each entry in */etc/passwd* must contain at least the following fields: name, password, numerical user id (UID) and numerical group id (GID). Each field must be separated by a colon. Any line in */etc/passwd* without at least this much information is ignored.

For example, a system manager must minimally enter the following:

```
jcj::508:10:
```

An encrypted password can be added with the *passwd* command. Be sure that the UID and the GID fields are defined. A more complete (and so more acceptable) */etc/passwd* entry is as follows:

```
jcj::508:10:Jack C. Johnson:/usr/staff/jcj:/bin/csh
```

The only exception to this rule is for lines that begin with plus or minus signs (+ or -) for yellow pages access. These lines may contain a single name, preceded by + or - and followed by a colon. For example:

```
+jcj:
```

- The user name is restricted to a maximum of 8 characters.
- The encrypted password is seldom directly entered with a text editor. Instead you should use the `passwd` command to add an encrypted password. You can omit an encrypted password in `/etc/passwd`, leaving a zero-length password field. An empty password field allows a user to login without supplying a password.
- The user ID field (UID) and the group ID field (GID) must contain only the digits 0-9. The UID must have a value less than 32000. If you omit either the UID or GID fields, the system defaults to an inappropriate value. See `UID_MAX` in `/usr/include/limits.h` for a definition of the maximum uid value.
- The group ID field (GID), like the UID, must consist only of the 10 digits, 0-9, and its value must be less than 32000.

3.9.2 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. For further information, see `ptrace(2)` and `dbx(1)` in the *ULTRIX Reference Pages*.

3.9.3 The sendmail Program

The `$x` macro, which represents the full personal name of 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. The `sendmail` program uses information in the `/etc/passwd` file to get personal names; the program has no access to remote password files.

3.9.4 Writing into a Remote a.out File

If a remote `a.out` image file is written to 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 up on a given client while processes on that client are being killed, the new process will fail and the system responds by displaying the message:

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

In this case, retry the process.

3.10 Networking and Communications

This section contains information on networks and networking protocols.

3.10.1 DLI/802 Passes Up Packets That Should Be Dropped

When a user opens a DLI/802 socket and enables an individual and group SAP, the socket can receive "Unnumbered Information" messages sent to that SAP with the poll bit set to 1. To detect this event, the user must use the `recvfrom` call and process the control field passed in the address/header structure. This is done by testing bit 4 (assuming the low order bit is numbered 0) of the `U_fmt` member of the `osi_802hdr` structure. "Unnumbered Information" packets should only have the poll bit set to 0. If the bit is set to 1, the packet should be ignored.

3.10.2 DLI/Ethernet Drivers

The "Unrecognized Frame Destination" counter is maintained by the Ethernet drivers. It is incremented when a driver receives a packet for which there is no end user. When the Data Link Interface (DLI) is built into the system, the Ethernet drivers automatically pass to it all unwanted inbound packets instead of incrementing this counter. If DLI cannot find a user for which the packet is destined, then the packet is dropped. However, there is currently no mechanism in place for DLI to inform the driver that the "Unrecognized Frame Destination" counter should be incremented.

There is no workaround for this condition, but it is unlikely to affect anyone but users trying to troubleshoot a network problem.

3.10.3 Incorrect UUCP Checksum Calculation with F Protocol

When transmitting a file to a non-ULTRIX operating system using `f` protocol, the `uucp` daemon does not calculate the checksum correctly if the file contains binary or 8-bit data.

3.10.4 Performance Improvement for YP and BIND

To improve on performance when using Yellow Pages (YP) and Berkeley Internet Name Domain (BIND) in a large network, remove all entries from the `/etc/hosts` file except for `localhost` and your host, and use YP and BIND to resolve all others. In the `/etc/svcorder` file, place "local" first. This is an improvement, because `localhost` and the local host name are requested frequently.

3.10.5 Manual Update of BIND server's Serial Number

Be sure to increment the serial number (integers only) in a BIND primary server's SOA (Start of Authority) record when the data files are manually updated. SOA records usually exist in the following files:

```
/etc/namedb/named.hosts  
/etc/namedb/named.rev
```

Other servers (secondary, root, and so on) check this number to decide if a zone refresh is necessary to update its information. For example, if a secondary server's serial number is less than or equal to the primary server's serial number, then a zone refresh is requested. If this number is not updated on the primary server, other BIND servers hold old BIND data until that data expires.

To increase the frequency of this serial number check, reduce the refresh time on the other servers or reduce the expire time on the primary server.

3.10.6 Use of Full Host Names When Running BIND

The following files must use the host name as defined by `/bin/hostname` on the client:

```
/etc/exports  
/etc/hosts.lpd  
/.rhosts  
~/.rhosts
```

If the machine is running Berkeley Internet Name Domain (BIND), the `/bin/hostname` should be the local host name plus the BIND domain name separated by periods, for example:

```
/bin/hostname student.harvard.edu
```

In this example, `student` is the local host name in the BIND domain `harvard.edu`. This command is in the `/etc/rc.local` file. For further information, see the *Guide to BIND Service*.

3.10.7 NFS Server Security Changes

Each NFS request is now checked to ensure that the file system being accessed is currently exported. If the file system is not exported, the operation is rejected, and the ESTALE errno value is returned to the client process. An unexported filesystem message is also written to the server console and error log.

An NFS server must run the mountd daemon. If a server does not run the mountd daemon, clients will not be able to access server file systems. Exported file systems are marked by the mountd daemon based on information found in the /etc/exports file. If this file is modified, the next time mountd is asked to process a request (for example, a showmount or remote mount), it updates the export state on the server.

If an administrator removes an entry from the exports file and follows that by executing a showmount -e command, all subsequent NFS access requests for the removed file system or directory are rejected.

NFS server daemons must be started in the correct order, which is portmap, mountd, and nfsd. The nfsd daemon now exits if started before either the portmap or mountd daemons. The nfssetup setup script starts these daemons in the correct order.

3.10.8 Blank Lines in Yellow Pages Map Files

Blank lines in the Yellow Pages map source data files cause errors. Do not include blank lines in these files.

3.11 Standards and External Specifications

The following notes include information on industry standards.

3.11.1 POSIX Interface Support

In ULTRIX-32 Operating System supports the POSIX interface, which can be selected in either of the following ways:

- Use the -YPOSIX option with the cc command.
- Set the PROG_ENV environment variable to POSIX.

See the ENVIRONMENT section of system and library calls for specific POSIX behavior. For further information, see the *POSIX Conformance Document*.

3.11.2 System Calls in the POSIX Environment

When your program is compiled in the POSIX environment, the following system calls behave as described below:

System Call	Behavior in POSIX Environment
chown(2)	The owner argument is of the type uid_t [‡] and the group argument is of the type gid_t.
getuid(2)	The getuid and geteuid functions return values of the type uid_t. The getgid and getegid functions return values of the type gid_t.
kill(2)	The pid argument is of the type pid_t.
umask(2)	The numask argument is of the type mode_t and the umask function returns a value of the type mode_t.
setuid(3)	<p>The setuid function returns a value of the type uid_t and the setgid function returns a value of the type gid_t.</p> <p>In the POSIX or SYSTEM V mode, the following semantics apply when using setuid or setgid functions:</p> <p>If the process is the superuser, then the real, effective, and saved set (as described in execve(2)) user/group IDs are set to uid.</p> <p>If the process is not the superuser, but uid is equal to the real or the saved set user/group ID, then the effective user/group ID is set to uid. The real and saved set user/group IDs remain unchanged.</p>

3.12 Miscellaneous Known Problems

The following are the miscellaneous know base system problems:

- Swap space

For information about increasing swap space, see Section 4.1.

[‡] The _t indicates a primitive system data type that is defined in /usr/include/sys/types.h.

- **dbx**
The “list main” command, where main is a function name, gets a bad line number.
dbx gets stuck with stopped tty output when it is put in the background (bg) and then put in the foreground (fg),
- **df**
The df command when run with only local disks (or with file systems that have less than 10 characters in their name) prints just “Filesystem”.
- **mh**
The send command does not recognize -nowait with “send: -nowait” in .mh_profile.
The send command with the -all option specified and a nonexistent alias file is encountered exits rather than ignores the nonexistent alias file.
- **nroff**
The nroff command with the -h tab option produces different output than on VAX-based systems.
- **SCSI tape**
The SCSI tape driver does not handle an end-of-tape (EOT) condition correctly. Thus, multivolume tar, cpio, and dd tapes cannot be created. Note that the dump, restore, and backup utilities do create multivolume tapes.
- **mt utility**
Because the SCSI tape driver does not return the status that the mt utility is expecting, EOT, Online, and Offline conditions are not reported correctly.
- **Hang up on last close**
In a POSIX environment or SVID environment, the HUPCL (hang up on last close) bit in the c_cflag field does not work correctly. This can be worked around by using the TIOCHPCL ioctl call from the Berkeley environment. For further information, see intro(2) in the *ULTRIX Reference Pages*.

- **Modems**

Limited modem support is provided on `/dev/tty00`. This support consists of only DTR (data terminal ready) and DSR (data set ready). Therefore, the port does not conform to DEC-STANDARD-52, because this CD (carrier detect) and CTS (clear to send) are also needed for conformance.

Because the port can only sense DSR, this is used to realize that a connection is trying to be established. Therefore, a modem connected to this port must be set such that the DSR lead follows carrier. The modem must also be set so that it does not use CTS/RTS flow control, because these leads are not present. See the modem's manual for specifics on setting these attributes.

- **Baud rates on serial ports**

The two serial ports do not support baud rates of 19,200 bits per second or 38,400 bits per second. These values can be set in the terminal structure using an `ioctl`, but their corresponding baud rate value will be undetermined.

- **Crash dump problems**

Do not use the mouse while the system is crash dumping; this will cause the system to panic again and lose the dump.

- **TZK50 error**

The TZK50 can lose filemark position information when spacing over multiple large files. This seems to be most noticeable when using the `setld` command to load subsets in a random order. To work around this problem, do not specify the `-a` option with the `setld` command. Rather, use the `-l` option with the `setld` command to read the files in the order in which they occur on the tape and, thus, minimize the amount of tape movement.

ULTRIX Documentation Notes 4

The notes in this chapter pertain to the ULTRIX documentation that was shipped as part of your ULTRIX Worksystem Software, Version 2.0 (RISC) or ULTRIX-32, Version 3.0 (RISC) system. This chapter discusses:

- Revised and new documents
- Reference pages
- Undocumented library functions

4.1 Revised and New Documents

The ULTRIX documentation has been revised to accommodate the changes required by the new RISC architecture. Therefore, any document that contains information that is specific to your RISC-based system has the phrase “for RISC Processors” as part of its title. Those documents that do not have this phrase in their title contain information that is relevant to both RISC-based and VAX-based systems. The documents that are specific to your RISC-based system are:

- *Release Notes for RISC Processors*
- *Documentation Overview for RISC Processors*
- *Technical Summary for RISC Processors*
- *Guide to Languages and Programming for RISC Processors*
- *Advanced Installation Guide for RISC Processors*
- *Introduction to System and Network Management for RISC Processors*
- *Guide to System Environment Setup for RISC Processors*
- *Guide to System Configuration File Maintenance for RISC Processors*
- *Guide to System Shutdown and Startup for RISC Processors*
- *Guide to System Backup and Restore for RISC Processors*
- *Guide to System Disk Maintenance for RISC Processors*
- *Guide to System Crash Recovery for RISC Processors*

- *Guide to the Error Logger System for RISC Processors*
- *Guide to Networking for RISC Processors*
- *Guide to Ethernet Communication Servers for RISC Processors*

The following documents have been revised to make applicable to both the RISC and VAX architectures:

- *Guide to Remote Installation Service for RISC and VAX Processors*
- *Guide to Diskless Management Services for RISC and VAX Processors*
- *Guide to Server Setup for RISC and VAX Processors*

In addition, three new ULTRIX documents have been created for your ULTRIX Worksystem Software, Version 2.0 (RISC) or ULTRIX-32, Version 3.0 (RISC) system:

- *Documentation Overview for RISC Processors*
- *Introduction to the ULTRIX Worksystem Software Environment*
- *Guide to Languages and Programming for RISC Processors*

4.2 Reference Pages

The *ULTRIX Reference Pages* documents now include pages for the graphics windowing software as well as RISC-specific and VAX-specific versions of the operating system commands, system calls, library functions, and so on. Your online reference pages, however, include pages for the graphics windowing software (1X) but have been custom-fit for your RISC-based system. That is, no VAX-specific pages have been included with your online reference pages.

In addition, some reference pages had changes made to them after their respective manuals went to print. Therefore, for the following reference pages, you should refer to the online versions for the most up-to-date information:

- `access(2)`
- `chmod(2)`
- `connect(2)`
- `mkdir(2)`
- `mknod(2)`
- `mount(2)`
- `swapon(2)`
- `unlink(2)`
- `utime(2)`

4.3 Undocumented Library Functions

The following C library functions, which can be used to convert floating values between VAX and IEEE formats, are not documented in the *ULTRIX Reference Pages*.

```
int ftoi(value)
    float *value;
```

The ftoi function converts the specified VAX ffloat number to IEEE single-precision format. It returns zero if successful and nonzero without performing the conversion if not successful (for example, underflow).

```
int itof(value)
    float *value;
```

The itof function converts the specified IEEE single-precision number to VAX ffloat format. It returns zero if successful and nonzero without performing the conversion if not successful (for example, overflow).

```
int dtoi(value)
    double *value;
```

The dtoi function converts the specified VAX dfloat number to IEEE double-precision format. It returns zero if successful and nonzero without performing the conversion if not successful (for example, underflow).

```
int itod(value)
    double *value;
```

The itod function converts the specified IEEE double-precision number to VAX dfloat format. It returns zero if successful and nonzero without performing the conversion if not successful (for example, underflow or overflow).

```
int gtoi(value)
    double *value;
```

The gtoi function converts the specified VAX gfloat number to IEEE double-precision format. It returns zero if successful and nonzero without performing the conversion if not successful (for example, underflow).

```
int itog(value)
    double *value;
```

The itog function converts the specified IEEE double-precision number to VAX gfloat format. It returns zero if successful and nonzero without performing the conversion if not successful (for example, underflow).

Reader's Comments

Note: This form is for document comments only. DIGITAL will use comments submitted on this form at the company's discretion. If you require a written reply and are eligible to receive one under Software Performance Report (SPR) service, submit your comments on an SPR form.

Did you find this manual understandable, usable, and well-organized? Please make suggestions for improvement. _____

Did you find errors in this manual? If so, specify the error and the page number.

Please indicate the type of user/reader that you most nearly represent.

- Assembly language programmer
- Higher-level language programmer
- Occasional programmer (experienced)
- User with little programming experience
- Student programmer
- Other (please specify) _____

Name _____ Date _____

Organization _____

Street _____

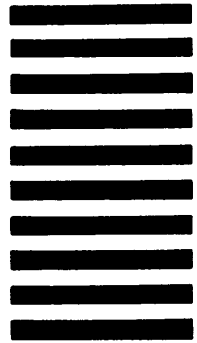
City _____ State _____ Zip Code _____
or _____
Country _____

--- Do Not Tear - Fold Here and Tape ---

digital™



No Postage
Necessary
if Mailed
in the
United States



BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 33 MAYNARD MASS.

POSTAGE WILL BE PAID BY ADDRESSEE

Digital Equipment Corporation
Documentation Manager
ULTRIX Documentation Group
ZKO3-3/X18
110 SPIT BROOK ROAD
NASHUA, NH 03062-9987



--- Do Not Tear - Fold Here ---

Cut Along Dotted Line