

**ULTRIX-32  
Guide to the  
Error Logger System**

Order No. AA-ME95A-TE

---

ULTRIX-32 Operating System, Version 3.0

Digital Equipment Corporation


Copyright © 1987, 1988 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	Q-bus	VAX
DECnet	RT	VAXstation
DECUS	ULTRIX	VMS
MASSBUS	ULTRIX-11	VT
MicroVAX	ULTRIX-32	ULTRIX Worksystem Software
PDP	UNIBUS	

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

IBM is a registered trademark of International Business Machines Corporation.

MICOM is a registered trademark of Micom System, Inc.

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

# Contents

## About This Manual

Audience .....	vii
Organization .....	vii
Conventions .....	vii

## 1 Error Logger Overview

1.1 Error Logger Components .....	1-1
1.1.1 The Kernel Buffer .....	1-2
1.1.2 The elcsd Daemon .....	1-2
1.1.3 The elcsd.conf File .....	1-2
1.2 Error Logger Operation .....	1-2
1.2.1 Local System Crash .....	1-3
1.2.2 Remote System Shutdown .....	1-4
1.2.3 Remote System Crash .....	1-4

## 2 Error Logger Management and Maintenance

2.1 Manual Control of Error Logger Functions .....	2-1
2.1.1 Disabling Error Logging .....	2-1
2.1.2 Enabling Error Logging .....	2-2
2.1.3 Enabling Error Logging in Single-User Mode .....	2-2

2.1.4 Clearing the Errorlog Buffer .....	2-3
2.1.5 Logging Messages to the Errorlog File .....	2-3
2.1.6 Restart the elcsd Daemon .....	2-3
2.2 Reconfiguration .....	2-3
2.3 elcsd.conf File Parameters .....	2-5
2.3.1 Status .....	2-5
2.3.2 Errorlog File Size .....	2-6
2.3.3 Errorlog Directory Path .....	2-6
2.3.4 Backup Errorlog Directory Path .....	2-6
2.3.5 Single-User Errorlog Directory Path .....	2-7
2.3.6 Remote Hosts Errorlog Directory Path .....	2-7
2.3.7 Remote Host Name To Log To .....	2-7
2.3.8 Hosts To Log .....	2-7
2.3.9 Sample Server and Client elcsd.conf Files .....	2-7
2.3.9.1 Server system file entries .....	2-8
2.3.9.2 Client system file entries .....	2-9
2.4 Error Logger File Maintenance .....	2-10
2.4.1 The elcsdlog File .....	2-10
2.4.2 Backing up the Errorlog File .....	2-10
2.4.3 Clearing an Errorlog File .....	2-10
2.5 Crucial Error Logger System Messages .....	2-12

### **3 The Error Report Formatter**

3.1 Generating Reports .....	3-2
3.1.1 Formatting Reports .....	3-3
3.1.2 Generating Reports from Specific Files .....	3-5
3.1.3 Generating Reports from Specific Host Systems .....	3-6
3.1.4 Reporting Errors As They Occur .....	3-6
3.1.5 Generating Reports in Reverse Chronological Order .....	3-6
3.1.6 Generating Reports in Single-User Mode .....	3-6
3.1.7 Generating Summary Reports .....	3-6
3.2 Selecting Error Types .....	3-7

3.2.1	Selecting Adapter Errors .....	3-7
3.2.2	Selecting Class Errors .....	3-8
3.2.3	Selecting Disk Errors .....	3-8
3.2.4	Selecting Mainframe Errors .....	3-8
3.2.5	Selecting Operating System Errors .....	3-9
3.2.6	Selecting Tape Errors .....	3-9
3.3	Selecting Specific Errors .....	3-9
3.3.1	Selecting Errors by Record Type .....	3-10
3.3.2	Selecting Errors by Sequence Number .....	3-11
3.3.3	Selecting Errors Using a Time Range .....	3-11
3.3.4	Excluding Error Types .....	3-12
3.4	Using uerf Options .....	3-13
3.5	uerf Option Reference .....	3-13

## Examples

2-1:	The elcsd.conf File with Default Entries .....	2-4
2-2:	elcsd.conf File Entries for a Server System .....	2-8
2-3:	elcsd.conf File Entries for a Client System .....	2-9
3-1:	The Help Display .....	3-3
3-2:	Memory Error in Brief Format .....	3-4

## Tables

3-1:	The uerf Command Options .....	3-13
------	--------------------------------	------



# About This Manual

This guide describes how the ULTRIX Error Logger System records and reports errors and other events occurring on your MicroVAX or VAX system.

## Audience

The audience for this manual is anyone who manages error information on the system. While this person is usually the system manager, others such as any DIGITAL service representative can use the Error Logger System for analyzing error information to help identify the cause of problems in the system. This guide assumes that the reader is familiar with basic ULTRIX-32 system functions.

## Organization

- Chapter 1      Error Logger Overview  
                  Describes the components and operation of the error logger system.
- Chapter 2      Error Logger Management and Maintenance  
                  Describes how to control error logger functions and how to reconfigure the error logger. It also describes file parameters and file maintenance procedures.
- Chapter 3      Error Report Formatter - uerf  
                  Describes how to generate and format error reports. It also describes how to select specific error types for specific purposes.

## 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.

- command(x)** In text, cross-references to the command documentation include the section number in the reference manual where the commands are documented. For example: See the **cat(1)** command. This indicates that you can find the material on the **cat** command in Section 1 of the reference pages.
- UPPERCASE** The ULTRIX-32 system differentiates between lowercase and uppercase characters. Enter uppercase characters only where specifically indicated by an example or a syntax line.
- italics* In syntax descriptions, this type indicates terms that are variable.
- example** In examples, computer output text is printed in this type.
- example** In examples, user input is printed in this bold type.
- #** This is the default superuser prompt.
- >>>** This is the console subsystem prompt.
- .** In examples, a vertical ellipsis indicates that not all of the lines of the example are shown.
- <KEYNAME>** In examples, a word or abbreviation in angle brackets indicates that you must press the named key on the terminal keyboard.



# Error Logger Overview 1

The error logger is a system management utility designed to help you and your DIGITAL field service representative analyze stored error information. The error logger records and reports errors and other events that occur on your MicroVAX or VAX system. Together with the error report formatter, you can produce error reports that will help you identify the cause of problems with your system.

This chapter describes the error logger system. It covers:

- Error logger components
- Error logger operation

Chapter 2 describes tools and files that you can use to control error logger functions manually. These functions include reconfiguring the errorlog configuration file, enabling error logging in single-user mode, and clearing the kernel buffer.

The error logger system also contains the ULTRIX error report formatter, `uerf`, which allows the user to generate and format several specific types of error reports. Information on `uerf` is contained in Chapter 3.

In all discussions of the error logger, the following definitions apply:

server: The system that receives error messages from other systems.

client: Any system that logs messages to the server.

## 1.1 Error Logger Components

The error logger consists of:

- A kernel buffer that automatically logs error messages in binary form
- The `elcsd` daemon, which transfers error messages from the kernel errorlog buffer to an errorlog file (local or remote)
- The `elcsd.conf` file, which contains configuration parameters for the `elcsd` daemon

The error logging system files reside in the `/etc` directory.

### 1.1.1 The Kernel Buffer

When you start up the system, the error logger automatically logs all error messages in binary form to an errorlog buffer in the kernel. These messages are stored in the kernel errorlog buffer temporarily until the elcsd daemon transfers them to an errorlog file on disk, where they are stored permanently.

The part of the error logger that sends error messages to the kernel errorlog buffer cannot be disabled.

### 1.1.2 The elcsd Daemon

The elcsd daemon is invoked when the system starts up in multiuser mode. The daemon transfers error messages from the kernel errorlog buffer to the errorlog file specified in the elcsd.conf file until you shut the system down.

You can start and stop the elcsd daemon by using the eli command. However, the elcsd daemon does not run when the system is in single-user mode. During single-user mode you must use the eli command with the -s option to run the elcsd daemon and transfer error messages to the errorlog file.

There are other occasions when you will want to enable or disable the elcsd daemon manually. See Chapter 2 for a discussion of the eli command and its options.

### 1.1.3 The elcsd.conf File

The elcsd.conf file contains the default information used by the elcsd daemon to configure error logging. The parameters in the elcsd.conf file determine the names and locations of the main, backup, and single-user errorlog files, the size limit for the files, and provide information on how messages are logged between systems.

The default errorlog file is */usr/adm/syserr/syserr.hostname*.

See Chapter 2 for a discussion of the elcsd.conf file and its parameters.

## 1.2 Error Logger Operation

When you start the system in multiuser mode, the elcsd daemon is invoked by an entry in the */etc/rc* file and begins logging error messages to the errorlog file. If you delete this entry, the elcsd daemon will not transfer error messages from the kernel errorlog buffer to the errorlog file. Here is the entry as it appears in the */etc/rc* file:

```
[ -f /etc/elcsd ] && {
    /etc/elcsd & echo 'start errlog daemon - elcsd' >/dev/console
}
```

See the rc(8) command in the ULTRIX Reference Pages for more information on the /etc/rc file.

The default errorlog directory path for multiuser error logging is /usr/adm/syserr. To change the default directory path, or any other parameters in the elcsd.conf file, see Section 2.3.

The following sections describe how the error logger operates when:

- The local system crashes
- The remote system crashes
- The remote system shuts down

### 1.2.1 Local System Crash

When a system crashes, the error messages in the kernel errorlog buffer are saved by an entry in the /etc/rc.local file. When the system comes back up, the messages are appended to the errorlog file. The default entry in the rc.local file is:

```
/etc/savecore /usr/adm/crash >/dev/console
```

This entry saves the core dump and the messages in the kernel errorlog buffer after a system crash. The core dump (vmcore and vmunix) is stored in the directory /usr/adm/crash. The kernel errorlog buffer data will be stored in the file /usr/adm/syserr/elbuffer temporarily. When the elcsd daemon is started, it appends the error messages to the errorlog file and removes the elbuffer file. See the Guide to System Crash Recovery for more information on savecore and system crashes.

If you do not have adequate disk space to save the system core dump, add the -e option to the savecore entry in rc.local. This entry saves only the data in the kernel errorlog buffer:

```
/etc/savecore -e /usr/adm/crash >/dev/console
```

#### Note

You must have a savecore entry in the rc.local file to save data in the kernel errorlog buffer after a crash. If you do not, you may not be able to determine why the system crashed.

### 1.2.2 Remote System Shutdown

If client systems are logging errors to a server and the server system is shut down for maintenance or repairs, the client system logging to the host is automatically informed of the shutdown. The client system then begins logging errors to its own errorlog file. When the server system comes back up, the client is informed that service has been restored and resumes logging error messages to the server.

After a shutdown occurs, there are two files containing client messages:

- The errorlog file on the server system
- The errorlog file on the client system

### 1.2.3 Remote System Crash

If client systems are logging errors to a server and the server system crashes, the client system is not informed of the crash and continues trying to log errors to the server. Because the server is down, these messages are not saved. When the server system reboots, the error messages from the client are again logged in the errorlog file on the server system.

If the server system is going to be down for some time, you can reconfigure the client remote system by changing its `elcsd.conf` file to log locally by changing `/etc/elcsd.conf`. Then use the `eli` command with the `-r` option to reconfigure error logging on the server.

If a client system goes down while logging errors to the server system, the server system is not affected. When the client system comes back up, it automatically begins sending error messages to the server again.

# Error Logger Management and Maintenance 2

This chapter describes tools and procedures used to manage and maintain the error logger system. It contains information on:

- Manual control of error logger functions
- Reconfiguration
- Error logger file maintenance
- Error messages

## 2.1 Manual Control of Error Logger Functions

The `eli` command lets you manually control these error logger functions:

- Enable and disable error logging
- Clear the errorlog buffer
- Log messages to the errorlog file
- Restart the `elcsd` daemon

The following sections describe how to use `eli` to perform these functions.

Refer to `eli(8)` in the ULTRIX Reference Pages for more information on the command options.

### 2.1.1 Disabling Error Logging

To disable error logging to an errorlog file, type:

```
# /etc/eli -d
```

You can use this option when your file system is full while you free up more disk space. However, when you use this option, the `elcsd` daemon stops logging error messages. Messages will queue up in the kernel buffer until the buffer becomes full. These messages will be logged to disk when the error log daemon is restarted.

### 2.1.2 Enabling Error Logging

To re-enable error logging in multiuser mode, type:

```
# /etc/eli -e
```

or:

```
# /etc/eli -f -e
```

When you use the `-f` option, the system suppresses the prompts you usually see when you select the `-e` option. The `elcsd` daemon will create a new errorlog file. The default pathname will be `/usr/adm/syserr/syserr.hostname`.

### 2.1.3 Enabling Error Logging in Single-User Mode

Error messages can be logged during single-user mode and multiuser mode. In single-user mode, however, you must start up the `elcsd` daemon manually to log error messages to the errorlog file.

Before starting the `elcsd` daemon in single-user mode, run `fsck` on the file system you will be logging to check for file system inconsistencies. The root file system is the default file system for error logging in single-user mode. Then invoke the `eli` command by typing:

```
# eli -s
```

The `-s` option enables error logging in single-user mode. Error messages are transferred from the kernel errorlog buffer to the file specified in the `elcsd.conf` file for messages generated during single-user mode.

The root (`/`) directory is the default location for the single-user errorlog file. When the system makes the transition to multiuser mode, by default, it appends the messages in `/syserr.hostname` to the end of the multiuser errorlog file, `syserr.hostname`, in `/usr/adm/syserr`. If you do not want to save error messages generated during single-user mode, remove the single-user errorlog file before you make the transition to multiuser mode. To change the single-user directory path or any other parameters in the `elcsd.conf` file, see Section 2.3.

### 2.1.4 Clearing the Errorlog Buffer

To clear the kernel errorlog buffer, type:

```
# /etc/eli -i
```

This command line initializes the kernel errorlog buffer. The messages in the kernel errorlog buffer are cleared without being logged to disk.

If a problem occurs anywhere in the system that stops the elcsd daemon from running, an error message informing you that error messages are not being logged to an error log file appears on the console every 15 minutes. To stop this message, type:

```
# /etc/eli -q
```

To re-enable logging this missed-error message to the console, type:

```
# /etc/eli -w
```

### 2.1.5 Logging Messages to the Errorlog File

You can log a one-line message to the errorlog file with this command line:

```
# /etc/eli -l
```

The system prompts you to enter the message. You can also log a message that is contained in a file. For example, using the file `myfile`, the command line entry is;

```
# eli -f -l < myfile > /dev/null
```

This example logs a message, up to and including the first newline, from the file named `myfile`. The example also directs the `eli` prompt to `/dev/null`.

### 2.1.6 Restart the elcsd Daemon

If you change any entries in the `elcsd.conf` file, you must restart the `elcsd` daemon so that the new parameters will take effect. Type:

```
# eli -r
```

## 2.2 Reconfiguration

Reconfiguration is the process of disabling the error logger, modifying the errorlog configuration file, and then restarting the error logger. This section describes the error logger configuration file, `elcsd.conf`. The `elcsd.conf` file contains the information used by the `elcsd` daemon to configure error logging. The entries in the `elcsd.conf` file direct local error logging as well as error logging between systems.

Anyone can view the `elcsd.conf` file, but you must be superuser to change the entries in the file. Whenever you change the entries in the `elcsd.conf` file, use the `eli` command with the `-r` option to reconfigure the `elcsd` daemon.

You can log error messages between systems by setting up an errorlog file on a server system to receive error messages for a client system. For example, if you have a MicroVAX system with limited disk space, you can log its error messages to a larger system. However, you must specify in the `elcsd.conf` files on both systems how you want to log errors. The client `elcsd.conf` file must specify where messages are to be sent, and the server `elcsd.conf` file must have an entry naming the client system it will accept messages from. Entries in the server `elcsd.conf` file must also specify the pathname for the client messages file.

To enable error logging between systems, you must have the following entry in the `/etc/services` file for both server and client systems:

```
elcsd    704/udp    #error log daemon
```

### Note

You can only log errors to another system when the systems share the same network. See the Guide to Networking for information on networks.

Example 2-1 shows an `elcsd.conf` file with default entries. These entries allow local logging only.

### Example 2-1: The `elcsd.conf` File with Default Entries

```
#static char *sccsid = "@(#)elcsd.conf (ULTRIX) ";
#
#   elcsd - errlog configuration file
#
{
    # delimiter DON'T remove or comment out!
    1      # 1-local,2-logrem,4-remlog,5-remlog+priloglocal
    # errlog file size limit num. of blocks
/usr/adm/syserr      # errlog dir. path
                    # backup errlog dir. path
/                    # single user errlog dir. path
/usr/adm/syserr      # log remote hosts errlog dir. path
                    # remote hostname to log to
}
# delimiter DON'T remove or comment out!
# hosts to log :S - separate file or :R - remotes file (together)
#remotel:S          - (example) log errors from remotel into separate file
```



## 2.3 elcsd.conf File Parameters

Here are the parameters for the elcsd.conf file:

- Status
- Errorlog file size
- Errorlog directory path
- Backup errorlog directory path
- Single-user errorlog directory path
- Remote hosts errorlog directory path
- Remote host name to log to
- Hosts to log

The entries in Example 2-1 appear at the left-hand margin, and the parameters are defined to the right. The following sections discuss each of the elcsd.conf file parameters.

### 2.3.1 Status

The first parameter in the elcsd.conf file is the status line, which indicates where error messages are logged.

The possible status conditions and their meanings are:

1-local	Log local messages here.
2-logrem	Log messages from remote systems here.
3	Log local and remote messages here.
4-remlog	Log messages from this system to a remote system.
5-remlog+priloglocal	Log messages from this system to a remote system, and log server as well as high priority messages here.

When you log error messages between systems, you must decide how to log the different priority levels of errors, based on the amount of disk space you have available.

There are three priority levels: severe, high, and low. A severe error means the system is going down. High-priority errors are errors such as recoverable machine checks and hard errors. Low-priority errors include soft errors, restarts and CRDs (corrected read data).

The default status condition is 1 (local). Errors that occur on the local system are logged on the local system.

Enter 2 (logrem) on the status line if you are a server system and you want to log messages from a client system.

Enter 3 on the status line to log server (local) and client (remote) messages.

Enter 4 (remlog) on the status line if you are a client system and you want to log messages to a server.

Enter 5 (remlog+priloglocal) on the status line if you have sufficient disk space and, in addition to logging all messages to a server, want to log severe and high-priority messages on your client system.

No matter what status entry you specify, severe kernel messages are sent to the local system console.

### **2.3.2 Errorlog File Size**

The errorlog file size parameter defines the maximum size of every errorlog file on the system. You can leave the errorlog file size parameter blank, and the system will notify you when the file system is 98% full.

Otherwise, specify the maximum number of blocks (for example, 512 bytes) that you want for each errorlog file.

When you do not specify the size of the errorlog file, a message is written to the errorlog status file, `/usr/adm/elcsdlog`, when the file system is 98% full. At this point, you should compress or back up and remove the current errorlog file before the system stops logging messages to disk. A message is also written to the system log. See the `syslog(8)` command in the ULTRIX Reference Pages.

### **2.3.3 Errorlog Directory Path**

To start an errorlog file, you must specify a directory path in the `elcsd.conf` file.

The errorlog directory path specifies the main errorlog file. The default path is `/usr/adm/syserr`. You can, however, direct error messages to a different directory. If you do, first verify that the alternate directory exists and then specify this alternate when you invoke the `uerf` command.

### **2.3.4 Backup Errorlog Directory Path**

The backup errorlog directory path is the file the system writes to when the main errorlog file becomes full or when there is an error and the system cannot access the main file. There is no default path.

Consequently, you should specify a backup errorlog directory that is different from the main errorlog directory.

### 2.3.5 Single-User Errorlog Directory Path

The single-user errorlog file is */syserr.hostname*. The root (/) directory is the default location for this file. You can direct single-user error messages to a different directory, but you must be sure that the directory you specify is mounted during single-user mode. When the system makes the transition from single-user mode to multiuser mode, errors logged in */syserr.hostname* are appended to *syserr.hostname* in the multiuser errorlog directory path and the single-user errorlog file is removed.

### 2.3.6 Remote Hosts Errorlog Directory Path

The remote hosts errorlog directory path is the directory for the errorlog file containing messages logged from remote, or client systems. The default path is */usr/adm/syserr*. It is a good idea to use the same path name that you used for the main errorlog directory.

### 2.3.7 Remote Host Name To Log To

This parameter specifies the name of the remote (server) system that you are logging to. This entry belongs in the client *elcsd.conf* file. There is no default remote host name.

### 2.3.8 Hosts To Log

The hosts to log parameter specifies the name of the remote (client) system that you are logging to, and specifies how to log the client's messages. The parameter takes two arguments:

- The argument, *S*, sets up a separate errorlog file, *syserr.hostname*, for each client system
- The default argument, *R*, logs error messages from all client systems to the *syserr.remotes* file.

### 2.3.9 Sample Server and Client *elcsd.conf* Files

The following examples show and describe the contents of a server *elcsd.conf* file (see Example 2-2) and a client *elcsd.conf* file (see Example 2-3). These examples are based on the following case:

Assume that you want to log error messages to a VAX server, *piano*, from two MicroVAX clients, *flute* and *violin*. You have decided to set up individual errorlog files on the server for each remote client. Plus, since your client systems have adequate disk space, you decide to log severe and high-priority error messages on each client system.

**2.3.9.1 Server system file entries** – To set up the required server files, you edit the server's `elcsd.conf` file as shown below.

### Example 2-2: `elcsd.conf` File Entries for a Server System

```
#static char *sccsid = "@(#)elcsd.conf (ULRIX) ";
#
#           elcsd - errlog configuration file
#
{ # delimiter DON'T remove or comment out!
3 # status 1-local,2-logrem,4-remlog,5-remlog+priloglocal
  # errlog file size limit num. of blocks
/usr/adm/syserr # errlog dir. path
                # backup errlog dir. path
/               # single user errlog dir. path
/usr/adm/syserr # log remote hosts errlog dir. path
                # remote hostname to log to
} # delimiter DON'T remove or comment out!
# hosts to log :S - separate file or :R - remotes file (together)
flute:S
violin:S
```

If you examine the contents of this file, you will notice the following entries:

- The error logging status, 3, indicates that both server and client messages are to be logged to the server, piano.
- The errorlog directory path default, `/usr/adm/syserr`, is specified as the main errorlog directory.
- The remote hosts errorlog directory path default, `/usr/adm/syserr`, is specified as the directory for error messages logged from remote (client) systems.
- Both flute and violin are listed as the hosts to log. Consequently, the VAX server will log messages for each of these client systems. Including the S argument with each host name (flute:S, for example) specifies that each client will have its own errorlog file on the server. These individual files will be subordinate to the remote hosts errorlog directory, `/usr/adm/syserr`. Therefore, error messages for flute will reside on the VAX server at `/usr/adm/syserr/syserr.flute`. Similarly, error messages for violin will reside on the VAX server at `/usr/adm/syserr/syserr.violin`.

**2.3.9.2 Client system file entries** – To set up the required client files, you edit the `elcsd.conf` file for both flute and violin. The following example shows the client file entries.

### Example 2-3: `elcsd.conf` File Entries for a Client System

```
#static char *sccsid = "@(#)elcsd.conf (ULTRIX) ";
#
#       elcsd - errlog configuration file
#
{ # delimiter DON'T remove or comment out!
5 # status 1-local,2-logrem,4-remlog,5-remlog+priloglocal
  # errlog file size limit num. of blocks
/usr/adm/syserr # errlog dir. path
                # backup errlog dir. path
/               # single user errlog dir. path
/usr/adm/syserr # log remote hosts errlog dir. path
piano         # remote hostname to log to
} # delimiter DON'T remove or comment out!
# hosts to log :S - separate file or :R - remotes file (together)
```

If you examine the contents of this file, you will notice the following entries:

- The error logging status, 5, specifies that all error messages are logged to the server, and that severe and high-priority error messages are logged to the client as well.
- The errorlog directory path default, `/usr/adm/syserr`, is specified as the main errorlog directory.
- The remote hosts errorlog directory path default, `/usr/adm/syserr`, is specified as the directory for error messages logged from remote (server) systems.
- The server, `piano` is listed as the remote hostname to log to. Consequently, the client will log messages to this server system.

## 2.4 Error Logger File Maintenance

This section discusses the files related to the error logger and explains how to maintain them. It describes the `elcsdlog` file and explains how to backup and restore an errorlog file.

### 2.4.1 The `elcsdlog` File

The `elcsdlog` file contains status messages about the `elcsd` daemon. It resides in the `/usr/adm` directory. Critical status messages in this file are also logged by the `syslog` utility to the system log and to the terminal of any user entered in the `syslog.conf` file. See `syslog(8)` in the ULTRIX Reference Pages for information about this program.

Critical messages should be acted upon as soon as possible or the system may become unable to log error messages. The `elcsdlog` file is cleared when you reboot the system.

### 2.4.2 Backing up the Errorlog File

You should periodically back up or delete the errorlog files to free up disk space. How often you do this depends on the amount of disk space available and the size of your file system. However, it is good practice to back up the errorlog file at least once a month for a MicroVAX system and at least once every two months for a VAX system. See Guide to System Backup and Restore for information on backing up files as part of normal system maintenance.

#### Note

When you do backups of any kind, be sure that the `elcsd` daemon is running. Do this with the command:

```
# ps -ax | grep elcsd
```

If the `elcsd` daemon is not running, start it manually with the `-s` option of the `eli` command for single-user mode, or the `-e` option for multi-user mode.

### 2.4.3 Clearing an Errorlog File

In addition to backing up your errorlog files, you can clear individual files

periodically. The following steps describe the procedure:

### Note

You do not need superuser privileges for every command shown below. It makes the task easier, however, if you simply log on and work as root until you complete the entire procedure.

1. Change your working directory to the main errorlog directory and get a listing of the contents. For example, type:

```
# cd /usr/adm/syserr;ls
```

2. Locate the errorlog file you intend to clear. For example, if the hostname is piano, you are looking for the file named `syserr.piano`.
3. Copy the errorlog file to a new location, and rename the file. For example, type:

```
# cp syserr.piano /usr/adm/copies/syserr.old.piano
```

You now have two copies of the errorlog file. The error logger program will continue to log error messages to the original file, `/usr/adm/syserr/syserr.piano` but not to the copy, `/usr/adm/copies/syserr.old.piano`.

### Note

Do not use the `mv` command to save the errorlog file, `/usr/adm/syserr.hostname`, or the system will continue to log error messages to that file in its new location.

You can compress the saved file to save disk space. See the `compact(1)` command in the ULTRIX Reference Pages for details.

If you want the data in the saved file to remain accessible, enter the `uerf` command with the `-f` option and the full pathname of the saved file. For example, type:

```
# uerf -f /usr/adm/copies/syserr.old.piano
```

In response, the error logger displays the contents of the saved file.

4. Remove the original errorlog file, `/usr/adm/syserr/syserr.piano` with the command:

```
# rm /usr/adm/syserr/syserr.piano
```

5. To re-enable error logging, type:

```
# eli -f -e
```

When you use the `-f` option, the system suppresses the prompts you usually see when you select the `-e` option. The `elcsd` daemon creates a new errorlog file, at `/usr/adm/syserr/syserr.hostname`.

## 2.5 Crucial Error Logger System Messages

There are three crucial messages that indicate a problem in the error logging procedure:

1. When the kernel identifies an overflow of its errorlog buffer, it generates the following error message which is sent to the system console every 15 minutes:

```
Errlog Buffer Full: <n> Missed Error(s)
```

If you receive this message from the kernel, you can respond by typing the `ps` command to see if the `elcsd` daemon is running.

- If it is not running, restart it using the `eli` command with the `-e` option. If you are in single-user mode, use the `eli` command with the `-s` option.
  - If the `elcsd` daemon is running, then the current system tasks may be causing a high volume of errors, which the `elcsd` daemon cannot move from the kernel errorlog buffer to the errorlog file quickly enough. You may want to suppress the printing of this message using the `eli` command with the `-q` option until these tasks have finished.
2. When the space needed for error messages is about to exceed the limit set in the `elcsd.conf` file, the system writes the following message to the `/etc/syslog.conf` and `/etc/elcsdlog` files:

```
exceeded errlog file limit size, error not logged to disk!
```

You can respond to this problem as follows:

- Save and then remove the current errorlog file. See the previous section, Clearing an Errorlog File, for instructions on how to do this.
- Increase the size limit in the `elcsd.conf` file. You should set the limit based on the disk device and the size of the file system. See the previous section, Errorlog File Size, for details.



- Reconfigure the `elcsd.conf` file by using the `eli` command with the `-r` option.
- 3. When the system is about to run out of file system space and is no longer logging error messages to disk, it writes the following message to the `/etc/syslog.conf` and `/etc/elcsdlog` files:

```
file system almost full, CAN'T log errors to \  
/usr/adm/syserr/syserr.hostname
```

This message can occur when the errorlog file size limit in the `elcsd.conf` is not specified. You can respond to this problem as follows:

- Find out why the file system is full. It may be necessary to remove or compress other files on the disk or to save and then remove the current errorlog file. See the previous section, `Clearing an Errorlog File`, for more information on how to perform these tasks.
- Consider setting the errorlog file size limit in the `elcsd.conf` file. You should set the limit based on the disk device and the size of the file system. See the previous section, `Errorlog File Size`, for details.
- If you change the `elcsd.conf` file, you must reconfigure it by using the `eli` command with the `-r` option.



## The Error Report Formatter 3

The ULTRIX error report formatter, `uerf`, reports errors and events that occur on your MicroVAX or VAX system. The `uerf` program accesses the error messages and events stored in the errorlog file, translates them from binary code to ASCII, and sends them to the output device you specify.

The `uerf` program uses the data files, `uerf.bin`, `uerf.hlp` and `uerf.err`. The `uerf.bin` file is the event information data base, `uerf.hlp` is the help file, and `uerf.err` is the error message file.

The `uerf` program searches sequentially for the `uerf` data files. It looks:

1. In the directory specified when using a full pathname with the `uerf` command.
2. In the `/etc` directory.
3. In the directories specified in your shell `PATH` environment variable.

The `uerf` command has options that let you generate various types of reports on specified errors. For example, you can produce a report containing all the error messages in the errorlog file, or you can use options to produce a report with only a few error types. In relation to these options, this chapter provides information on:

- Generating reports
- Selecting error types
- Selecting specific errors

For your convenience, this chapter also contains a quick reference for `uerf` options.

### Note

You do not need superuser privileges to use the `uerf` command.

### 3.1 Generating Reports

The `uerf` program produces reports based on the error messages stored in an errorlog file. Using the `uerf` command with its options, you can produce error reports to a terminal screen, to a file, or to a printer. For example, to generate an error report and display it on your terminal one screenful at a time, type:

```
# /etc/uerf | more
```

By looking at the types and number of errors and events, you can tell how reliable the system is. For example, if a report shows a large number of errors for a particular device, you can call a DIGITAL service representative before the device fails. Furthermore, when a failure does occur, the error report provides information on the events that led up to the failure.

The error logger system records and reports the following errors and events:

- Errors – Device errors, device timeouts, machine checks, bus errors, and memory errors, such as hard or soft error correcting code (ECC) errors
- Events – System startup and shutdown messages, system failures or panics, and informational messages

To see what options are available with the `uerf` command, use the help option:

```
# /etc/uerf -h
```

This command displays a list of the available options and their meanings. Example 3-1 shows the `uerf` help display.

The rest of this section explains the `uerf` options. These options enable you to:

- Format reports
- Generate reports from specific files
- Generate reports from specific host systems
- Report on errors as they occur
- Generate reports in reverse chronological order
- Generate reports in single-user mode
- Generate summary reports

For a detailed list of options in quick reference format, refer to the `uerf` Option Reference in Section 3.5 or see `uerf(8)` in the ULTRIX Reference Pages.

## Example 3-1: The Help Display

```

                                U E R F   H E L P

uerf<cr>          - process all events in the default input file

PROCESSING OPTIONS

-h                - display this help screen.
-f input_filename - process events from the specified input file.
-n                - process events from kernel errorlog buffer immediately.
-H ascii_host_name - process events for the specified hosts.
-R                - process events from the errorlog file in reverse order.
-t time_range     - process events in the specified time range.
-o output_format  - format the output in brief, full or terse format.
-S                - display a summary of selected events.
-x                - exclude selected events (below) from output.
-Z                - display the entry in hex format

SELECTION OPTIONS - Events may be selected as follows.

-A adapter_list      -D disk_devices          -M mainframe_type
-O os_specific        -T tape_devices          -c class_of_events
-e error_types        -r record type           -s sequence number

***** For a detailed list of options see uerf(8) *****
```

### 3.1.1 Formatting Reports

Use the `-o` option to format `uerf` error reports. This option enables you to format reports in three ways:

- brief Reports event information in a short format
- full Reports all available information for each error
- terse Reports binary event information and displays register values and other ASCII messages in a condensed format

This option requires a parameter. If you do not select the `-o` option, the default output format is brief.

Some errors, such as ASCII messages, produce the same information whether you use the full format or the brief format. Errors that do provide more information in the full format are: machine checks, panics, memory errors, MSCP disk errors, and TMSCP tape errors.

The brief output format is shown in Example 3-2 for this command line:

```
# /etc/uerf -M mem
```

### Example 3-2: Memory Error in Brief Format

```
***** ENTRY                2 *****
----- EVENT INFORMATION SEGMENT -----
EVENT CLASS                      ERROR EVENT
OS EVENT TYPE                    MEMORY ERROR
SEQUENCE NUMBER                  101.
OPERATING SYSTEM                 ULTRIX 32
OCCURRED/LOGGED ON              Thu May 14 18:14:46 1987 EDT
OCCURRED ON SYSTEM             guitar
SYSTEM ID                       x01845106
PROCESSOR TYPE                 KA780
----- UNIT INFORMATION -----
UNIT CLASS                      MEMORY
UNIT TYPE                      MS780E
ERROR SYNDROME                 MEMORY CRD ERROR
----- 780-E CSR REGS -----
CSRA                            x00101E6C  INTERLEAVE:  INTERNAL 2-WAY
                                RAM: 64K
                                ADAPTER CODE:  x3
                                MEMORY SIZE:  15.
                                ERROR SUMMARY
CSR B                            x00001000  DIAG ECC BITS:  x0
                                MEM HAS VALID DATA
                                START ADDR:  x0
CSR C                            x148F020E  ERR SYNDR/CHK BITS:  xE
                                CRD
                                ERROR ADDR:  x91E0
                                ERROR LOG REQUEST
CSR D                            x080AD002  ERR SYNDR/CHK BITS:  x2
                                ERROR ADDR:  x1015A
```

To show the full output format for this report, which displays all memory-related mainframe errors, type:

```
# /etc/uerf -o full -M mem
```

In addition to the information shown in Example 3-2, the full output format produces this information:

```
----- ADDITIONAL MEMORY INFO -----  
  
CNTRLR NO                1.  
NO. ERRS ON THIS ADD     1.
```

To produce the terse output format for all memory-related mainframe errors, type:

```
# /etc/uerf -o terse -M mem
```

This report gives you binary event information, for the events you specify. The terse memory-related mainframe errors report looks like this:

```
MEMORY ERROR:           KA780  
MEMORY CRD ERROR       MS780E  
CSRA                    x00101E6C  
CSRB                    x00001000  
CSRC                    x00000000  
CSRD                    x1A0AD275  
CNTRLR NO              1.  
NO. ERRS ON THIS ADD  1.
```

### 3.1.2 Generating Reports from Specific Files

The file name (`-f`) option of `uerf` selects errors from the errorlog file you specify. For example,

```
# /etc/uerf -f /usr/adm/syserr/syserr.guitar.old
```

Use this option to look at old or backup errorlog files rather than at the default file `/usr/adm/syserr/syserr.hostname`. You can also use this option to access the single-user errorlog file, `/syserr.hostname`, in the root directory. Be sure to specify the full path name for the file. Do not use the `-n` option with the `-f` option.

You must specify a parameter with the `-f` option or you will get an invalid parameter message.

### 3.1.3 Generating Reports from Specific Host Systems

The host (-H) option of uerf selects errors for a specific host system from the `/usr/adm/syserr/syserr.hostname` or `/usr/adm/syserr/syserr.remotes` file. Consequently, you must specify a host name as a parameter. This option is useful when you are logging errors from several remote systems to the errorlog file `/syserr.remotes` on the host. See Chapter 2 for information on remote logging.

### 3.1.4 Reporting Errors As They Occur

The now (-n) option of uerf reports errors to the terminal as they occur, before they are logged to the errorlog buffer. Here is the command line:

```
# /etc/uerf -n
```

You can use this option when you run disk or tape exercisers. Do not use the -f option with this option.

### 3.1.5 Generating Reports in Reverse Chronological Order

To generate reports in reverse chronological order, use the -R option of the uerf command. As shown in the following example, you can select the starting point of the report by combining -R with other uerf options.

```
/etc/uerf -R -r 300
```

In response to this command, the uerf program produces a report that lists all startup messages, beginning with the most recent.

### 3.1.6 Generating Reports in Single-User Mode

You can run uerf in single-user mode. However, if the errorlog file and the uerf data files, `uerf.bin`, `uerf.hlp` and `uerf.err`, are not located in the root directory, you must first mount the file systems on which they reside.

In addition, the line printer spooler is not operational during single-user mode. Therefore, to print an error report to a line printer while in single-user mode, you must direct the output to a printer special file, such as:

```
# /etc/uerf -f /syserr.hostname > /dev/lp
```

### 3.1.7 Generating Summary Reports

The summary (-S) option of uerf produces a summary of the errorlog file, or a summary of those records that you select. All uerf selection options support summaries. The default format for summary output is terse. However, you can generate summary output in full or brief format by including the output (-o) option and specifying the desired format.



The following example shows the command that you would type to generate a terse summary of all errors recorded for the day in the errorlog file:

```
# /etc/uerf -t s:00 -S
```

Or, to generate a full summary of the same information, you would type:

```
# /etc/uerf -t s:00 -S -o full
```

## 3.2 Selecting Error Types

The `uerf` command allows you to generate error reports for the following:

- Adapter errors
- Class errors
- Disk errors
- Mainframe processor errors
- Operating system errors
- Tape errors

For each of these error types, the `uerf` command has a corresponding option. Each of the options has corresponding report parameters that you can specify to generate various types of reports.

The following sections explain how to select each of these error types.

### 3.2.1 Selecting Adapter Errors

The `-A` option selects errors for the adapter parameters that you specify. The adapter parameters the error report formatter supports are:

```
aie    BVP-type controller
aio    BVP-type controller
bla    BI LESI adapter
bua    BI UNIBUS adapter
nmi    Nautilus memory interconnect
uba    VAX UNIBUS adapter
```

The following example shows how to generate a report for `uba` (VAX UNIBUS) errors that have been logged in the errorlog file:

```
# /etc/uerf -A uba
```

To generate a report for more than one parameter, use the comma (,) as a delimiter. For example, the following command generates a report for all uba and nmi adapter errors:

```
# /etc/uerf -A uba,nmi
```

### 3.2.2 Selecting Class Errors

The `-c` option selects errors for the class event that you specify. The class parameters that the error report formatter supports are:

- err Hardware-detected and software-detected error conditions
- oper Operational events, such as device status or error messages, time changes, and system startup and shutdown messages
- maint Events that occur during system maintenance, such as running the on-line functional exercisers

### 3.2.3 Selecting Disk Errors

The `-D` option selects errors for the MSCP disk parameters you specify. See the `ra(4)` command in the ULTRIX Reference Pages for the disk types that the error report formatter supports.

### 3.2.4 Selecting Mainframe Errors

The `-M` option selects these types of processor errors for the systems for which you are logging errors:

- cpu CPU-related errors, such as machine checks
- mem memory-related errors, such as single-bit CRD (corrected read data) and double-bit uncorrectable errors

When you do not specify any parameters, all mainframe errors are reported.

The following command produces a report containing CPU-related mainframe errors for the system:

```
# /etc/uerf -M cpu
```

### 3.2.5 Selecting Operating System Errors

The `-O` option selects errors for the operating system parameters you specify. When you do not specify any parameters, all operating system events are reported. The operating system parameters that the error report formatter supports are:

<code>aef</code>	Arithmetic exception faults
<code>ast</code>	Asynchronous trap exception faults
<code>bpt</code>	Breakpoint instruction faults
<code>cmp</code>	Compatibility mode faults
<code>pag</code>	Page faults
<code>pif</code>	Privileged instruction faults
<code>pro</code>	Protection faults
<code>ptf</code>	Page table faults
<code>raf</code>	Reserved address faults
<code>rof</code>	Reserved operand faults
<code>scf</code>	System call exception faults
<code>seg</code>	Segmentation faults
<code>tra</code>	Trace exception faults
<code>xfc</code>	Xfc instruction faults

### 3.2.6 Selecting Tape Errors

The `-T` option selects errors for the TMSCP tape parameters that you specify. See the `tms(4)` command in the ULTRIX Reference Pages for the tape types that the error report formatter supports.

## 3.3 Selecting Specific Errors

Some `uerf` options let you select the following errors specifically:

- Errors specified by record number
- Errors specified by sequence number
- Errors within a specified time range
- Errors except the ones that you specifically exclude

The following subsections describe the `uerf` options that let you select or exclude specific errors.

### 3.3.1 Selecting Errors by Record Type

The `-r` option selects errors for specific record types. Consequently, you must specify a parameter that defines which record type you want the program to select. Use this option to access errors, such as ASCII messages, which are not accessed by other `uerf` options. The `-r` option also offers an alternate way to specify some error events, such as disk and tape errors.

Devices, such as UNIBUS communications devices, log to the `errorlog` file in an ASCII rather than binary message format.

#### Note

It is possible that the text of an ASCII error message, when reported in the brief or the full output format, will generate more than one error message. If another error occurs and is reported at approximately the same time, these ASCII text messages may not print out consecutively. You can use the terse output format to see such messages as one unit.

The record parameters supported by the error report formatter are:

#### Hardware-Detected Errors

- 100 machine check
- 101 memory corrected read data/read data substitute (`crd/rds`)
- 102 disk errors
- 103 tape errors
- 104 device controller errors
- 105 adapter errors
- 106 bus errors
- 107 stray interrupts
- 108 asynchronous write errors
- 109 exceptions/faults
- 112 stack dump
- 113 ka650 error and status registers
- 114 6200 vector 60
- 115 6200 vector 54

#### Software-Detected Errors

- 200 panics (bug checks)
- 201 ci ppd info

## Informational ASCII Messages

250 informational  
251 8600/8650 snapshot taken

## Operational Messages

300 startup  
301 shutdown  
310 time change  
350 diagnostic information  
351 repair information

The following example produces all system startup messages, including hardware devices configured and their CSR (control status register) addresses:

```
# /etc/uerf -r 300
```

The following example outputs all of the operational messages - startup, shutdown, time change, and diagnostic - within the record option:

```
# /etc/uerf -r 300-350
```

### 3.3.2 Selecting Errors by Sequence Number

The `-s` option selects specific errors from the errorlog file. The selection corresponds to the sequence number given as a parameter to the `-s` option. Consequently, you must specify the sequence number. A sequence number is assigned to an event when it is logged to the errorlog file. You can use this option to output specific errors after viewing the errorlog file at your terminal.

#### Note

Sequence numbers are restarted when the system is rebooted. If the errorlog file contains error messages from before and after a reboot, there may be duplicate sequence numbers in the file.

### 3.3.3 Selecting Errors Using a Time Range

The `-t` option selects errors for the time period you specify. The format for the `-t` option is:

```
/etc/uerf -t s:dd-mmm-yyyy, hh:mm:ss e:dd-mmm-yyyy, hh:mm:ss
```

You must specify a start date or time or an end date or time when you use this option. However, you can abbreviate the format using the following default parameters:

- The current date is the default date when none is specified.
- The default start time is 00:00:00.
- The default end time is 23:59:59.

When you do not use this option, the `uerf` command processes the entire errorlog file.

The parameters for the `-t` option are:

<code>s</code>	Specifies the start date and time
<code>e</code>	Specifies the end date and time
<code>dd</code>	Day
<code>mmm</code>	Month
<code>yyyy</code>	Year
<code>hh</code>	Hour
<code>mm</code>	Minute
<code>ss</code>	Second

The following example shows how to produce an error report containing all errors for the 24-hour period of October 23, 1988:

```
# /etc/uerf -t s:23-oct-1988,00:00:00 e:23-oct-1988,23:59:59
```

The following command produces errors from the beginning of the errorlog file until February 29 of the current year; the default end time is 23:59:59:

```
# /etc/uerf -t e:29-feb
```

### 3.3.4 Excluding Error Types

The `-x` option excludes the error types that you specify from the error report. Although this option works with most other options, you cannot exclude the `-f` option, the `-h` option, the `-H` option, the `-n` option, the `-o` option, the `-t` option, or the `-R` option. The following command reports all errors except disk errors and operating system events:

```
# /etc/uerf -O -x -o full -D.
```

### 3.4 Using uerf Options

This section provides some notes and examples of uerf use. See Table 3-1 for a quick reference of the command options.

- Be sure to type options in the correct case since the uerf command has both uppercase and lowercase options. For example, `-o` specifies output format, while `-O` specifies operating system events.
- Options can be used together. For example:

```
# /etc/uerf -A uba,nmi -f syserr.guitar.old -TH guitar
```

The preceding command produces an error report from the file `syserr.guitar.old` for the system guitar. The report contains adapter errors for the VAX UNIBUS adapter and the Nautilus memory interconnect, as well as TMSCP (Tape Mass Storage Control Protocol) device errors of all types.

In the following example, the time and output options produce error messages for the current day in a terse format at your terminal:

```
# /etc/uerf -t s:00 -o terse
```

### 3.5 uerf Option Reference

Table 3-1 lists each uerf option alphabetically and provides brief descriptions and examples.

**Table 3-1: The uerf Command Options**

Option	Usage	Examples
- A adapters	Selects adapter and device controller errors.	<code>/etc/uerf - A bua,nmi</code>
- c classes	Selects classes of errors.	<code>/etc/uerf - c err</code>
- D disks	Selects mscp disk devices.	<code>/etc/uerf - D rx</code> <code>/etc/uerf - D ra60,rd31</code>
- f filename	Specifies the file from which error messages are read. Use full pathname. Do not use with the <code>- n</code> option.	<code>/etc/uerf - f</code> <code>/usr/adm/syserr/syserr.old</code>

Option	Usage	Examples
- h	Displays help message. Do not use with another option.	/etc/uerf - h
- H hostname	Selects errors for specified system. Use for remote logging to syserr.remotes.	/etc/uerf - f /usr/adm/syserr/syserr.remotes - H guitar
- M mainframe	Specifies processor error types.	/etc/uerf - M mem /etc/uerf - M cpu,mem
- n	Displays errors as they occur. Do not use with - f option.	/etc/uerf - n - D
- o output	Selects output format for report. Default is brief format.	/etc/uerf - o brief /etc/uerf - o full /etc/uerf - o terse
- O operating system	Selects operating system errors.	/etc/uerf - O /etc/uerf - O raf,ptf,ast
- R reverse chronological order	Selects records in reverse chronological order.	/etc/uerf - R /etc/uerf - A bua - R /etc/uerf - R - O
- r records	Selects errors for specified record codes.	/etc/uerf - r 300,301 /etc/uerf - r 100-109
- s sequence numbers	Selects errors for specified sequence numbers.	/etc/uerf - s 750-800
- S	Produces summary report.	/etc/uerf -S -o brief
- t time	Selects errors within specified time range.	/etc/uerf - t s:13-apr-1986 /etc/uerf - t s:13-apr,18:30
- T tapes	Selects errors for tmscp tape types.	/etc/uerf - T /etc/uerf - T tu81
- x	Excludes specified options. You cannot exclude - f, - H, - h, - n, - o, - t, or - R options.	/etc/uerf - A - x - D
- Z	Displays entry in hex format.	/etc/uerf - Z



## C

### client

defined, 1-1

### configuration file (error logger)

See *elcsd.conf* file

## E

### elcsd daemon

disabling, 1-2

enabling, 1-2

rc file entry, 1-3

### elcsd.conf file

changing entries, 2-3

contents, 1-2

for client system, 2-9e

for server system, 2-3, 2-8e

parameters, 2-5 to 2-8

reconfiguring, 2-3 to 2-8

specifying client system, 2-7

specifying server system, 2-7

with default entries, 2-4e

### elcsdlog file

contents, 2-10

### eli command

options, 2-1 to 2-3

### error logger

components, 1-1 to 1-2

controlling manually, 2-1 to 2-3

### error logger (cont.)

defined, 1-1

description, 2-1 to 2-8

disabling, 2-1

enabling in multi-user mode, 2-2

enabling in single-user mode, 2-2

error message types, 2-5

error messages, 2-12

errors reported, 3-1

events reported, 3-1

file maintenance, 2-10

local system crash and, 1-3

remote system crash and, 1-4

remote system shutdown and, 1-4

### error report

ASCII messages and, 3-10n

brief format, 3-4e

directing to terminal, 3-1

displaying as errors occur, 3-6

excluding error types, 3-12

for adapter errors, 3-7

for class errors, 3-8

for disk errors, 3-8

for error type, 3-7 to 3-9

for mainframe errors, 3-8

for operating system, 3-9

for record type, 3-10

for sequence number, 3-11

for specific errors, 3-9 to 3-12

for tape errors, 3-9

for time period, 3-11

### **error report (cont.)**

- formatting, 3-3 to 3-5
- from specific file, 3-5
- from specific systems, 3-6
- full format, 3-5e
- generating, 3-1 to 3-6
- generating in reverse order, 3-6
- generating in single-user mode, 3-6
- specifying brief format, 3-4
- specifying full format, 3-4
- specifying terse format, 3-5
- terse format, 3-3

### **error report formatter**

*See* uerf command

### **errorlog buffer**

- clearing, 2-3, 1-2

### **errorlog file**

- backing up, 2-10
- clearing, 2-10
- logging message to, 2-3
- move command and, 2-11n
- remote system directory path, 2-7
- remote systems and, 2-5
- single-user directory path, 2-7
- specifying backup, 2-6
- specifying directory path, 2-6
- specifying size, 2-6

### **event file**

*See* errorlog file

## **S**

### **server**

- defined, 1-1

## **U**

### **uerf command**

- data files, 3-1
- help display, 3-3e

### **uerf command (cont.)**

- help option, 3-2
- option list, 3-14t
- options, 3-13 to 3-14
- record parameters supported, 3-10
- using, 3-1 to 3-14

# HOW TO ORDER ADDITIONAL DOCUMENTATION

## DIRECT TELEPHONE ORDERS

In Continental USA  
and New Hampshire,  
Alaska or Hawaii  
call **800-DIGITAL**

In Canada  
call **800-267-6215**

## DIRECT MAIL ORDERS (U.S. and Puerto Rico\*)

DIGITAL EQUIPMENT CORPORATION  
P.O. Box CS2008  
Nashua, New Hampshire 03061

## DIRECT MAIL ORDERS (Canada)

DIGITAL EQUIPMENT OF CANADA LTD.  
100 Herzberg Road  
Kanata, Ontario K2K 2A6  
Attn: Direct Order Desk

## INTERNATIONAL

DIGITAL EQUIPMENT CORPORATION  
PSG Business Manager  
c/o Digital's local subsidiary  
or approved distributor

Internal orders should be placed through the Software Distribution Center (SDC), Digital Equipment Corporation, Westminister, Massachusetts 01473

\*Any prepaid order from Puerto Rico must be placed  
with the Local Digital Subsidiary:  
809-754-7575



## 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 \_\_\_\_\_

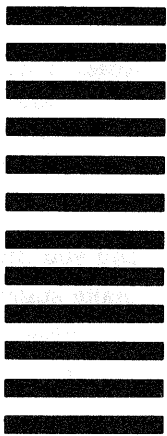
City \_\_\_\_\_ State \_\_\_\_\_ Zip Code  
or  
Country \_\_\_\_\_

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  
ZK03-3/X18  
Spit Brook Road  
Nashua, N.H.  
03063

Do Not Tear - Fold Here and Tape

Cut Along Dotted Line