Digital UNIX

Guide to Prestoserve

Order Number: AA-PQT0D-TE

March 1996

Product Version:	Prestoserve Version 2.1A
Operating System and Version:	Digital UNIX Version 4.0 or higher

This manual describes how to install and use the Prestoserve software.

Digital Equipment Corporation
Maynard, Massachusetts

Contents

3 Prestoserve Administration

4 Recovering from System Failures

Index

Examples

Figures

About This Manual

Prestoserve speeds up synchronous disk writes, including NFS server access, by reducing the amount of disk I/O. Prestoserve stores synchronous writes in nonvolatile memory instead of writing them to disk. The stored data is then written to disk asynchronously as needed or when the machine is halted.

This manual shows how to install, use, and monitor Prestoserve.

Audience

This manual is written for the person who manages and maintains the Digital UNIX® operating system. The manual assumes that this individual is familiar with Digital UNIX commands, the system configuration, and the system hardware. This manual also assumes that the Prestoserve hardware is already installed.

Digital has changed the name of its UNIX operating system from DEC OSF/1 to Digital UNIX. The new name reflects Digital's commitment to UNIX and its conformance to UNIX standards.

Organization

This manual consists of four chapters:

Chapter 1	Provides an overview of disk operations, the Network File System (NFS), and Prestoserve.
Chapter 2	Describes how to install the Prestoserve software subset, register the Prestoserve software license, and configure Prestoserve into your kernel. This chapter also contains information about setting up Prestoserve using the `prestosetup` command and the manual method.
Chapter 3	Describes the Prestoserve states and buffers. This chapter also explains how to manage Prestoserve and how to handle the Prestoserve cache.
Chapter 4	Describes how to recover from a system failure and how to handle disk errors.

Related Documents

You should have the hardware documentation for your system, peripherals, and the Prestoserve hardware. The printed version of the Digital UNIX documentation set is color coded to help specific audiences quickly find the books that meet their needs. (You can order the printed documentation from Digital.) This color coding is reinforced with the use of an icon on the spines of books. The following list describes this convention:

Audience	Icon	Color Code
General users	G	Blue
System and network administrators	S	Red
Programmers	P	Purple
Device driver writers	D	Orange
Reference page users	R	Green

Some books in the documentation set help meet the needs of several audiences. For example, the information in some system books is also used by programmers. Keep this in mind when searching for information on specific topics.

The *Documentation Overview, Glossary, and Master Index* provides information on all of the books in the Digital UNIX documentation set.

Reader's Comments

Digital welcomes any comments and suggestions you have on this and other Digital UNIX manuals.

You can send your comments in the following ways:

- Fax: 603-881-0120 Attn: UEG Publications, ZK03-3/Y32
- Internet electronic mail: `readers_comment@zk3.dec.com`

  A Reader's Comment form is located on line in the following location:

  `/usr/doc/readers_comment.txt`

- Mail:

  Digital Equipment Corporation
  UEG Publications Manager
  ZK03-3/Y32
  110 Spit Brook Road
  Nashua, NH 03062-9987

A Reader's Comment form is located in the back of each printed manual. The form is postage paid if you mail it in the United States.

Please include the following information along with your comments:

- The full title of the book and the order number. (The order number is printed on the title page of this book and on its back cover.)
- The section numbers and page numbers of the information on which you are commenting.
- The version of Digital UNIX that you are using.
- If known, the type of processor that is running the Digital UNIX software.

The Digital UNIX Publications group cannot respond to system problems or technical support inquiries. Please address technical questions to your local system vendor or to the appropriate Digital technical support office. Information provided with the software media explains how to send problem reports to Digital.

Conventions

The following conventions are used in this manual:

`#`	A number sign represents the superuser prompt.
`% cat`	Boldface type in interactive examples indicates typed user input.
file	Italic (slanted) type indicates variable values, placeholders, and function argument names.
[|] **{ | }**	In syntax definitions, brackets indicate items that are optional and braces indicate items that are required. Vertical bars separating items inside brackets or braces indicate that you choose one item from among those listed.
. . .	In syntax definitions, a horizontal ellipsis indicates that the preceding item can be repeated one or more times.
`cat`(1)	A cross-reference to a reference page includes the appropriate section number in parentheses. For example, `cat`(1) indicates that you can find information on the `cat` command in Section 1 of the reference pages.

Understanding Prestoserve 1

The Prestoserve product is a combination of the Prestoserve NVRAM hardware and the Prestoserve software. This manual assumes that the Prestoserve hardware is already installed in your system.

This chapter explains how Prestoserve improves disk I/O performance by caching synchronous disk writes. It also describes the disk operations that can utilize Prestoserve and describes how Prestoserve can alleviate Network File System (NFS) performance problems.

1.1 Prestoserve and Synchronous Write Operations

Prestoserve speeds up any application that requires synchronous writes to ensure data reliability. A file modification is synchronous if it must be immediately written to disk before the application can continue. Synchronous writes ensure data reliability because the writes are not stored in volatile memory and then later written to disk. For example, all UFS and NFS file system modifications due to creating or deleting files are written synchronously. In addition, all NFS data writes are written synchronously. Many database or transaction systems require synchronous writes and can show significant performance improvements with Prestoserve.

Applications that require synchronous writes are sometimes implemented by opening files requiring synchronous update with the O_FSYNC synchronous write flag. This flag can also be set by using the `fcntl` system call. An alternative to making every write synchronous is to commit a series of write operations with the `fsync` system call. This call synchronously writes all modified blocks of a file to disk. See `fcntl`(2) and `fsync`(2) for more information on the system calls.

In addition, the `mount -o sync` command causes all file system writes to be synchronous. Refer to `mount`(8) for more information.

1.2 How Prestoserve Works

Prestoserve uses the Prestoserve buffer cache (NVRAM hardware) to temporarily, but securely, store synchronous disk I/O. Instead of immediately writing the I/O to disk, Prestoserve stores the data in the cache's nonvolatile memory and then writes the data to disk when appropriate. Nonvolatile memory is used to ensure that data is not lost because of a power

failure or a system crash. To the operating system, Prestoserve appears to be a very fast disk.

Prestoserve accelerates synchronous writes to mounted file systems by making synchronous disk writing more efficient. The Prestoserve software allows you to specify which file systems you want to accelerate.

Prestoserve works in a way that is similar to the way the system buffer cache speeds up asynchronous disk I/O requests. The Prestoserve buffer cache is interposed between the operating system and the device drivers for the disks on a server. When a synchronous write request is issued to a file system that has been accelerated with Prestoserve, the write is intercepted by the Prestoserve pseudodevice driver, which stores the data in the cache's nonvolatile memory instead of on the disk. This causes the synchronous write to occur at memory speed, not at disk speed.

As the nonvolatile memory fills up, the cache asynchronously flushes the data to disk in portions that are large enough to allow the disk drivers to optimize the order of the writes. A modified form of Least Recently Used (LRU) replacement is used to determine the order. Reads that hit or match blocks in the Prestoserve cache's nonvolatile memory can also realize performance benefits because the data does not have to be read from disk.

Note

Note that some database applications use raw character device disk partitions to manage their own file system data structures. Prestoserve will neither accelerate nor interfere with raw character device I/O.

There are several reasons why reliable write caching can boost performance:

- A single UFS file system write operation causes two or three writes to disk because each write must update not only the data block but also the file definition blocks (inodes and indirect blocks). Because the same file definition block is updated for each data block in the file, 50 percent to 65 percent of all disk writes can be eliminated by rewriting the definition block cached in the Prestoserve nonvolatile memory buffers. Data blocks can also be found in the Prestoserve cache, although the frequency of these cache hits is significantly less than the frequency of hits on the file definition blocks.
- The data in the Prestoserve nonvolatile cache can be flushed asynchronously to optimize disk I/O performance. This allows blocks of data to be scheduled in order to take advantage of disk arm position. Because disk seek times are significant, this represents a major performance improvement.

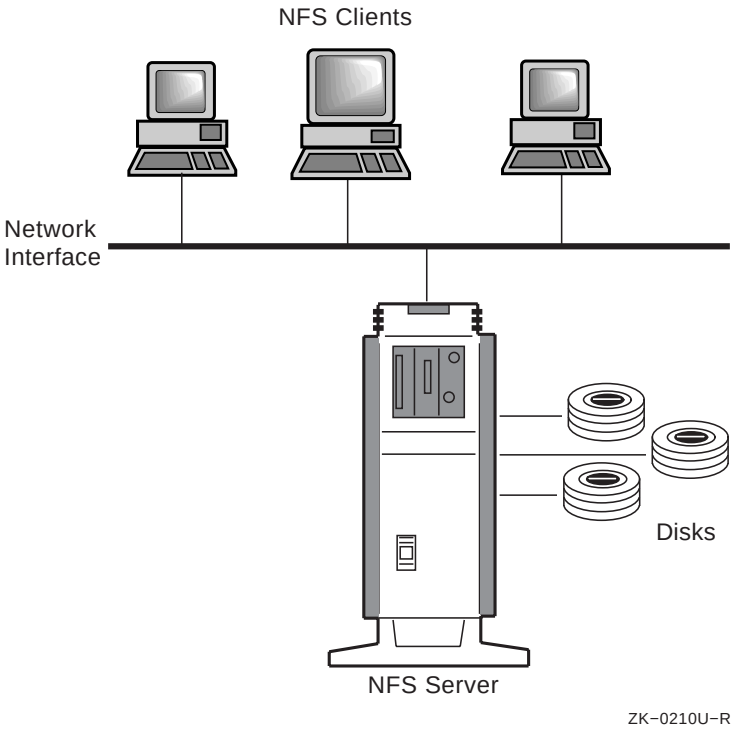
- Because read caching is already effective, operations that modify file data account for a disproportionately large amount of actual disk traffic. However, read operations that are not utilized by the traditional system buffer cache are essentially synchronous (some read-ahead is possible) and must compete with the heavy write traffic. Altogether, operations that modify data typically make up about 20 percent of a normal operation mix, but about 60 percent of the requests for disk I/O are due to these data modification operations.

1.3 NFS Environment and Performance Problems

The Network File System (NFS) allows users to access files transparently across networks. The NFS supports a spectrum of network topologies, from small and simple networks to large and complex networks. To gain the maximum advantage that Prestoserve can provide, it is necessary to understand how different network design defects affect performance.

Figure 1-1 shows a typical NFS environment: one server supporting several clients connected by the Ethernet. The server manages the shared resources, such as data files and applications, and is responsible for the multiplexing of its resources among the various clients. The server also must maintain and protect the data within these shared resources.

Figure 1-1: Example of NFS Environment



NFS performance problems can be broken down into three basic areas: client, network, and server problems. The following sections describe each of these areas and show why the server and, in particular, the server's I/O subsystem are usually the primary causes of poor NFS performance.

1.3.1 Network Problems

The network used to communicate between the client and server does not normally cause a performance problem. There are, however, two conditions to look out for: network delays and high retransmission rates. If the Ethernet is overutilized, clients experience long delays waiting for a free slot to send requests. Ethernet utilization over 50 percent often indicates excessive network delay.

Network topology often contributes to excessive delay. If clients are located across many gateways from the servers that they use often, their requests experience long delays. You may be able to solve the problem by restructuring the network topology to distribute the load more evenly.

Excessive retransmissions can cause poor performance because the client must wait for the server to respond before it retransmits a request. Excessive retransmissions can be caused by the following problems:

- Overloaded servers that drop packets due to insufficient buffering
- Inadequate Ethernet transceivers that cause packets to be dropped under busy conditions
- Physical network errors, such as those caused by a noisy coaxial cable

You can use the `nfsstat -c` command to measure the NFS retransmission rate on client machines. You can then determine the rate of retransmissions. Refer to `nfsstat`(8nfs) for more information.

The average NFS response time to a client request under a low to medium load is approximately 30 milliseconds. Most clients retransmit a request after approximately 1 second. If a 10 percent reduction in performance is acceptable, then a 3 millisecond increase in response time is an acceptable limit. This reduction gives an acceptable NFS retransmission rate of 0.3 percent. The calculation is as follows:

```
  .003 sec/request
----------------------  =  0.003 retransmission/request
1.0 sec/retransmission
```

Because the worst case NFS request (read or write 8 kilobytes over the Ethernet) requires seven packets (one request and six fragmented replies), the error rate of the network must be less than 0.04 percent. The calculation is as follows:

```
  0.3 percent
---------  = 0.04 percent
    7
```

The calculation shows the overall acceptable error rate for both the client and the server, so the acceptable error rate measured at either machine is half of this rate (0.02 percent).

You can use the `netstat -i` command to measure the network error rate. If this rate is unacceptably high, determine if an individual machine is generating an excessive number of errors. If the problem appears to be pervasive, analyze the cabling technology that is being used. For example, if you have difficulties with noisy nonstandard coaxial cable, you could switch to a twisted-pair Ethernet. Refer to `netstat`(1) for more information.

1.3.2 Client Problems

Adding disks or memory to a client can improve performance in two ways: by improving access time and by reducing the overall load on the server and network. A client can avoid NFS performance problems for files that are not

shared (such as root, swap, and temporary files) by using local disks for these files. For diskless clients, increased memory can make a big improvement in performance by allowing the client to swap and page less often. By adding local resources, the demands on the server and the network can be reduced.

While it is easy to improve client performance by adding memory or disks, these improvements may not be cost effective because of the additional administrative tasks that are needed to maintain the operating system. For example, if you store valuable data on local disks, you must ensure that the disks are backed up. If the data is shared, you may also have to ensure that other systems have access. If you add resources to the server, the additional administrative costs are less than if you add the resources to the client.

1.3.3 Server Problems

On most NFS servers, the limiting factor is the speed of the disk. Most high-speed disks can sustain from 30 to 40 disk operations per second. Most of the time spent waiting for a disk operation occurs during head seeks or rotational delay. If you use a faster disk or disk controller and if you spread the load over multiple disks, you can obtain a small improvement in I/O performance. However, the best way to improve I/O performance is to reduce the number of disk operations.

To alleviate performance problems, you should concentrate your resources on the server. If you have already added memory to your server to increase the size of the buffer cache and the server is still too slow, you could obtain another server and split the load between the two servers. However, not only does this solution have a large direct cost, but there is a significant administrative cost associated with supporting an additional server. Prestoserve is an alternative solution that can increase the performance of the NFS server without an additional server and its added administrative cost.

1.3.4 NFS Server Performance

Digital UNIX uses a buffer cache in memory to avoid disk operations whenever possible. This memory is effective in reducing the client waiting time for relatively slow disk I/O. It also makes disk I/O more efficient by allowing the staging and scheduling of disk operations.

You can improve performance by allowing the disk device driver to schedule several requests at a time to take advantage of the position of the disk arm. The total amount of disk I/O is reduced, because repeat requests may be found in the cache. If NFS read activity is high, then adding more memory to your server can improve server performance because the size of the buffer cache is a percentage of the size of memory.

Performance problems at the server make the system buffer cache inefficient when serving remote write requests. NFS uses a simple stateless protocol,

which requires that each client request be complete and self-contained and that the server completely process each request before sending an acknowledgment back to the client. If the server crashes or if an acknowledgment is lost, the client retransmits its request to the server. Because of this, the following events occur:

- The server cannot acknowledge the client's request until data is safely written to stable storage.
- The client knows exactly how much modified data has been safely stored by the server.
- The server cannot cache modified data in volatile storage because the data may be lost if the server crashes.

You cannot use the system buffer cache to improve performance with NFS requests that modify data. If a server writes modified data only to volatile memory, a server crash would jeopardize the data integrity. The client may assume that its data is safely stored, but if a crash occurs and the data was stored only in volatile memory, the data may be lost. Because a single server stores data for many clients, many clients can be affected. However, if modifications are always synchronously written to disk, data will not be lost, and you can recover from server crashes.

Client operations that modify data, such as file creation, file removal, and attribute modification must be written synchronously to disk before the server responds to the client. For example, when the client creates a new file, the server may have to update the data and file definition blocks for the directory that contains the file. To ensure file system integrity in the local case, these operations are also written synchronously to disk.

Because NFS operations are synchronously committed to disk, a server can survive system failures because data integrity is ensured. However, performance is degraded because these operations take place at disk speeds and not at the memory speeds available to cachable operations. In addition, because these operations are processed serially, there is no opportunity to optimize the scheduling of the disk arm. Modifications to the cache are written synchronously to disk, so there is no opportunity to decrease write-disk traffic.

Unless your server is only supplying read-only access to files, some NFS operations must be synchronously committed to disk. Because disks are much slower than memory, this is a large burden. Prestoserve stores synchronous writes to nonvolatile memory; therefore data is secure without a corresponding decrease in performance.

1.3.5 Prestoserve's Impact on NFS Server Performance

Prestoserve's performance impact on any particular server can vary widely as a result of the demands placed on the NFS server by its client systems. Heavily loaded NFS servers (those performing more than 10 percent of NFS writes, creates, and deletes) will benefit the most from Prestoserve. Conversely, lightly loaded NFS servers (those performing less than 4 percent of NFS writes) may have no noticeable benefits from Prestoserve.

In addition to increased response time, Prestoserve uses the server's disk more efficiently. For example, in many cases, Prestoserve allows you to double the number of diskless clients that a single NFS server can support if it has the necessary disk capacity and a sufficient amount of main memory. Prestoserve's improvement to an NFS server is most noticeable when the server is busy.

Getting Started with Prestoserve 2

This chapter describes how to start using Prestoserve. The following sections describe how to do the following:

- Install the `dxpresto` software subset

  Note that the Prestoserve base utilities and kernel components are installed when you install the operating system.
- Register the Prestoserve software license
- Configure Prestoserve support and the Prestoserve controller device into your kernel
- Set up and enable Prestoserve

2.1 Installing the dxpresto Subset

You must install the subset containing the `dxpresto` software if you want to use the `dxpresto` command. The `dxpresto` command graphically displays information about the Prestoserve state and performance statistics. You can install the `dxpresto` software subset when you install Digital UNIX or by using the `setld` command.

To install the `dxpresto` software subset when installing Digital UNIX, you must perform an advanced installation. During the installation, you are prompted to select the optional software subsets that you want to install. Type the number associated with the following subset description:

```
Additional DECwindows Applications
```

Refer to the *Installation Guide* for more information about the advanced installation.

If you are already running Digital UNIX, you can install the Prestoserve software subset by using the `setld` command.

To display the status of all the subsets known to the system, use the

following command:

```
# setld -i
```

The operating system displays a table that lists the name, status, and description of each software subset. The name of the subset is a string of seven or more characters used to uniquely identify the subset. The following is a description of the subset that contains the `dxpresto` software that you must install:

```
Additional DECwindows Applications
```

Note the name of the subset because you must specify that name to install the subset.

Load the subset by using the following command syntax:

setld –l *location subset_name*

The *location* variable specifies the location of the subset. The *subset_name* variable specifies the name that you obtained from the `setld -i` command. Refer to `setld`(8) for more information about loading software subsets.

2.2 Registering the Prestoserve License

After you install the Prestoserve software subset, you must register the software license by using the License Management Facility (LMF). If you try to use Prestoserve without registering the license, the following message is displayed on your terminal:

```
Prestoserve license not registered
```

To register the Prestoserve license, you must have your Product Authorization Key (PAK), which contains information about the license. A PAK is sent as part of your product kit. In order to comply with Digital's license terms, always register a PAK in the License Database using the `lmfsetup` script or the `lmf` command.

Note

If you do not have a PAK, contact your Digital Customer Services representative.

To make registering the Prestoserve license easy, you are provided with the `PRESTOSERVE-OA` PAK template file, which includes some of the license information. The file is located in the `/usr/var/adm/lmf` directory.

An example of the /usr/var/adm/lmf/PRESTOSERVE-OA Prestoserve PAK template file is as follows:

```
PAK ID:
                    Issuer: DEC
     Authorization Number:

PRODUCT ID:
              Product Name: PRESTOSERVE-OA
                  Producer: DEC

NUMBER OF UNITS:
           Number of units:

KEY LEVEL:
                   Version:
     Product Release Date:

KEY TERMINATION DATE:
     Key Termination Date:

RATING:
  Availability Table Code:
       Activity Table Code:

MISCELLANEOUS:
               Key Options:
             Product Token:
               Hardware-Id:
                  Checksum:
```

The lmfsetup script allows you to register data supplied by a PAK. The lmfsetup script prompts you for the data associated with each field on a PAK.

To use the lmfsetup script to register the Prestoserve license, enter the following command:

```
# lmfsetup /usr/var/adm/lmf/PRESTOSERVE-OA
```

Once you enter all the data, the LMF makes sure you have supplied entries for all mandatory fields and that the value in the Checksum field validates the license data. If the data is correct, LMF registers the PAK in the License Database. If any data is incorrect, LMF displays the appropriate error message and gives you an opportunity to reenter the data. For more information, refer to lmfsetup(8).

You can also register the Prestoserve license by entering the following lmf

`register` command:

```
# lmf register /usr/var/adm/lmf/PRESTOSERVE-OA
```

If you use the `lmf register` command, the template file is displayed, and an editor is invoked so that you can edit the fields and include your PAK information. The EDITOR environment variable defines the editor that is used. If the EDITOR variable is not defined, the `vi` editor is used.

After you exit from the editor, LMF scans the template file to ensure that all the license data is correct. If information is incorrect or missing, a descriptive error message is displayed, and you are given the opportunity to reenter the editor and correct any mistakes.

If the license data is correct, it is copied into the License Database. You must then use the `lmf reset` command to copy the license information from the License Database to the kernel cache. For example:

```
# lmf reset
```

For more information, refer to `lmf`(8).

2.3 Configuring Prestoserve

You must make sure that the Prestoserve software is configured into your kernel before you use Prestoserve to accelerate file systems. There are various Prestoserve hardware configurations that require different forms of kernel configuration.

If the Prestoserve hardware was installed in your system when the operating system was installed, the Prestoserve software was automatically configured into your kernel. If not, you may have to reconfigure your kernel to include Prestoserve support and the correct Prestoserve controller device.

2.3.1 Adding the presto Pseudodevice

To run Prestoserve, you must have the Prestoserve pseudodevice definition in your system configuration file, `/usr/sys/conf/`*`NAME`*. The *NAME* variable usually specifies the system host name. The Prestoserve pseudodevice definition is as follows:

```
pseudo-device         presto
```

If this definition is not included in your system configuration file, you must add it and then reconfigure your kernel.

Perform the following steps to add the Prestoserve support:

1. Edit the current configuration file and include the Prestoserve definition.
2. Shut down the system to single-user mode.

3. Mount the local file systems by using the `mount` command with the `-a` and `-t ufs` options.
4. Run the `doconfig` program with the `-c` *`config_file`* option, specifying the name of the current configuration file.

   The `doconfig` program displays the following message as it begins to reconfigure your kernel:

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

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

   ```
   The new kernel is /sys/NAME/vmunix
   ```

5. Make a copy of the original kernel and then move the new kernel to the `root` directory. Use the following commands, replacing the *`NAME`* variable with the system host name in uppercase letters:

   ```
   # cp /vmunix /vmunix_old
   # mv /sys/NAME/vmunix /vmunix
   ```

   Prestoserve is activated when you reboot the system using the new kernel. If you cannot boot the new kernel, use the original kernel that you saved. Once you successfully boot with the new kernel, you can delete the original kernel that you saved.
6. Notify users that the system is going down and reboot the system using the `shutdown -r` command.

2.3.2 Adding the Prestoserve Controller Device

Some systems require that a Prestoserve controller device be configured into your kernel. If your system requires a Prestoserve controller device, the name may be specified in either the system-specific sections in the release notes or in the *System Administration* manual.

If your system requires a Prestoserve controller device, you must include it in your `/usr/sys/conf/`*`NAME`* system configuration file, where *`NAME`* specifies your system host name. You probably will not have to add the controller device if the Prestoserve hardware was already attached when you installed your system. If you added Prestoserve hardware support after you installed your system, you must add the device to the configuration file and reconfigure your kernel as specified in Section 2.3.1.

The following is an example of the Prestoserve controller device for the DEC

3000 Model 500:

```
controller          nvtc0    at *   slot ? vector nvtcintr
```

The following is an example of the Prestoserve controller device for the DEC 2000 Model 300 and the DEC 2000 Model 500:

```
controller          envram0   at eisa?
```

You can also build a new configuration file that will contain an entry for the Prestoserve controller device if one is needed. You can do this by saving the running kernel (`/vmunix`), installing the `/genvmunix` generic kernel, and then using the `doconfig` program. You should specify a configuration file name that is different from your current one, because any customizations that you made to your current configuration file will not be included in the new file. You can then use the `diff` command to determine any differences between the configuration files and determine the controller device name.

Refer to the *System Administration* manual and `doconfig`(8) for more information on reconfiguring the kernel.

2.4 Setting Up and Enabling Prestoserve

To use the Prestoserve software, you must perform some setup tasks. At a minimum, your system must meet the following requirements:

- The Prestoserve control device, `/dev/pr0`, must exist.
- The `portmap` daemon must be running.
- If you want to allow remote systems to administer a Prestoserve cache and its driver, the `prestoctl_svc` daemon must be running.

You can use the `prestosetup` command to set up and enable Prestoserve, or you can manually invoke commands. The `prestosetup` command invokes an interactive facility that performs all the tasks necessary to use Prestoserve. The two methods are described in the following sections.

2.4.1 Using the prestosetup Command

The `prestosetup` command invokes an interactive facility that prompts you for information about how you want to set up Prestoserve and performs all the setup tasks. The facility does the following:

- Verifies that the license is registered
- Verifies that the Prestoserve utilities are installed
- Verifies that the software is configured into your kernel
- Verifies that the `portmap` daemon is running

- Creates the `/dev/pr0` Prestoserve control device if necessary

In addition to performing the tasks necessary to set up and use Prestoserve, the `prestosetup` command can also do the following:

- Create the `/etc/prestotab` file and prompt you for the file systems to automatically accelerate when the system starts up. To specify a file system, use the mount point. Do not specify a block device because some functional subsystems, such as the Advanced File System (advfs), can map more than one block device to a mount point. If you do not specify any file systems, then all the currently mounted file systems are automatically accelerated when the system starts up.
- Set the appropriate run-time variables in the `/etc/rc.config` file to automatically accelerate file systems and start the `prestoctl_svc` daemon at system startup.
- Immediately accelerate file systems and start the `prestoctl_svc` daemon without rebooting the system.

After you enter the necessary information, the `prestosetup` command displays the information that you entered and prompts you to confirm that it is correct. If you enter no, the `prestosetup` command exits and no changes are made. If you enter yes, the `prestosetup` command sets up Prestoserve according to your specifications.

After you have set up Prestoserve, you can start to use it. If you chose to immediately accelerate file systems and start the `prestoctl_svc` daemon without rebooting the system, Prestoserve is ready to be used.

If you chose the option of automatically accelerating the file systems and starting the `prestoctl_svc` daemon when the system starts up, you can reboot your system to start using Prestoserve.

If you did not set up Prestoserve to automatically accelerate file systems, you can invoke the `presto` command with the `-u` or `-U` option and specify the file systems to accelerate. You can also manually start the `prestoctl_svc` daemon if necessary. Refer to Section 2.4.2.5 and Chapter 3 for more information.

Note that after you set up Prestoserve, you can use the `prestosetup` command to add to the list of file systems in the `/etc/prestotab` file. To remove file systems from the file, you must manually edit the file.

The following example shows how to use the `prestosetup` command:

```
# /usr/sbin/prestosetup

Checking LMF licensing...
Checking kernel configuration...

Note: If the Prestoserve hardware was not present in your system
at installation time it may be necessary to add device specific
information to your system configuration file and to reconfigure
your kernel.  For more information, refer to the Guide to
Prestoserve.

Verifying that the Prestoserve control device is present...

You will be asked a series of questions about which Prestoserve
utilities to run.  Default answers are shown in square
brackets ([]).  To use a default answer, press the RETURN key.

Do you wish to have the Prestoserve enabled automatically at
system startup time?  This involves executing the presto command
with the -u option.

Automatically enable Prestoserve [y]? y

You have selected to automatically enable Prestoserve.  Now
enter the names of the filesystems you want to accelerate.  These
names will be entered into the /etc/prestotab file.  If no names
are specified then all writable filesystems will be accelerated.
Consider the implications of this question carefully.

When finished entering filesystems, press only the RETURN key.

Enter the filesystem: /usr

Enter the filesystem:  Return

Prestoserve acceleration list complete...

Do you wish to have the prestoctl_svc daemon enabled automatically
at system startup time?  This involves executing the prestoctl_svc
command.  The prestoctl_svc daemon must be running if you intend to
use the dxpresto graphical interface or if you are allowing remote
administration of the Prestoserve functions.

Automatically enable prestoctl_svc [y]? y

You have selected to run the prestoctl_svc daemon.  Do you wish to
allow any network client to be able to change your Prestoserve
state?  Consider the security implications of this question
carefully.  This involves executing the prestoctl_svc daemon with
the -n option.
Allow remote Prestoserve management [n]? y

Verifying that the portmap daemon is running...

Please confirm the following information which you
have entered for your Prestoserve setup:

        Automatically start up Prestoserve
        Accelerate the following filesystems:
```

```
/usr

        Automatically start up prestoctl_svc
        Any network host can change presto state

Enter "c" to CONFIRM the information, "q" to QUIT prestosetup
without making any changes, or ‘‘r’’ to RESTART the procedure: c

Updating files:
        /etc/rc.config
        /etc/prestotab

The necessary Presto daemon entry and Presto enable command have
been placed in the file /sbin/init.d/presto.  In order to begin
using Presto, you must now start the daemon and enable Presto.
You may either allow prestosetup to perform these tasks
automatically or you may invoke them by hand, but in either case
they will be started automatically on subsequent reboots.

If you choose to have prestosetup stop and start Presto
acceleration now (without a reboot), all Presto acceleration will
be stopped, then those functions you chose to be run in the
preceding questions will be started.  You probably do not want to
automatically startup Prestoserve acceleration unless all the
filesystems targeted for acceleration are already created
and mounted.

Would you like to stop/start Presto acceleration now [n]? y

state = DOWN, size = 0x1ffc00 bytes
statistics interval: 00:00:00  (0 seconds)
write cache efficiency: 0%
All batteries are ok
Prestoserve acceleration has been disabled.
Starting Prestoserve: presto -u for the following:
/usr - Presto enabled
Presto has been enabled.
Starting prestoctl_svc
Presto daemon started.

The Presto daemon for your machine has been started and
Prestoserve acceleration has been enabled.

***** PRESTOSETUP COMPLETE *****
#
```

2.4.2 Manually Setting Up Prestoserve

If you do not use the `prestosetup` command to automatically set up Prestoserve on your system, you can manually set up Prestoserve by entering commands and editing files. To manually set up Prestoserve on your system, you must perform the following steps:

1. Create the `/dev/pr0` generic Prestoserve control device if necessary. Refer to Section 2.4.2.1 for information.

2. Start the portmap daemon. Refer to Section 2.4.2.2 for information.
3. Optionally, set the run-time configuration variables in the /etc/rc.config file to automatically accelerate file systems and start the prestoctl_svc daemon when the system starts up. Refer to Section 2.4.2.3 for information.
4. Optionally, create an /etc/prestotab file and include the mount points for the file systems that you want automatically accelerated when the system starts up. Refer to Section 2.4.2.4 for information.
5. Optionally, start the prestoctl_svc daemon if you want to allow remote systems to administer a Prestoserve cache and its driver. Refer to Section 2.4.2.5 for information.

After you perform the previous tasks to set up Prestoserve, you can start using it. If you set the Prestoserve run-time configuration variables to automatically accelerate file systems when the system starts up, you can reboot the system to start using Prestoserve.

If you did not set the run-time variables, you can use the presto command with the -u or -U option to accelerate file systems. Refer to Chapter 3 for more information.

The following sections describe in detail how to manually set up Prestoserve.

2.4.2.1 Creating the Prestoserve Control Device

The /dev/pr0 generic Prestoserve control device must exist in order for you to use Prestoserve. If the device exists, you do not have to create the device. If the device does not exist, then you must create the device by using the MAKEDEV command. Refer to MAKEDEV(8) for more information.

To create the /dev/pr0 control device, use the following commands:

```
# cd /dev
# MAKEDEV pr0
```

2.4.2.2 Starting the portmap Daemon

You must ensure that the portmap daemon is running to use the prestoctl_svc daemon. If the portmap daemon is not running, you can start the daemon manually.

The syntax for the portmap daemon is as follows:

/usr/sbin/portmap

The portmap daemon can also be started by the /sbin/init.d/nfs script.

2.4.2.3 Specifying Configuration Variables in the rc.config File

To automatically accelerate file systems or start the `prestoctl_svc` daemon when the system starts up, use the `rcmgr` command to set Prestoserve run-time configuration variables stored in the `/etc/rc.config` file. These variables are used to configure the Prestoserve subsystem with the `/sbin/init.d/presto` script.

You can set the following Prestoserve run-time variables:

- `PRESTO_CONFIGURED`

  Set this variable to 1 to indicate that Prestoserve is configured and set up on your system. If this variable is set, you can use the `prestosetup` command to add to the list of file system mount points in the `/etc/prestotab` file that are automatically accelerated when the system starts up. Refer to Section 2.4.2.4 for information on creating the `/etc/prestotab` file.

- `PRESTO_ENABLE`

  Set this variable to 1 to automatically accelerate the file systems whose mount points are specified in the `/etc/prestotab` file when the system starts up. If this variable is set and the file is empty or does not exist, then all the currently mounted file systems are accelerated. A 0 (zero) value specifies that no file systems are automatically accelerated. Refer to Section 2.4.2.4 for information on creating the `/etc/prestotab` file.

- `PRESTO_SVC_ENABLE`

  Set this variable to 1 to automatically start the `prestoctl_svc` daemon when the system starts up. This daemon allows remote client systems to monitor a Prestoserve cache and its driver. A 0 (zero) value specifies that the daemon should not be started when the system starts up.

- `PRESTO_SVC_ANY`

  Set this variable to 1 to automatically start the `prestoctl_svc` daemon with the `-n` option when the system starts up. This option allows remote client systems to both monitor and administer a Prestoserve cache and its driver. A 0 (zero) value specifies that the daemon should not be started with the `-n` option when the system starts up.

For example, to display the current setting in the `/etc/rc.config` file for

the PRESTO_ENABLE variable, use the following command:

```
# /usr/sbin/rcmgr get PRESTO_ENABLE
```

To set the PRESTO_ENABLE variable to 1, use the following command:

```
# /usr/sbin/rcmgr set PRESTO_ENABLE 1
```

Refer to rcmgr(8) for more information.

2.4.2.4 Creating the prestotab File

The /etc/prestotab file includes the mount points for the file systems that you want to automatically accelerate when the system starts up. The /etc/prestotab file is created by the prestosetup command, which prompts you for the file systems to automatically accelerate when the system starts up. You can also manually create the file.

Note

If you want to automatically accelerate file systems, you must use the rcmgr command to set the PRESTO_ENABLE variable in the /etc/rc.config file. Refer to Section 2.4.2.3 for more information.

The /etc/prestotab file contains a list of directory mount points (for example, /usr/users). Do not specify a block device because some functional subsystems, such as the Advanced File System (advfs), can map more than one block device to a mount point. Entries in the /etc/prestotab file must be separated by spaces or must be located on separate lines. You cannot specify comments in the file.

If the /etc/prestotab file is empty or does not exist, and the appropriate run-time variables are set, then all the local writable file systems that are currently mounted are accelerated when the system starts up.

An example of an /etc/prestotab file is as follows:

```
/usr/users/disk1
/usr/users/disk2
/var/spool
```

Refer to prestotab(4) for more information.

2.4.2.5 Running the prestoctl_svc Daemon

The prestoctl_svc daemon is an RPC-based daemon that allows interrogation (and, in some cases, administration) of a Prestoserve cache and its driver. The prestoctl_svc daemon must be running on a host if you want to specify that host's name in the presto -h command line or if you want to use the dxpresto application to monitor that host. See Chapter 3

for information about the dxpresto command.

Note

You must ensure that the portmap daemon is running to use the prestoctl_svc daemon. If the portmap daemon is not running, you can start the daemon manually. The portmap daemon can also be started by the /sbin/init.d/nfs script.

The command that starts the prestoctl_svc daemon has the following syntax:

/usr/sbin/prestoctl_svc [**–n**]

If you specify the –n option, any network client can change your Prestoserve state (either UP or DOWN) or change the size of your Prestoserve cache by using the presto command option –h with the –d, –u, and –s administrative options. You must also specify the –n option if you want to use the dxpresto command to change your Prestoserve state. Because of security problems, it is recommended that the –n option not be specified on production machines.

You can also automatically start the prestoctl_svc daemon when the system starts up by setting the PRESTO_SVC_ENABLE and PRESTO_SVC_ANY run-time variables in the /etc/rc.config file. Refer to Section 2.4.2.3 for more information.

Prestoserve Administration 3

This chapter explains how to administer the Prestoserve software. It explains how to select file systems to accelerate and how to use the `presto` and `dxpresto` commands to perform the administrative procedures for the day-to-day operation of Prestoserve. It also describes how to check to determine if Prestoserve is working properly.

3.1 Prestoserve Operation

The following sections explain how Prestoserve operates. It describes the Prestoserve buffers and states.

3.1.1 Prestoserve Buffer Management

Prestoserve is implemented as a pseudodevice driver and uses nonvolatile memory to cache synchronous write requests. Write requests are written synchronously to the Prestoserve cache buffers; as the cache fills, old data is written asynchronously to the appropriate disks.

Prestoserve is interposed between other disk drivers and the rest of the Digital UNIX kernel. Stubs replace the original driver's entry points in the device switch tables. Whenever Prestoserve needs to perform actual I/O (for example, when the data in the cache needs to be written to disk), it uses the real device driver routines.

Buffers in the Prestoserve cache undergo several phases or states. The buffer transition diagram is roughly as follows:

inval –> dirty –> active –> clean –> dirty

The following list describes the buffer states:

`inval` An invalid buffer does not presently contain a disk block image.

`dirty` A dirty buffer contains a valid disk block image that has not yet been written to disk.

`active` An active buffer is currently in transition to the disk, which means that a write operation has started, but it has not been completed on that buffer.

`clean` A clean buffer contains a valid disk block image that has been written to disk.

3.1.2 Prestoserve States

The Prestoserve buffer cache is similar to a disk because it contains data. At appropriate times, the data is written to the actual disks. The Prestoserve driver tries to ensure that data is not lost. When a failure occurs, the driver does not discard cache data unless explicitly requested to do so by the system administrator.

Prestoserve is always in one of three states: UP (enabled), DOWN (disabled), or ERROR (error). When the Prestoserve state is UP, Prestoserve improves I/O performance to accelerated file systems by caching synchronous disk write operations to nonvolatile memory. When the Prestoserve state is DOWN, all I/O requests are passed to the actual devices.

Whenever Prestoserve makes a state transition from UP to DOWN, all Prestoserve buffers are successfully flushed (that is, the data is written to disk) and invalidated. If there are dirty buffers in the Prestoserve cache when the system is rebooted, they are flushed, and Prestoserve enters the DOWN state unless an error occurred during the flushing. Some possible disk errors are: the disk drive is write protected or off line, a cable problem exists, or a bad disk block exists.

Note

Because the Prestoserve state is DOWN after a reboot, you may want to set up Prestoserve so that file systems are automatically enabled when the system starts up. Refer to Chapter 2 for information about automatically acclerating file systems.

If an error occurs, Prestoserve enters the ERROR state. When in the ERROR state, the Prestoserve cache is effectively read-only until the error condition is cleared; then, Prestoserve enters the DOWN state. After you fix the disk error, use the `presto -u` or the `presto -U` command to verify that the error is corrected. If there are no disk errors, the remaining cached data is written to disk and Prestoserve is reenabled. Refer to Section 3.2.1 for more information about the `presto -u` and `presto -U` commands.

The commands that use the `reboot` system call cause Prestoserve to enter the DOWN state if all dirty buffers can be successfully flushed. If the buffers cannot be successfully flushed, Prestoserve enters the ERROR state. Commands that are used to reboot the system include the `halt`, `shutdown`, and `reboot` commands. Refer to Chapter 4 for more information on recovering from the ERROR state.

3.2 Managing Prestoserve

The following sections describe how to manage the Prestoserve software. They describe how to select file systems to accelerate, perform remote Prestoserve administration, display status, and manage the Prestoserve buffer cache.

The `presto` command is used to administer Prestoserve. The `dxpresto` command is used to perform some administrative tasks and also to monitor Prestoserve.

The `presto` command can perform the following administrative tasks:

- Enable and disable file system acceleration
- Administer Prestoserve on remote systems
- Display information about the accelerated file systems
- Display information about Prestoserve state and buffer status
- Reset Prestoserve
- Write the contents of the Prestoserve cache to disk
- Change the size of the Prestoserve cache
- Display Prestoserve troubleshooting information

Refer to `presto`(8), `dxpresto`(8X), and the following sections for more information.

3.2.1 Accelerating File Systems

Prestoserve can accelerate all mounted file systems on a server, regardless of how many disks or controllers are involved. You should accelerate file systems that receive many synchronous write requests. Read-only file systems do not generate synchronous write requests; therefore they are usually not accelerated.

The following list describes some of the types of file systems that may derive benefits from Prestoserve:

- File systems that are accessed through the NFS (because many requests for such files are synchronous)
- File systems that are used heavily for synchronous I/O
- Local file systems (because some operations, such as creating or removing a file, generate synchronous writes)
- Remote swapping done to NFS files may benefit from Prestoserve

Prestoserve maintains full block and raw disk semantics. The performance benefits of Prestoserve are not available to raw character device disk

partitions. Raw character device reads and writes will flush blocks that are in the Prestoserve cache to disk.

You can use the `presto` command with the `-u` or `-U` option to set the Prestoserve state to `UP` and enable acceleration on the specified file systems. The `-U` option sets the Prestoserve state to `UP` only if the specified directory is the root of a mounted file system. Otherwise, the following message is displayed:

```
presto: directory is not a file system root
```

Note that you can set up Prestoserve to automatically accelerate mounted file systems when the system starts up by specifying the appropriate run-time variables in the `/etc/rc.config` file and including the file systems in the `/etc/prestotab` file. Otherwise, you will have to manually accelerate the file systems each time you reboot. Refer to Chapter 2 for more information.

The `presto` command with the `-u` or `-U` option has the following syntax:

presto –u | –U [*filesystem ...*]

Only those file systems specified by the *`filesystem`* variable will have Prestoserve enabled. You specify *`filesystem`* as a directory mount point (for example, `/usr`). Do not specify a block device because some functional subsystems, such as the Advanced File System (advfs), can map more than one block device to a mount point. If *`filesystem`* is not specified, all local writable file systems that are mounted will have Prestoserve enabled. File systems that are presently accelerated will remain accelerated.

If the Prestoserve state was `DOWN`, the `-u` and `-U` options also reset the Prestoserve statistics and buffers to their initial values. If Prestoserve was in the `ERROR` state, Prestoserve attempts to write to disk any blocks that are still in its cache to make sure that the error has been corrected.

If you mount a local file system using the `mount` command after the system is running in multiuser mode, you must use the `presto -u` or `presto -U` command and specify the mount point to accelerate the file system.

Note

When you use the `presto` command option `-h` with the `-u` or `-U` option, Prestoserve is enabled only on those remote file systems that were previously accelerated and have not been disabled by the remote host's administrator.

The following examples enable Prestoserve on all mounted read/write local file systems, on all previously accelerated file systems on a remote host, on a specific mounted file system, and on a directory mount point that is the root of a mounted file system, respectively:

```
# presto -u
# presto -h mmate3 -u
# presto -u /rz1g
# presto -U /usr
```

3.2.2 Disabling File System Acceleration

You can use the `presto` command with the `-d` or `-D` option to stop Prestoserve acceleration and write any Prestoserve cache data to disk.

The `-D` option is similar to the `-d` option, but it sets the Prestoserve state to `DOWN` only if the specified directory is the root of a mounted file system. Otherwise, the following message is displayed:

```
presto: directory is not a file system root
```

The `presto` command with the `-d` or `-D` option has the following syntax:

presto -d | -D [*filesystem ...*]

Only those file systems specified by the *`filesystem`* variable are disabled. You specify *`filesystem`* as a directory mount point (for example, `/usr`). If *`filesystem`* is not specified, all accelerated file systems are disabled, and the Prestoserve state is set to `DOWN`.

Note that the `-d` and `-D` options do not reset Prestoserve statistics, and they take effect before the `-u`, `-U`, or `-R` option.

The following command disables the mounted file system `/usr`:

```
# presto -d /usr
```

3.2.3 Administering Prestoserve from a Remote System

You can use the `presto` command with the `-h` option to administer Prestoserve on a remote machine by using a Remote Procedure Call (RPC) protocol. You can combine the `-h` option with all the `presto` command options except `-R` and `-L`.

The `presto -h` command has the following syntax:

presto -h *hostname*

The *`hostname`* variable specifies the name of the remote host.

The remote machine must be running the `prestoctl_svc` Prestoserve control daemon to allow remote operations on that host. In addition, the remote host must be running `prestoctl_svc` with the `-n` option to allow the use of the `-u`, `-U`, `-d`, `-D`, and `-s` administrative options on the remote host. Refer to Chapter 2 and to `prestoctl_svc`(8) for more information.

You can automatically start the `prestoctl_svc` daemon when the system starts up by setting the `PRESTO_SVC_ENABLE` and `PRESTO_SVC_ANY` run-time variables in the `/etc/rc.config` file on the remote host. This enables the remote host to use the `presto -h` command each time it starts up. Refer to Chapter 2 for more information.

3.2.4 Displaying the Status of File Systems

You can use the `presto` command with the `-l` and `-L` options to display information about the accelerated file systems.

The `-l` option lists the accelerated file systems and their mount points in a format that is similar to the `mount` command. The `-l` option also displays NFS file systems if the server is running the `prestoctl_svc` daemon and if the NFS file systems have been accelerated.

For example:

```
# presto -l
/dev/rz0a on /
/dev/rz1g on /usr/staff
/dev/rz2a on /rz2a
/dev/rz2g on /rz2g
mmate3:/usr/staff on /mmate3
```

The `-L` option displays NFS file systems if the server is running the `prestoctl_svc` daemon. In addition, the `-L` option displays any unusual Prestoserve state for the file systems. The unusual states include the following:

`bounceio`	Instead of directly accessing the Prestoserve cache, the disk device receives the data only after it is first copied to main memory.
`disabled`	The file system is not accelerated.
`error`	An error occurred using the file system, and the data has still not been written successfully.

For example:

```
# presto -L
/dev/rz0a on /
/dev/rz0g on /usr (disabled)
/dev/rz1a on /rz1a
/dev/rz1g on /usr/staff
/dev/rz2a on /rz2a
/dev/rz2g on /rz2g
mmate3:/usr/staff on /mmate3
sunk:/home on /sunk (bounceio)
```

3.2.5 Displaying the Prestoserve State and Buffer Status

If invoked with no options, the `presto` command displays the Prestoserve state (either `UP`, `DOWN`, or `ERROR`), the number of bytes of nonvolatile memory the Prestoserve cache is using, the length of time the cache has been enabled, the write cache efficiency, and the current condition of the batteries.

The following is an example of the `presto` command with no options specified:

```
# presto
state = DOWN, size = 0x7e000 bytes
statistics interval: 00:00:00  (0 seconds)
write cache efficiency: 0%
All batteries are ok
```

You can use the `presto` command with the `-p` option to display additional information about the current Prestoserve state and the statistics for write, read, and total operations. The information displayed by the `-p` option is similar to the information displayed by the `dxpresto` command.

Example 3-1 shows an example of the `presto -p` command and its output. A description of the output follows the example.

Example 3-1: Prestoserve Status

```
# presto –p
dirty = 0, clean = 61, inval = 0, active = 0
        1         2          3          4           5           6
       count  hit rate  clean hits  dirty hits  allocations  passes
write:  1188     65%       595         182          93         318
 read:     6      0%         0           0           0           6
total:  1194     65%       595         182          93         324
state = UP, size = 0x7e000 bytes
statistics interval: 00:00:35  (35 seconds)
write cache efficiency: 21%      7
All batteries are ok          8
```

For each cache read or write operation, Prestoserve increments a counter. A hit occurs when a requested block is matched to a block in a buffer. The previous example shows the following information:

1 The `count` specifies the sum of the `clean hits`, `dirty hits`, `allocations`, and `passes` counters.

2 The `hit rate` percentage is the ratio of the `clean hits` and `dirty hits` counters to the `count`.

Note

> The `hit rate` percentage for Prestoserve cache writes indicates the effectiveness of the Prestoserve cache. If the number of read operations is high in proportion to the total count of read and write operations (75% or more), you may improve system performance by increasing the amount of main memory allocated to the file system buffer cache.

3 The `clean hits` counter specifies the number of hits on the clean buffers.

4 The `dirty hits` counter specifies the number of hits on the dirty buffers. Each dirty hit represents a physical disk write that was avoided entirely.

5 The `allocations` counter specifies the number of new buffers that had to be allocated for disk block images.

6 The `passes` counter specifies the number of I/O operations that Prestoserve passed directly to the real device driver.

7 The `write cache efficiency,` percentage is computed from the ratio of write dirty hits to the number of writes copied into the Prestoserve cache (write count - write passes).

8 The battery state indicates the condition of the batteries. In general, the battery state can be `OK`, `low`, or `disabled`, but some processors support chargeable batteries and use self-tests to determine if a battery

needs charging. If your processor supports chargeable batteries and is running Prestoserve locally, the battery state can also be specified as `in self-test` or `is charging`.

Note that if you use the `-p` option with the `-h` option (or if you use the `dxpresto` command), batteries that are being self-tested or charged will be displayed as `disabled`.

The following is an example of the `presto` command with the `-l` and the `-p` options specified:

```
# presto -lp
dirty = 54, clean = 3, inval = 0, active = 4
       count hit rate clean hits dirty hits allocations passes
write:  1236      65%          0        808         421      6
 read:     2       0%          0          0           0      2
total:  1238      65%          0        808         421      8
state = UP, size = 0x7e000 bytes
statistics interval: 00:00:10  (10 seconds)
write cache efficiency: 66%
All batteries are ok
/dev/rz0a on /
/dev/rz0g on /usr
/dev/rz1a on /rz1a
/dev/rz1g on /usr/staff
/dev/rz2c on /rz2c
mmate3:/usr/staff on /mmate3
sunk:/home on /sunk
```

The following example shows the output of the `presto` command when you use the `-h` option with the `-p` option:

```
# presto -h mmate3 -p
mmate3:
dirty = 0, clean = 0, inval = 126, active = 0
       count hit rate clean hits dirty hits allocations passes
write:    46      61%          0         28          17      1
 read:     0     100%          0          0           0      0
total:    46      61%          0         28          17      1
state = DOWN, size = 0xffc00 bytes
statistics interval: 00:00:01  ( seconds)
write cache efficiency: 62%
All batteries are ok
```

3.2.6 Using dxpresto to Administer and Monitor Prestoserve

The `dxpresto` command starts the worksystem software application that graphically displays information about Prestoserve in a window. You can use the command to monitor Prestoserve activity. It also allows you to enable or disable Prestoserve on machines that allow that operation.

You can invoke the `dxpresto` command on a machine running Prestoserve to obtain that machine's Prestoserve information, or you can specify a remote

host running Prestoserve to obtain that host's Prestoserve information by using remote procedure calls.

Note

Because `dxpresto` is a worksystem software application, the DISPLAY environment variable must be set to a machine that is running the worksystem software. See `putenv`(3) for information on setting environment variables.

The `dxpresto` command displays the following information:

- Prestoserve state
- Number of kilobytes of nonvolatile memory that the Prestoserve cache is utilizing
- Amount of time that Prestoserve has been enabled
- Battery condition
- Current state of all the Prestoserve buffers
- History of Prestoserve writes per second
- History of Prestoserve cache hits per second
- Prestoserve statistics for write, read, and total operations
- Prestoserve and `dxpresto` command error messages

The `dxpresto` command also allows you to modify the displayed information by:

- Changing the name of the system to be monitored
- Changing the Prestoserve state to `Enabled` (`UP`) or `Disabled` (`DOWN`)
- Changing the interval of time between Prestoserve queries
- Displaying the Prestoserve statistics for write, read, and total operations since Prestoserve was last enabled
- Displaying Prestoserve statistics for write, read, and total operations since Prestoserve was last queried
- Displaying Prestoserve statistics for write, read, and total operations since a specific time

The `dxpresto` command has the following syntax:

/usr/sbin/dxpresto [*hostname*]

The *`hostname`* variable specifies the name of the machine you want to monitor; this machine must be running the Prestoserve software. If you do not specify the *`hostname`* variable, the local machine running the Prestoserve software is monitored. If the *`hostname`* variable is not specified and the local machine is not running the Prestoserve software, the `dxpresto` window opens but is not functional until you enter the name of a host running the Prestoserve software in the Host field in the `dxpresto` window. See `dxpresto`(8X) for more information.

Note

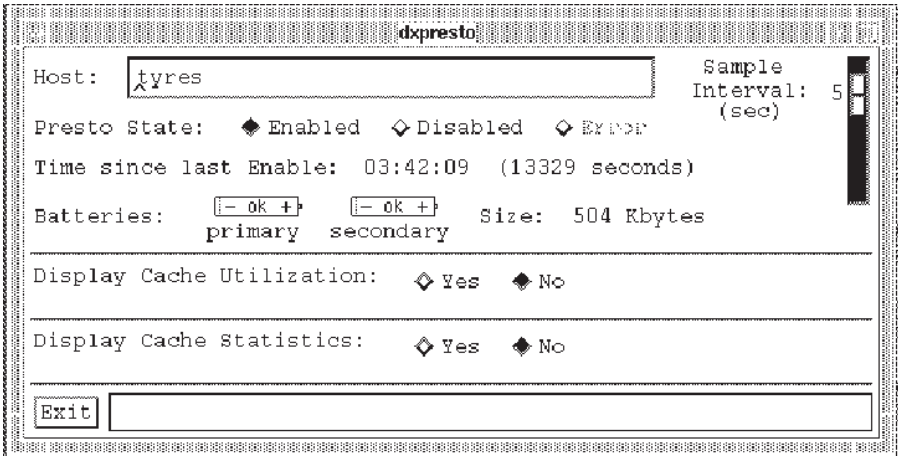
To use the `dxpresto` command to monitor a machine's Prestoserve activity, the `prestoctl_svc` daemon must be running on that machine. Refer to Chapter 2 for information on the `prestoctl_svc` daemon.

An example of the `dxpresto` command is as follows:

```
# dxpresto tyres
```

Figure 3-1 shows a `dxpresto` window.

Figure 3-1: dxpresto Window



ZK-0481Un-R

Figure 3-1 shows the following:

`Host`
: This field shows the host that you are monitoring. You can type another name in the field and press the Return key to monitor that host.

`Presto State`
: These buttons show the Prestoserve state, either `Enabled`, `Disabled`, or `Error`. If the machine being monitored is running the `prestoctl_svc` daemon with the `-n` option, you can change the machine's Prestoserve state to either `Enabled` or `Disabled`. You cannot change an `Error` state; contact your Digital Customer Services representative if an `Error` state occurs.

`Sample Interval`
: This slider shows the interval of time between Prestoserve queries; it allows you to change that interval. When you invoke the `dxpresto` command, the default `Sample Interval` is 5; therefore Prestoserve information is gathered every 5 seconds. If you want Prestoserve queried more often, move the slider to the left and click on MB1 until 2 appears for example; Prestoserve is then queried every two seconds.

`Time since last Enable`
: This field shows the time since Prestoserve was last enabled. The time is displayed in hours, minutes, and seconds and total number of seconds.

`Batteries`
: These graphics show the state of the Prestoserve backup battery system. An intact battery with the word `ok` indicates that the battery has sufficient power. An intact battery with the word `low` indicates that the battery's power is low. A broken battery indicates that the battery is

disabled. Prestoserve goes into the ERROR state when the backup battery power falls below a minimum amount. Refer to your hardware documentation to determine the minimum amount of backup battery power. Contact your hardware Field Service representative if a battery has insufficient power or is disabled.

`Size`
: This field displays the number of kilobytes of nonvolatile memory that the Prestoserve cache is utilizing. Note that Prestoserve can utilize less than the default maximum size of its Prestoserve cache if you changed the cache size with the `presto -s` command.

`Display Cache Utilization`
: These buttons allow you to display graphs that demonstrate how the Prestoserve cache is being utilized.

`Display Cache Statistics`
: These buttons allow you to display the cache statistics table.

`Exit`
: This button allows you to exit the `dxpresto` window.

Message bar
: This area at the bottom of the window displays informational and error messages for the `dxpresto` command and for Prestoserve. For example, if the `prestoctl_svc` daemon with the `-n` option is not running on the machine you are monitoring, then a message is displayed indicating that changes to Prestoserve operation are not allowed.

  Error messages, such as those indicating RPC communication failure, are displayed on the terminal from which you invoked `dxpresto`.

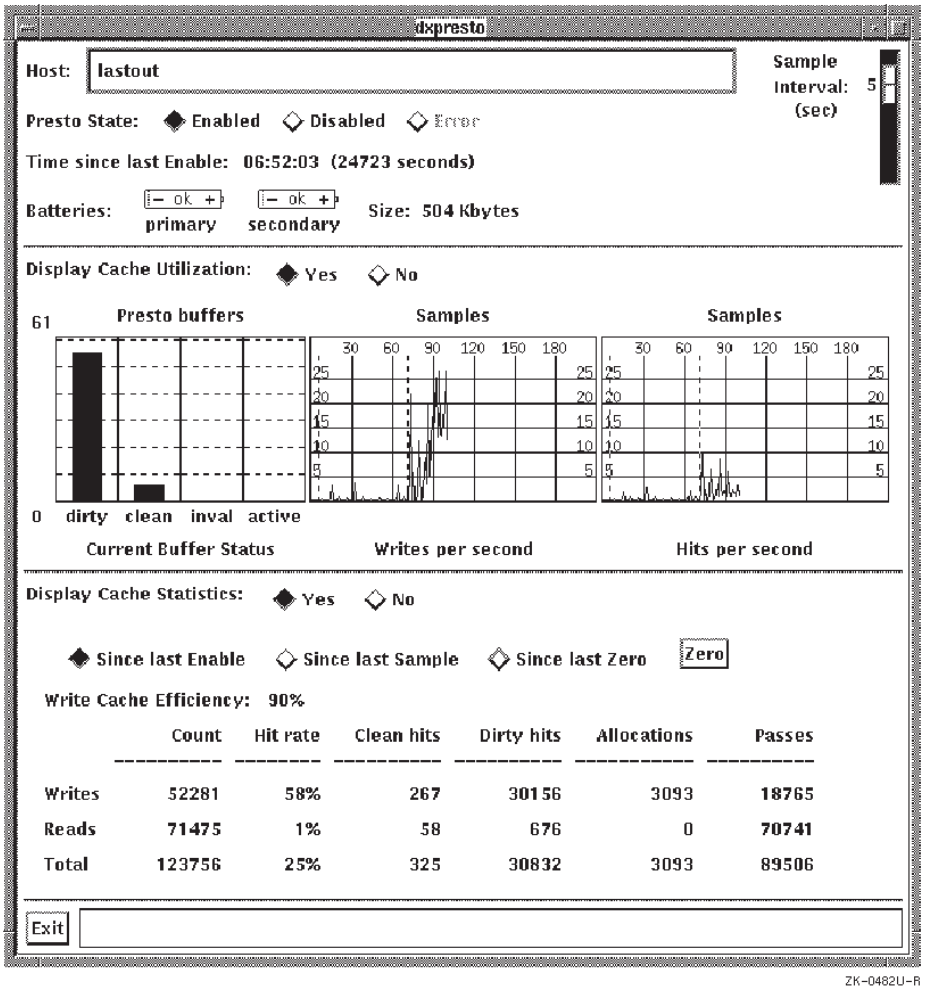
Figure 3-2 shows an example of the `dxpresto` window with both the `Display Cache Utilization` graphs and the `Display Cache Statistics` table displayed. The example shows the `Writes per second` and `Hits per second` trend line graphs. Each point in the horizontal axis of each graph represents a sample time interval as determined by the `Sample Interval` slider; the maximum number of samples that can be shown is 210. When you reach the maximum number of samples, the graph shifts to the left so you can see at least the last 105 samples, which is half the maximum number of samples. If you choose 5 in the `Sample Interval` slider, Prestoserve is queried every 5 seconds; therefore it takes 1050 seconds (5 x 210) to obtain the maximum of 210 samples.

The vertical axis shows the average number of writes performed per second within the sample time interval. For example, if you choose 2 in the `Sample Interval` slider, Prestoserve is queried every 2 seconds, and each point in the graph shows the average number of writes performed within the interval of 2 seconds. If the graph shows that an average of 5 writes per second were

performed within 2 seconds, Prestoserve actually performed 10 writes within those 2 seconds.

If you change hosts, each graph displays a vertical line of dashes to distinguish the new host's information from the previous host's information.

Figure 3-2: Expanded dxpresto Window



ZK-0482U-R

Figure 3-2 shows the following:

`Current Buffer Status`
: This bar graph shows how the Prestoserve cache operations are distributed among the Prestoserve buffer states, which are described in Section 3.1.1. The vertical axis shows the maximum number of objects

or disk blocks that the entire Prestoserve cache can contain. The sum of the four bars is the total number of buffers used in the Prestoserve cache. Note that the size of the Prestoserve cache can be changed by using the `presto -s` command. See Section 3.3.3 for more information.

`Writes per second`
: This trend line graph shows a recent history of the average number of writes per second over the time intervals that are determined by the `Sample Interval` slider.

`Hits per second`
: This trend line graph shows a recent history of the average number of Prestoserve cache hits per second over the time intervals that are determined by the `Sample Interval` slider. The Prestoserve cache hits represent the total number of clean and dirty read and write hits.

`Since last Enable`
: This button allows you to display Prestoserve statistics since Prestoserve was last enabled. This is useful when you want to determine how Prestoserve performs over a long period of time.

`Since last Sample`
: This button allows you to display the Prestoserve statistics for each sample time interval as determined by the `Sample Interval` slider. If no Prestoserve activity occurs during the time interval, the numbers in the statistics table remain at zero. For example, if the `Sample Interval` slider is set to 5 and the `Since last Sample` button is enabled, the statistics table shows the Prestoserve statistics for each interval of 5 seconds.

`Since last Zero`
: This button allows you to display Prestoserve statistics since you clicked on the `Zero` button. This button allows you to determine how Prestoserve performs over a specific period of time.

`Zero`
: This button sets the numbers in the table to zero, allowing you to specify a time reference for the Prestoserve statistics table. At a later time, you can click on the `Since last Zero` button to display the Prestoserve statistics since you clicked on the `Zero` button.

`Write Cache Efficiency`
: This field shows the ratio of write dirty hits to the number of writes copied into the Prestoserve cache.

Prestoserve statistics table
: This table is similar to the information that is displayed when you use the `presto -p` command. For each Prestoserve cache read or write

operation, Prestoserve increments an appropriate counter. The table shows the following:

`count`
Specifies the sum of the `clean hits`, `dirty hits`, `allocations`, and `passes`

`hit rate percentage`
Specifies the ratio of `clean hits` and `dirty hits` to the total `count`

`clean hits`
Specifies the number of hits on the clean buffers

`dirty hits`
Specifies the number of hits on the dirty buffers (each dirty hit represents a physical disk write that was avoided entirely)

`allocations`
Specifies the number of new buffers that had to be allocated for the disk block images

`passes`
Specifies the number of I/O operations that Prestoserve passed directly to the actual device driver

3.3 Handling the Prestoserve Cache

The following sections describe how to write the contents of the cache to disk, how to reset Prestoserve and clear the cache, and how to change the size of the cache.

3.3.1 Writing the Contents of the Cache to Disk

You can use the `presto` command with the `-F` option to write the contents of the Prestoserve cache to the available disks but keep the contents of the cache intact.

If the `-F` option is used and the Prestoserve state is `UP`, the contents of the cache are written to disk, and the state remains `UP`. If the Prestoserve state is `DOWN`, then there is nothing to write to disk, and the state remains `DOWN`.

If the Prestoserve state is `ERROR`, as much of the contents of the cache as possible is written to disk. Note that, unlike the `-R` option, the data in the cache remains after it is written to disk. The state remains `ERROR` until all the cache data is successfully written to disk. Note that if you cannot write all the cache data to disk and the state remains `ERROR`, you can use the `presto -R` command to reset Prestoserve, clear the cache, and set the state to `DOWN`.

The `presto -F` command can be used to flush dirty Prestoserve buffers to a disk that was temporarily disabled. For example, if a disk is powered down or disconnected from a bus, the Prestoserve cache could enter the ERROR state. When the disk is again available, you can use the `presto -F` command to move the cache data to disk and change the Prestoserve state from ERROR to UP.

3.3.2 Resetting Prestoserve and Clearing the Cache

If you are unable to clear the contents of the Prestoserve cache and write the data to disk, you can force Prestoserve out of the ERROR state. You reset Prestoserve and clear the cache by using the `presto` command with the `-R` option. The `-R` option writes as much of the Prestoserve cache data as possible to the appropriate disks, discards the data it cannot write, purges all Prestoserve buffers, and sets the Prestoserve state to DOWN.

Note

The `-R` option clears the Prestoserve cache by writing the data to the appropriate disks if possible. If a disk is unavailable, the data from the cache is lost, so you should use the option carefully.

Unlike the `-d`, `-D`, and `-F` options, the `-R` option discards the Prestoserve cache data that it could not write to disk. The option is useful when cache data is not needed. Note that the `-R` option takes effect before the `-u` or `-U` option.

In the following example, the `-R` option changes the Prestoserve state to DOWN:

```
# presto -Rp
dirty = 0, clean = 61, inval = 0, active = 0
       count hit rate clean hits dirty hits allocations passes
write:  1188      65%        595        182          93    318
 read:    10       0%          0          0           0     10
total:  1198      65%        595        182          93    328
state = DOWN, size = 0x7e000 bytes
statistics interval: 00:00:00  (0 seconds)
write cache efficiency: 0%
All batteries are ok
```

3.3.3 Changing the Cache Size

You can use the `presto` command with the `-s` option to change the size of the Prestoserve cache to the specified number of bytes. The size of the Prestoserve cache should be specified in the Prestoserve hardware documentation or product description. The `presto -s` command has the following syntax:

presto –s *size*

You can specify the *size* variable using decimal or hexadecimal conventions. For example, both 262144 and 0x40000 represent 256 kilobytes.

You may want to use the –s option to determine how Prestoserve performs with a reduced amount of nonvolatile memory. Note that the size of the Prestoserve cache cannot exceed the default maximum size; the default maximum size is used if you specify a size larger than this size. Refer to your processor hardware documentation for information about the default maximum size of the Prestoserve cache.

If you specify the –s option and the current Prestoserve state is UP, the state is set to DOWN, the Prestoserve cache is resized, and the state is set to UP.

For example, the following command changes the size of a Prestoserve cache to 512 kilobytes:

```
# presto –h mate –s 0x80000 –p
mate:
dirty = 119, clean = 3, inval = 0, active = 4
       count hit rate clean hits dirty hits allocations passes
write:  1350      66%          0        893         455      2
 read:     0     100%          0          0           0      0
total:  1350      66%          0        893         455      2
state = UP, size = 0x80000 / 0xffc00 bytes
statistics interval: 00:00:00  (0 seconds)
write cache efficiency: 33%
All batteries are ok
```

3.4 Displaying Debugging Information

You can use the presto command with the –v option to obtain information that you can use to debug Prestoserve operation. The –v option is used with other presto command options and displays extra information to standard output.

3.5 Checking Prestoserve

The system administrator can check to determine if Prestoserve is working properly by performing the following steps:

1. Log in to the server as root and disable Prestoserve:

   ```
   # presto –d
   ```

   See Chapter 3 for information on the `presto` command.

2. Log in to a client system and mount one of the server's file systems that is exported by the NFS and that has at least as much available space as the size of the client's `/vmunix` file or some other large file. Use a mount point where the client can create files. The following example uses `/usr/tmp` as a mount point; the commands establish the client's level of performance without Prestoserve:

   ```
   client% mount  server:/usr/tmp  /mnt
   client% cd /mnt
   client% /bin/time  cp  /vmunix  bigfile
     34.1 real         0.0 user         1.1 sys
   client% rm  bigfile
   ```

3. Enable Prestoserve on the server:

   ```
   server# presto –u
   ```

4. Establish the client's level of performance with Prestoserve:

   ```
   client% /bin/time  cp  /vmunix  bigfile
    10.3 real         0.0 user         1.1 sys
   client% rm  bigfile
   client% cd /
   client% umount  /mnt
   ```

The real time reported by the commands in step 4 is expected to be about one third of (or about three times faster than) the real time reported by the commands in step 2 while Prestoserve was disabled. Your improvement will vary, but the expected range is between three and five times faster with Prestoserve enabled. If you see much less than a factor of three, make sure that all the other clients are idle and that your network is not being used by others at this time.

Recovering from System Failures 4

This chapter describes how to manage Prestoserve under abnormal conditions. These conditions include cases when the system is shut down abnormally and when disks accelerated by Prestoserve encounter errors or failure.

Processor-specific information about recovering from system crashes and descriptions of any Prestoserve console commands are contained in the hardware documentation for your processor.

4.1 Normal and Abnormal System Shutdowns

A normal (clean) shutdown occurs when the system is halted by using either the `shutdown`, the `halt`, or the `reboot` command. If a normal shutdown occurs or if you unmount a device that was accelerated, the contents of the Prestoserve cache are flushed (moved) to the appropriate disks.

In addition, you can cleanly shut down Prestoserve by using the `presto` command with the `-d` or `-D` option before you halt a running system. The command flushes the Prestoserve cache, and Prestoserve enters the `DOWN` state. When Prestoserve is in the `DOWN` state, all requests are directly passed to the device drivers, and other forms of system shutdown do not affect system operation with respect to Prestoserve. Refer to Chapter 3 for more information about Prestoserve states.

An abnormal (unclean) shutdown results from a power or hardware failure, operating system software failure, or by manually halting or restarting the system when Prestoserve is still in the `UP` state. After an abnormal shutdown, the Prestoserve cache may contain data that Prestoserve was unable to flush to disk. In this case, it is important to ensure that the cache data is not lost or does not corrupt your disks.

In most cases, after an abnormal shutdown, data in the Prestoserve cache is recovered automatically when you reboot; that is, the data is flushed to the appropriate disks. However, if you reboot a different kernel or change your system configuration, you may encounter problems recovering the cache data. The following sections describe how to handle cache data.

4.1.1 Recovering Cache Data After an Abnormal Shutdown

The Prestoserve cache usually contains data when it is in the UP state. If your system shuts down abnormally, data may remain in the Prestoserve cache. A Prestoserve cache that contains data is referred to as a dirty cache.

Usually, if you reboot the system, the system startup procedure repairs file system inconsistencies. This process flushes the cache and moves the data to the appropriate disks. Therefore, Prestoserve can usually recover easily after an abnormal shutdown, and no user action is necessary.

However, if your system shut down abnormally, and then you changed your system or hardware configuration, you may encounter some problems when the system reboots. Prestoserve uses physical device numbers internally to identify data blocks. If you reconfigure your system or hardware after an abnormal shutdown, data in the Prestoserve cache may be flushed to the wrong device or lost, or file systems may be corrupted. This could happen in the following cases:

- You installed a kernel that has different disk device numbers than the kernel that was last used with Prestoserve.
- You booted a non-Prestoserve kernel with disks that previously were accelerated.
- You changed your device configuration.
- You removed or added a disk controller.

Note

To ensure that you can recover the Prestoserve cached data after an abnormal shutdown, do not reboot the generic kernel (`genvmunix`) if the target kernel (`vmunix`) will not boot. If you have renumbered device numbers, the generic kernel will not be aware of those changes and, as a result, when Prestoserve attempts to access the filesystem drivers to restore its cashed data, the data may be lost or written to the wrong place.

To avoid this problem, Digital recommends that you create a copy of your running target kernel with Prestoserve configured into it and boot that kernel in the event that your target kernel is corrupted after an abnormal shutdown and cannot be booted.

If you want to reconfigure your system, you should ensure that no data is in the Prestoserve cache and shut down the system cleanly.

If you cannot recover the Prestoserve cache data when the system reboots, a diagnostic message is displayed. Prestoserve prompts you to confirm that you want to continue rebooting the system. You are given the option to do one of the following:

- Discard the Prestoserve cache data
- Write the data to the intended disks
- Halt the machine

If you choose to continue rebooting, the system startup procedure checks the file systems and performs any corrections that it knows are correct. During the reboot, you can note the extent of the disk data corruption. You may have to use a file system repair program such as the `fsck` command after the system reboots to repair any file system inconsistencies. You can then recover data by restoring file systems from backups, by rerunning programs, or by reentering data if necessary.

4.1.2 Recovering Cache Data After Replacing a CPU Board

If the system was shut down abnormally, and you installed a new CPU board, the power-up diagnostics will indicate that the CPU board identification number does not match the Prestoserve cache identification number.

If you reboot the system and the Prestoserve cache contains data, you are given the option to do one of the following:

- Discard the Prestoserve cache data
- Write the data to the intended disks
- Halt the machine

Usually, you can continue to reboot the system with no adverse affects.

4.1.3 Handling Failed Prestoserve Hardware

If the Prestoserve hardware fails the power-up diagnostics, install new hardware. If the Prestoserve cache contained data when it failed, the data is lost.

You may have to use a file system repair program such as the `fsck` command after the system reboots to repair any file system inconsistencies. You can then recover data by restoring file systems from backups, by rerunning programs, or by reentering data if necessary.

4.1.4 Moving the Prestoserve Hardware

If the Prestoserve cache contains data, and the Prestoserve hardware is moved to another system along with the disks, the power-up diagnostics will indicate that the CPU board identification number does not match the Prestoserve hardware identification number.

When you boot the system and the Prestoserve cache contains data, you are given the option to do one of the following:

- Discard the Prestoserve cache data
- Write the data to the intended disks
- Halt the machine

Usually, you can continue to reboot the system with no adverse affects. To avoid any problems, you should shut down the system cleanly before moving the Prestoserve hardware.

4.2 Disk Failures

The following sections describe how Prestoserve manages disk failures. Temporary disk failures are those that can be fixed without requiring major repair, such as a disk being off line or write protected. Serious disk failures (such as a disk head crash) entail significant repair and may cause data to be lost.

4.2.1 Temporary Disk Failures

Because Prestoserve caches disk blocks, data used by an application may not be written to disk for some time. If a disk fails with Prestoserve enabled, the system will not notice the failure until Prestoserve attempts to flush its cache. When this occurs, Prestoserve enters the ERROR state and attempts to flush its entire cache immediately. If the cache is flushed successfully, Prestoserve leaves the ERROR state, and no other user action is necessary.

However, if the cache cannot be completely flushed, Prestoserve effectively becomes a read-only data repository, and subsequent writes that do not match blocks already in the Prestoserve cache are passed directly through to the actual disk driver.

When Prestoserve is in the ERROR state, new data written to a block already in the Prestoserve cache replaces the existing block within the cache. This block is then flushed synchronously to the disk to see if the error condition still exists. If the error still exists, the application receives the error from the failed write operation.

If the write succeeds, Prestoserve leaves the ERROR state if it can successfully flush all of its buffers. The first time Prestoserve enters the ERROR state, a message similar to the following is displayed on the console terminal, listing the major and minor numbers of the actual device:

```
presto: error on dev (%d, %d)
```

A device-specific error message from the actual device driver may have been previously displayed. Note that any retries normally performed by a disk driver in an error condition are still performed for each I/O request by Prestoserve.

Prestoserve exits the ERROR state only when it can successfully flush its entire cache to the disk. It only attempts to flush its cache when a request is made to write a block that is already in the cache and when this block is successfully written to disk. Requests to write blocks not already in the cache are passed directly to the actual disk driver. Thus, Prestoserve does not accelerate writes when it is in the ERROR state, and Prestoserve may remain in the ERROR state even after the disk problem is corrected if the cache data cannot be moved to disk.

If you can locate the cause of the I/O failure and fix it, reenable Prestoserve so it can verify that the error was corrected and exit the ERROR state. You can accomplish this by issuing the following command:

```
# presto -F
```

Rebooting the system also causes Prestoserve to flush its cache to the appropriate disks if they are available.

4.2.2 Serious Disk Failures

If you must replace a disk because of a major I/O failure that is not easily repaired, you can use the `presto -R` command, which attempts to flush all cached data and then destroys any data that cannot be written to disk. Before you replace a bad disk, use the `presto -R` command to ensure that you do not flush disk blocks logically belonging to the bad disk to the new disk device, thus corrupting the data on the new disk. However, if you install a new disk that contains no valid data, you can flush the cached data to it because there is no data on the disk to corrupt.

If there are disk errors but you want to continue running with the faulty disk disabled, perform the following steps:

1. Use the `presto -R` command to write as much of the Prestoserve cache data as possible to the appropriate disks, discard any data it could not write, purge the Prestoserve buffers, and disable Prestoserve.
2. Unmount the bad disk.
3. Use the `presto -u` command to enable Prestoserve on the viable disks.

The Prestoserve ERROR state affects all accelerated disks, so you must disable the defective disk before reenabling Prestoserve on the viable disks. Refer to Chapter 3 for information about the `presto` command.

Index

E

F

G

H

I

L

M

N

P

R

S

W

How to Order Additional Documentation

Technical Support

If you need help deciding which documentation best meets your needs, call 800-DIGITAL (800-344-4825) before placing your electronic, telephone, or direct mail order.

Electronic Orders

To place an order at the Electronic Store, dial 800-234-1998 using a 1200- or 2400-bps modem from anywhere in the USA, Canada, or Puerto Rico. If you need assistance using the Electronic Store, call 800-DIGITAL (800-344-4825).

Telephone and Direct Mail Orders

Your Location	Call	Contact
Continental USA, Alaska, or Hawaii	800-DIGITAL	Digital Equipment Corporation P.O. Box CS2008 Nashua, New Hampshire 03061
Puerto Rico	809-754-7575	Local Digital subsidiary
Canada	800-267-6215	Digital Equipment of Canada Attn: DECdirect Operations KAO2/2 P.O. Box 13000 100 Herzberg Road Kanata, Ontario, Canada K2K 2A6
International	————	Local Digital subsidiary or approved distributor
Internal[a]	————	SSB Order Processing – NQO/V19 *or* U. S. Software Supply Business Digital Equipment Corporation 10 Cotton Road Nashua, NH 03063-1260

[a] For internal orders, you must submit an Internal Software Order Form (EN-01740-07).

Reader's Comments

Digital UNIX
Guide to Prestoserve
AA-PQT0D-TE

Digital welcomes your comments and suggestions on this manual. Your input will help us to write documentation that meets your needs. Please send your suggestions using one of the following methods:

- This postage-paid form
- Internet electronic mail: `readers_comment@zk3.dec.com`
- Fax: (603) 881-0120, Attn: UEG Publications, ZKO3-3/Y32

If you are not using this form, please be sure you include the name of the document, the page number, and the product name and version.

Please rate this manual:	Excellent	Good	Fair	Poor
Accuracy (software works as manual says)				
Completeness (enough information)				
Clarity (easy to understand)				
Organization (structure of subject matter)				
Figures (useful)				
Examples (useful)				
Index (ability to find topic)				
Usability (ability to access information quickly)				

Please list errors you have found in this manual:

Page	Description

Additional comments or suggestions to improve this manual:

What version of the software described by this manual are you using?

Name/Title Dept.

Company Date

Mailing Address

Email Phone

Do Not Cut or Tear − Fold Here and Tape

digital™

NO POSTAGE
NECESSARY IF
MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL
FIRST−CLASS MAIL PERMIT NO. 33 MAYNARD MA

POSTAGE WILL BE PAID BY ADDRESSEE

DIGITAL EQUIPMENT CORPORATION
UEG PUBLICATIONS MANAGER
ZKO3−3/Y32
110 SPIT BROOK RD
NASHUA NH 03062−9987

Do Not Cut or Tear − Fold Here

Cut on
Dotted
Line