



*Keeping the World's Business-Critical Information Available*

# DiskXtender<sup>®</sup> UNIX<sup>®</sup>

## FTP/NFS User Guide

Release 2.5

© 2003, Legato Systems, Inc. All rights reserved. This product may be covered by one or more of the following patents: U.S. 5,359,713; 5,519,853; 5,649,152; 5,799,141; 5,812,748; 5,835,953; 5,978,565; 6,073,222; 6,085,298; 6,145,089; 6,308,283; 6,324,654; 6,338,126. Other U.S. and international patents pending.

DiskXtender UNIX, Release 2.5, FTP/NFS User Guide

February 2003

e2-2015-01

Legato and the Legato logo are registered trademarks, and Legato NetWorker, NetWorker, LM:, Celestra, GEMS, SmartMedia, Co-StandbyServer, RepliStor, SnapShotServer, QuikStartz, SAN Academy, AlphaStor, ClientPak, Xtender, XtenderSolutions, DiskXtender, ApplicationXtender, ArchiveXtender, and EmailXtender are trademarks or registered trademarks of Legato Systems, Inc. This is a nonexhaustive list of Legato trademarks, and other trademarks may be the property of their respective owners.

The following may be trademarks or registered trademarks of the companies identified next to them, and may be used in this document for identification purposes only.

Acrobat, Adobe / **Adobe Systems, Inc.**

Apple, Macintosh / **Apple Computer, Inc.**

AIX, DB2, DB2 Universal Database, DYNIX, DYNIXptx, IBM, Informix, Lotus, Lotus Notes, OS/2, PTX, ptx/ADMIN, Raid Plus, ServeRAID, Sequent, Symmetry, Tivoli, / **IBM Corporation**

InstallShield / **InstallShield Software Corporation**

UNIX / **X/Open Company Ltd<sup>a</sup>**

Access Logix, Celerra, CLARiiON, EMC, EMC<sup>2</sup>, MirrorView, MOSAIC:2000, Navisphere, SnapView, SRDF, Symmetrix,

TimeFinder / **EMC Corporation**

Date ONTAP, NetApp, NetCache, Network Appliance, SnapMirror, SnapRestore / **Network Appliance, Inc.**

DLTtape / **Quantum Corporation**

Fujitsu / **Fujitsu, Ltd.**

Hewlett-Packard, HP, HP-UX, HP Tru64, HP TruCluster / **Hewlett-Packard Company**

IntraNetWare, NetWare, Novell / **Novell, Inc.**

Microsoft, MS-DOS, SQL Server, Windows, Windows NT / **Microsoft Corporation**

Intel / **Intel Corporation**

Linux / **Linus Torvalds**

Netscape, Netscape Navigator / **Netscape Communications Corporation**

IRIX, OpenVault, SGI / **Silicon Graphics, Inc.**

Oracle, Oracle8i, Oracle9i / **Oracle Corporation**

Red Hat / **Red Hat, Inc.**

R/3, SAP / **SAP AG**

Caldera Systems, SCO, SCO OpenServer, UnixWare / **Caldera, Inc.**

SPARC / **SPARC International, Inc.<sup>b</sup>**

Solaris, Solstice Backup, Sun, SunOS, Sun StorEdge / **Sun Microsystems, Inc.**

REELbackup, StorageTek / **Storage Technology Corporation**

Sybase / **Sybase, Inc.**

VERITAS / **VERITAS Software Corporation**

Unicenter / **Computer Associates International, Inc.**

Turbolinux / **Turbolinux, Inc.**

SuSE / **SuSE, Inc.**

Check Point, FireWall-1 / **Check Point Software Technologies, Ltd.**

#### Notes:

- a. UNIX is a registered trademark in the United States and other countries, licensed exclusively through X/Open Company, Ltd.
- b. Products bearing SPARC trademarks are based on an architecture developed by Sun Microsystems, Inc.

All other brand or product names may be trademarks or registered trademarks of their respective owners.

## Legato Systems, Inc. End-User License Agreement

THIS PRODUCT CONTAINS CERTAIN COMPUTER PROGRAMS AND OTHER PROPRIETARY MATERIAL, THE USE OF WHICH IS SUBJECT TO THIS END-USER LICENSE AGREEMENT (THE "AGREEMENT"). DO NOT PROCEED WITH THE INSTALLATION OF THIS PRODUCT UNTIL YOU (LATER DEFINED AS LICENSEE) HAVE READ THIS AGREEMENT AND AGREE TO BE BOUND BY AND BECOME A PARTY TO THIS AGREEMENT. BY PROCEEDING WITH THE INSTALLATION OF THIS PRODUCT (OR AUTHORIZING ANY OTHER PERSON TO DO SO), YOU AND YOUR COMPANY ACCEPT THIS AGREEMENT AND YOU REPRESENT THAT YOU ARE AUTHORIZED TO DO SO. IF YOU ARE ACCESSING THE PRODUCT ELECTRONICALLY INDICATE YOUR ACCEPTANCE OF THESE TERMS BY SELECTING THE "ACCEPT" BUTTON AT THE END OF THE AGREEMENT. IF YOU DO NOT AGREE TO THE TERMS OF THIS AGREEMENT, YOU MAY RETURN THIS PRODUCT, ALL MEDIA AND DOCUMENTATION, AND PROOF OF PAYMENT, TO THE PLACE YOU OBTAINED THEM FOR A FULL REFUND WITHIN THIRTY (30) DAYS OF FIRST ACQUIRING THIS PRODUCT OR, IF THE PRODUCT IS ACCESSED ELECTRONICALLY, SELECT THE "DECLINE" BUTTON AT THE END OF THIS AGREEMENT AND RETURN PROOF OF PAYMENT IN ACCORDANCE WITH THE ABOVE REFERENCED RETURN/REFUND PROCESS. WRITTEN APPROVAL IS NOT A PREREQUISITE TO THE VALIDITY OR ENFORCEABILITY OF THIS AGREEMENT AND NO SOLICITATION OF ANY SUCH WRITTEN APPROVAL BY OR ON BEHALF OF LEGATO SHALL BE CONSTRUED AS AN INFERENCE TO THE CONTRARY. IF YOU HAVE ORDERED THIS PRODUCT, LEGATO'S ACCEPTANCE IS EXPRESSLY CONDITIONAL ON YOUR ASSENT TO THESE TERMS TO THE EXCLUSION OF ALL OTHER TERMS; IF THESE TERMS ARE CONSIDERED AN OFFER BY LEGATO, ACCEPTANCE IS EXPRESSLY LIMITED TO THESE TERMS.

### 1. DEFINITIONS

**1.1 Authorization Code:** means the code provided to Licensee by Legato for permanent authorization to use the Software. The Authorization Code is provided to Licensee once the Enabler Code is registered with Legato.

**1.2 Documentation:** means any user reference materials on any media, provided by Legato for use with the Software.

**1.3 Enabler Code:** means the code provided by Legato for activation of the Software.

**1.4 Licensee:** means the person or entity acquiring this License or for whom this License was acquired.

**1.5 Software:** means the object code copy of the software program provided to You in association with this Agreement, together with the associated original electronic media and all accompanying manuals and other documentation, and together with all enhancements, upgrades, and extensions thereto that may be provided by Legato to You from time to time.

### 2. OWNERSHIP AND ADMINISTRATION OF SOFTWARE

**2.1 Ownership and Title.** As between the parties, Legato, and its licensors, own and shall retain all right, title, and interest in and to: (i) the Software including all intellectual property rights embodied therein; (ii) all of the service marks, trademarks, trade names, or any other designations associated with the Software; and (iii) all copyrights, patent rights, trade secret rights, and other proprietary rights relating to the Software.

**2.2 Software Activation.** Legato employs Enabler Codes and Authorization Codes that enable the use of the Software. The Software is shipped in a "Media Kit" which consists of object code software on CD-ROM and an Enabler Code for initial activation of the Software or the Software and Enabler Code may be delivered electronically. Once Legato receives confirmation from Licensee that the Enabler Code is installed and is provided with the host ID information, Legato will provide the Authorization Code to Licensee. Legato administers the generation and distribution of Enabler and Authorization Codes, which administration may be modified by Legato from time to time.

**2.3 Administration of Software.** Legato may include on the media with the Software additional computer programs which are not currently licensed for use by Licensee and to which the Enabler Code or Authorization code will not permit access. Inclusion of such additional computer programs in no way implies a license from Legato and access or use of such programs is strictly prohibited unless Licensee procures the right to use any such program and the applicable Enabler Code is provided thereto.

### 3. LICENSE GRANT

**3.1 Grant.** Legato grants to Licensee a nonexclusive, nontransferable, nonsublicensable, perpetual, unless terminated in accordance with the provisions of this Agreement, license (the "License") to (i) use the Software installed in accordance with the Documentation and only on the licensed computer solely for its own internal operations; and (ii) move the Software temporarily in case of computer system malfunction. The License granted under this Agreement does not constitute a sale of the Software or any portion or copy of it. Licensee may not use the Software on more than one computer system unless otherwise specifically authorized by an explicit Software product, or additional licenses for additional computers are purchased. Rights not expressly granted are reserved by Legato. Where the Software is provided to Licensee at no charge for evaluation purposes only, the License granted is limited to a continuous thirty (30) day period,

commencing with the acceptance of this Agreement (the "**Evaluation Period**"). At the conclusion of the Evaluation Period, Licensee agrees to destroy the Software and certify its destruction to Legato, in writing, within ten (10) days, or shall return the Software to Legato or purchase a perpetual license.

**3.2 Copies.** Licensee may make copies of the Software provided that any such copy is: (i) created as an essential step in utilization of the Software on the licensed computer and is used in no other manner; or (ii) used for archival purposes to back up the licensed computers. All trademark and copyright notices must be reproduced and included on such copies. Licensee may not make any other copies of the Software.

**3.3 Restrictions on use.** Licensee shall not, and shall not aid, abet, or permit any third party to: (i) decompile, disassemble, or otherwise reverse engineer or attempt to reconstruct or discover any source code or underlying ideas or algorithms of the Software by any means whatsoever; (ii) remove any identification, copyright, or other notices from the Software; (iii) provide, lease, lend, use for timesharing or service bureau purposes; (iv) create a derivative work of any part of the Software; or (v) develop methods to enable unauthorized parties to use the Software. If EC law is applicable, the restrictions in Section 3.3 (i) are limited so that they prohibit such activity only to the maximum extent such activity may be prohibited without violating the EC Directive on the legal protection of computer programs. Notwithstanding the foregoing, prior to decompiling, disassembling, or otherwise reverse engineering any of the Software, Licensee shall request Legato in writing, to provide Licensee with such information or assistance and Licensee shall refrain from decompiling, disassembling, or otherwise reverse engineering any of the Software unless Legato cannot or has not complied with such request in a commercially reasonable amount of time.

**3.4 Purchase Orders.** Nothing contained in any purchase order, acknowledgment, or invoice shall in any way modify the terms or add any additional terms or conditions to this Agreement.

**3.5 Updates.** This section applies if the Software acquired is an update to the original Software (the "**Update**"). An Update does not constitute a legally licensed copy of the Software unless purchased as an Update to a previous version of the same Software. The Update may only be used in accordance with the provisions of this Agreement. The Update, together with the original Software, constitutes one (1) legally licensed copy of the Software.

**3.6 Evaluation License.** This Section applies if the Software is being used for an initial thirty (30) day evaluation period. The license is valid only for a period of thirty (30) days from the delivery of the Software, and is designed to allow Licensee the right to evaluate the Software during such period. In the event that Licensee desires to enter into a longer-term license agreement with Legato, Licensee shall obtain an appropriate Enabler and Authorization Code in accordance with Section 2.2 above, upon payment of applicable fees, which authorizes use of the Software after such evaluation period, but only subject to all of the terms and conditions of this Agreement. In the event Licensee determines not to enter into a licensing transaction with Legato at the end of such thirty (30) day evaluation period, then Licensee's rights under this Agreement shall terminate automatically and Licensee shall promptly return to Legato or destroy all copies of the Software and so certify to Legato.

### 4. MAINTENANCE AND SUPPORT

**4.1** Legato has no obligation to provide support, maintenance, upgrades, modifications, or new releases under this Agreement. Legato may provide such services under separate agreement.

## 5. LIMITED WARRANTY

5.1 **Media and Documentation.** Legato warrants that if the media or documentation are damaged or physically defective at the time of delivery of the first copy of the Software to Licensee and if defective or damaged product is returned to Legato (postage prepaid) within thirty (30) days thereafter, then Legato will provide Licensee with replacements at no cost.

5.2 **Limited Software Warranty.** Subject to the conditions and limitations of liability stated herein, Legato warrants for a period of thirty (30) days from the delivery of the first copy of the Software to Licensee that the Software, as delivered, will materially conform to Legato's then current published Documentation for the Software. This warranty covers only problems reported to Legato during the warranty period. For customers outside of the United States, this Limited Software Warranty shall be construed to limit the warranty to the minimum warranty required by law.

5.3 **Remedies.** The remedies available to Licensee hereunder for any such Software which does not perform as set out herein shall be either repair or replacement, or, if such remedy is not practicable in Legato's opinion, refund of the license fees paid by Licensee upon a return of all copies of the Software to Legato. In the event of a refund this Agreement shall terminate immediately without notice.

## 6. TERM AND TERMINATION

6.1 **Term.** The term of this Agreement is perpetual unless terminated in accordance with its provisions.

6.2 **Termination.** Legato may terminate this Agreement, without notice, upon Licensee's breach of any of the provisions hereof.

6.3 **Effect of Termination.** Upon termination of this Agreement, Licensee agrees to cease all use of the Software and to return to Legato or destroy the Software and all Documentation and related materials in Licensee's possession, and so certify to Legato. Except for the License granted herein and as expressly provided herein, the terms of this Agreement shall survive termination.

## 7. DISCLAIMER AND LIMITATIONS

7.1 **Warranty Disclaimer.** EXCEPT FOR THE LIMITED WARRANTY PROVIDED IN SECTION 5 ABOVE, LEGATO AND ITS LICENSORS MAKE NO WARRANTIES WITH RESPECT TO ANY SOFTWARE AND DISCLAIMS ALL STATUTORY OR IMPLIED WARRANTIES, INCLUDING WITHOUT LIMITATION WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR ARISING FROM A COURSE OF DEALING OR USAGE OF TRADE AND ANY WARRANTIES OF NONINFRINGEMENT. ALL SOFTWARE IS PROVIDED "AS IS" AND LEGATO DOES NOT WARRANT THAT THE SOFTWARE WILL MEET ANY REQUIREMENTS OR THAT THE OPERATION OF SOFTWARE WILL BE UNINTERRUPTED OR ERROR FREE. ANY LIABILITY OF LEGATO WITH RESPECT TO THE SOFTWARE OR THE PERFORMANCE THEREOF UNDER ANY WARRANTY, NEGLIGENCE, STRICT LIABILITY OR OTHER THEORY WILL BE LIMITED EXCLUSIVELY TO THE REMEDIES SPECIFIED IN SECTION 5.3 ABOVE. Some jurisdictions do not allow the exclusion of implied warranties or limitations on how long an implied warranty may last, so the above limitations may not be applicable.

## 8. LIMITATION OF LIABILITY

8.1 **Limitation of Liability.** EXCEPT FOR BODILY INJURY, LEGATO (AND ITS LICENSORS) WILL NOT BE LIABLE OR RESPONSIBLE WITH RESPECT TO THE SUBJECT MATTER OF THIS AGREEMENT UNDER ANY CONTRACT, NEGLIGENCE, STRICT LIABILITY, OR OTHER LEGAL OR EQUITABLE THEORY FOR: (I) ANY INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES, HOWEVER CAUSED AND WHETHER OR NOT ADVISED IN ADVANCE OF THE POSSIBILITY OF SUCH DAMAGES; OR (II) DAMAGES FOR LOST PROFITS OR LOST DATA; OR (III) COST OF PROCUREMENT OF SUBSTITUTE GOODS, TECHNOLOGY, SERVICES, OR RIGHTS; OR FOR AMOUNTS IN EXCESS OF THOSE RECEIVED BY LEGATO FOR THE PARTICULAR LEGATO SOFTWARE THAT CAUSED THE LIABILITY. Because some jurisdictions do not allow the exclusion or limitation of incidental or consequential damages, Legato's liability in such jurisdictions shall be limited to the extent permitted by law.

## 9. MISCELLANEOUS

9.1 **Governing Law.** This Agreement shall be governed by the laws of the State of California, as applied to agreements entered into and to be performed entirely within California between California residents, without regard to the

principles of conflict of laws or the United Nations Convention on Contracts for the International Sale of Goods.

9.2 **Government Restricted Rights.** This provision applies to Software acquired directly or indirectly by or on behalf of any government. The Software is a commercial software product, licensed on the open market at market prices, and was developed entirely at private expense and without the use of any government funds. All Software and accompanying Documentation provided in connection with this Agreement are "commercial items," "commercial computer software," and/or "commercial computer software documentation." Any use, modification, reproduction, release, performance, display, or disclosure of the Software by any government shall be governed solely by the terms of this Agreement and shall be prohibited except to the extent expressly permitted by the terms of this Agreement, and no license to the Software is granted to any government requiring different terms. Licensee shall ensure that each copy used or possessed by or for any government is labeled to reflect the foregoing.

9.3 **Export and Import Controls.** Regardless of any disclosure made by Licensee to Legato of an ultimate destination of the Products, Licensee will not directly or indirectly export or transfer any portion of the Software, or any system containing a portion of the Software, to anyone outside the United States (including further export if Licensee took delivery outside the U.S.) without first complying with any export or import controls that may be imposed on the Software by the U.S. Government or any country or organization of nations within whose jurisdiction Licensee operates or does business. Licensee shall at all times strictly comply with all such laws, regulations, and orders, and agrees to commit no act which, directly or indirectly, would violate any such law, regulation or order.

9.4 **Assignment.** This Agreement may not be assigned or transferred by Licensee without the prior written consent of Legato, which shall not be unreasonably withheld. Legato may assign or otherwise transfer any or all of its rights and obligations under this Agreement upon notice to Licensee.

9.5 **Sole Remedy and Allocation of Risk.** Licensee's sole and exclusive remedies are set forth in this Agreement. This Agreement defines a mutually agreed-upon allocation of risk, and the License price reflects such allocation of risk.

9.6 **Equitable Relief.** The parties agree that a breach of this Agreement adversely affecting Legato's intellectual property rights in the Software may cause irreparable injury to Legato for which monetary damages may not be an adequate remedy and Legato shall be entitled to equitable relief in addition to any remedies it may have hereunder or at law.

9.7 **No Waiver.** Failure by either party to enforce any provision of this Agreement will not be deemed a waiver of future enforcement of that or any other provision, nor will any single or partial exercise of any right or power hereunder preclude further exercise of any other right hereunder.

9.8 **Severability.** If for any reason a court of competent jurisdiction finds any provision of this Agreement, or portion thereof, to be unenforceable, that provision of the Agreement will be enforced to the maximum extent permissible so as to effect the intent of the parties, and the remainder of this Agreement will continue in full force and effect.

## 10. ENTIRE AGREEMENT

10.1 This Agreement sets forth the entire understanding and agreement between the parties and may be amended only in a writing signed by authorized representatives of both parties. No vendor, distributor, dealer, retailer, sales person, or other person is authorized by Legato to modify this Agreement or to make any warranty, representation, or promise which is different than, or in addition to, the warranties, representations, or promises made in this Agreement. No preprinted purchase order terms shall in any way modify, replace, or supersede the terms of this Agreement.

---

# TABLE OF CONTENTS

---

## CHAPTER 1 *Getting Started* 1

- Welcome to DiskXtender!* 1
- UNIX and DiskXtender* 1
- DiskXtender Features 3
- How DiskXtender Works 3
  - NFS* 3
  - FTP* 3
  - File Migration, Purging and Caching* 3
- Accessing DiskXtender 5

## CHAPTER 2 *NFS* 7

- Getting Started 8
  - NFS Commands* 8
  - DiskXtender NFS Commands* 8
  - Common NFS Replies and Errors* 9
  - Sample NFS Session* 9

## CHAPTER 3 *FTP* 13

- About FTP 13
- Logging In 13
- UNIX FTP Commands 14
- DiskXtender FTP Commands 21
- Common FTP Replies and Errors 26
- Sample FTP Session 33

---

---

---

**CHAPTER 4** *Special Features* 37

Trash Cans 37

**APPENDIX A** *Glossary* 1

**APPENDIX B** *FTP Replies* 1

**INDEX** 1

# CHAPTER 1 *GETTING STARTED*

## *INTRODUCTION*

### **Welcome to DiskXtender!**

The Storage Manager, DiskXtender, is system software that works in concert with your UNIX system. This software provides UNIX users with hierarchical file storage that is automatically managed, virtually unlimited in capacity, and transparent to the user. The DiskXtender file system appears to the user just like a standard UNIX file system, and DiskXtender uses the same commands for FTP and NFS access that are used in standard UNIX systems.

### **UNIX and DiskXtender**

UNIX systems use a variety of storage media to provide users with a mix of short-term and long-term storage (see Figure 1 on page 2). For short-term file storage, files are maintained on magnetic disks. For backup and long-term file storage, files are generally stored on tapes. User files on UNIX systems must be regularly backed up from the magnetic disk (usually by a system administrator) in case of disk failure.

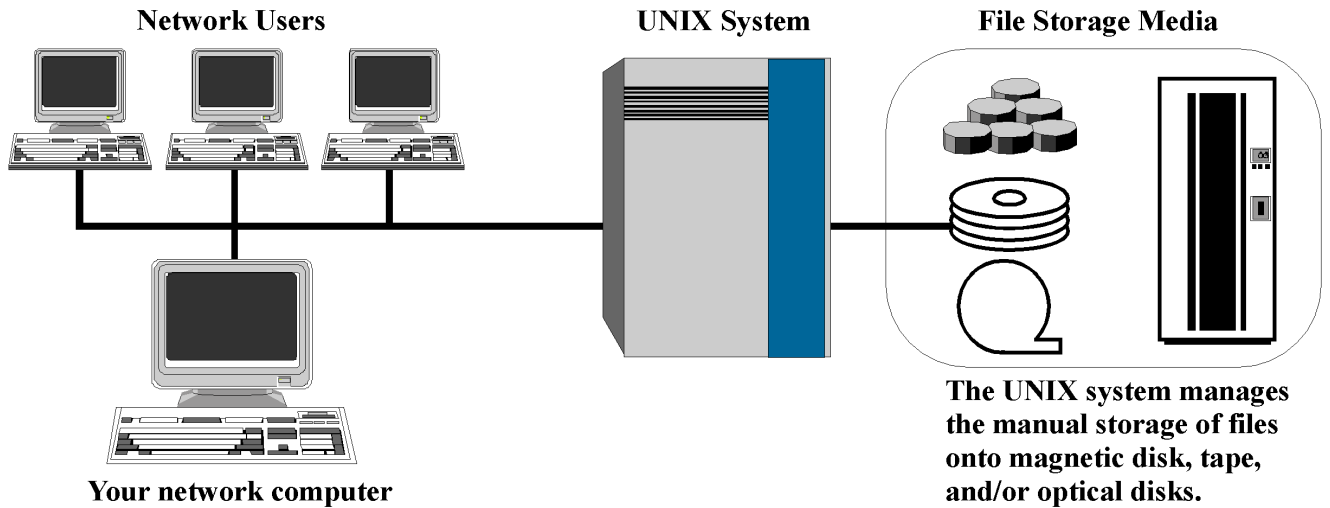
In addition, adequate free space for routine operations must be maintained on the magnetic disks. When the magnetic disks are full, users are responsible for moving their least-used files from the magnetic disk to tape media. Once these files have been moved off the disk, they no longer appear in the user's directory and must be separately cataloged.

This creates a storage management problem for UNIX users. As time passes and users work with larger numbers of files, it becomes increasingly difficult to keep track of them. As the level of difficulty rises, productivity begins to suffer.

DiskXtender overcomes these UNIX limitations by automatically managing files in a storage media hierarchy. DiskXtender assumes the responsibility for moving

files off the magnetic disk and assists system administrators by automating file backups.

To achieve this, DiskXtender creates a link to all files, including those stored on off-line tapes. With these links, files stored on tape still appear in the user's directory and can be easily copied back to the magnetic disk when needed. In this way, DiskXtender provides virtually unlimited storage space, and lets users transparently access all files, regardless of their physical location.



**FIGURE 1. Typical UNIX system configuration.**

---



## *DISKXTENDER FEATURES*

- Appears to users like a standard UNIX file system
- Automatic migration, caching and purging
- Access through industry-standard NFS and FTP interfaces
- Unlimited file size and filename length
- Protection from accidental file deletion through the use of trash cans

## *HOW DISKXTENDER WORKS*

### **NFS**

To users who access DiskXtender through the NFS protocol, the DiskXtender file system is a transparent extension of the UNIX file system. Users interact with their files by using standard UNIX commands and processes. See the chapter titled “NFS” starting on page 7 for more information.

### **FTP**

Users who access DiskXtender through the FTP protocol will find the standard FTP interface accompanied by an extended set of FTP commands and capabilities. Through FTP, the user can put and get files, set file permissions with special commands, make symbolic links and cache files. See the chapter starting on page 13 for more information about FTP.

### **File Migration, Purging and Caching**

While maintaining the appearance of a UNIX file system, DiskXtender continuously manages user files in the storage hierarchy. DiskXtender uses magnetic disks attached to the UNIX system as a disk cache. This means storage on magnetic disks is reserved for files which have been accessed most recently. Files are regularly moved within the storage hierarchy through the DiskXtender processes of migration, purging, and caching.

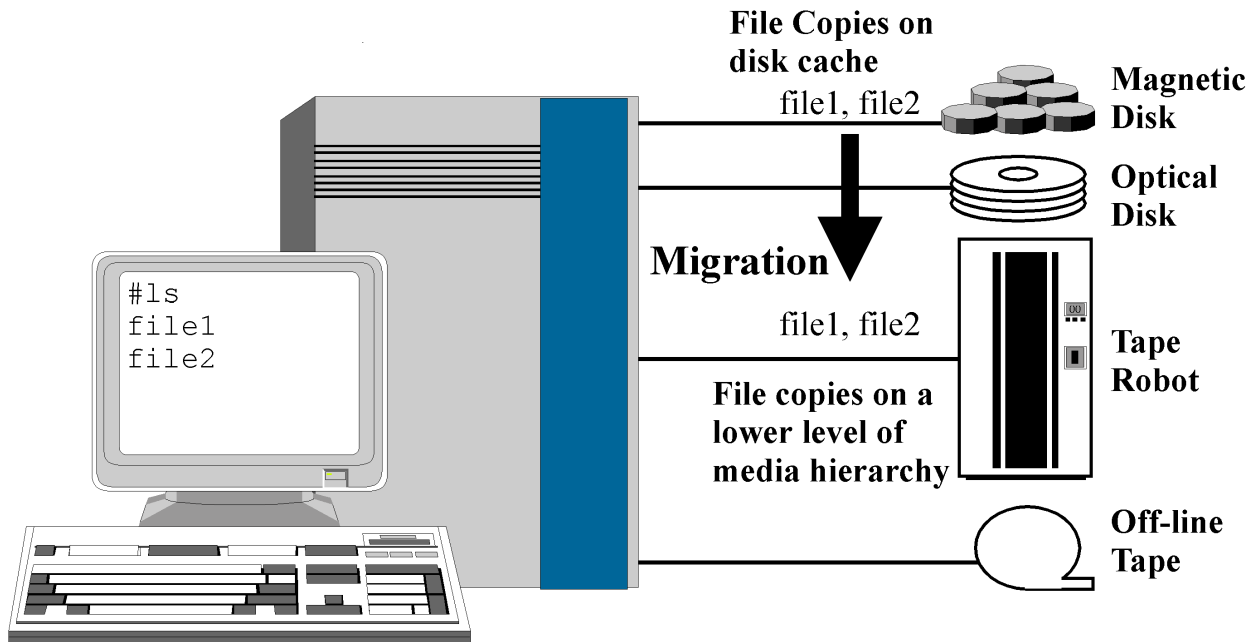


FIGURE 2. DiskXtender File Migration.

Migration is the copying of files from the disk cache (magnetic disk) down the storage media hierarchy onto another media level (see Figure 2 on page 4). At regular time intervals, or when disk cache space use exceeds preset limits, all files that have not been previously migrated are migrated. After migration, copies of migrated files exist on both the disk cache and the lower level of the hierarchy.

Purging occurs when files on the disk cache exceed a preset percentage of the disk cache space, called the “high water mark”. Files which have been migrated to a lower level in the hierarchy are eligible for purging from the disk cache. Purging of files from the disk cache continues until a preset percentage of free space, called the “low water mark”, has been created on the disk cache.

DiskXtender maintains a link to all files and shows them in the user's directory, regardless of the level of storage hierarchy on which they are located. When a user requests access to a file that has been purged from the disk cache, DiskXtender caches, or copies, the file directly back to the disk cache from the level to which the file was previously migrated (see Figure 3 on page 5). Users may notice some delay during the caching process, depending on the type of

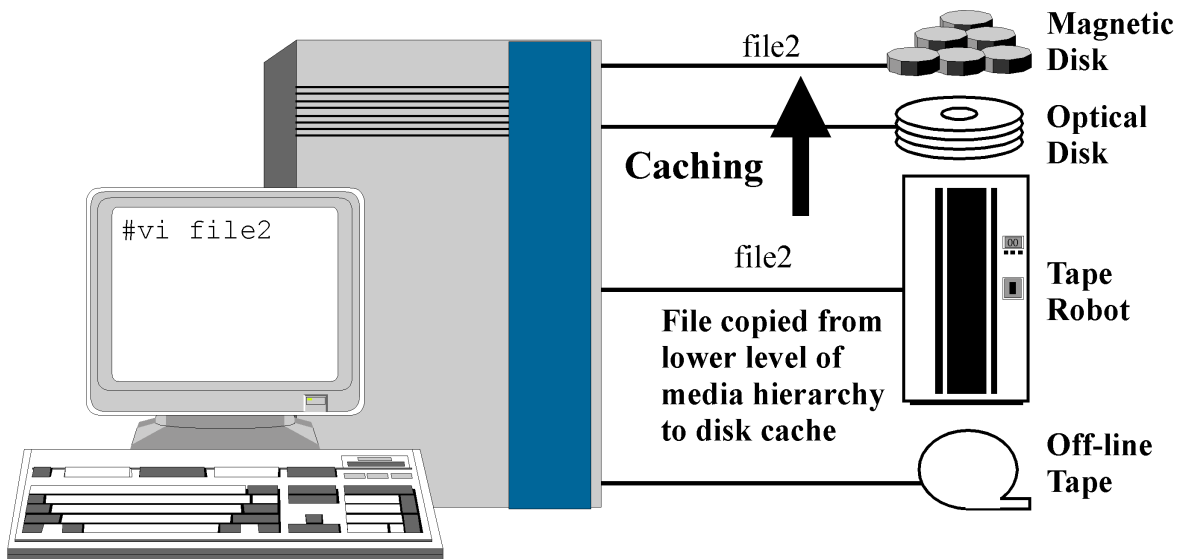


FIGURE 3. DiskXtender File Caching.

storage media on which the file is located. Once the file is available on the disk cache, the user regains full access.

## ACCESSING DISKXTENDER

To access DiskXtender, follow this simple procedure:

1. Login to your local computer. If you need assistance with the computer login procedure, see your system user manual or consult your system administrator.
2. After you have successfully logged into your local system, you may access DiskXtender on a remote file server by using FTP or NFS protocols, as explained in the following chapters.



## CHAPTER 2 NFS

NFS, the Network File System, is an industry standard that provides interconnection of file systems between independent computers. NFS gives your local machine the ability to access DiskXtender files and directories from a remote machine, and makes DiskXtender appear to be part of the local file system (see Figure 4 on page 7). Unlike FTP, NFS provides standard UNIX commands and is transparent to the user, except for potential time delays when replying to requests.

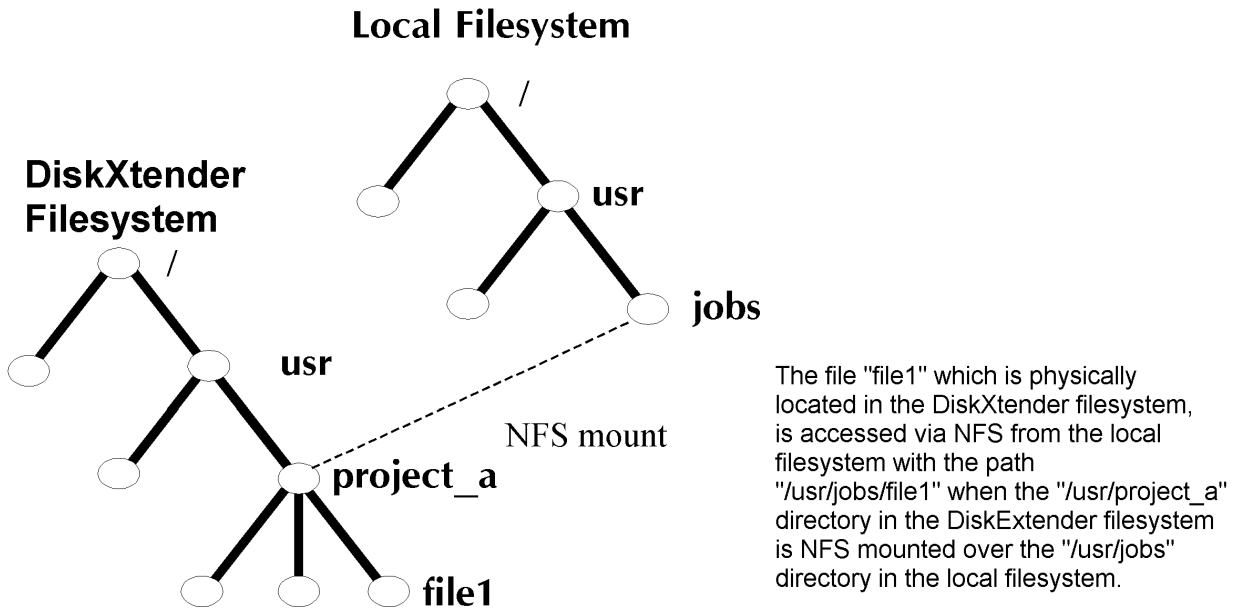


FIGURE 4. NFS Access.

## *GETTING STARTED*

Before you can use NFS, you must first mount an appropriate remote file system. The **mount** command attaches a remote file system to the local file system so that it can be accessed transparently. It can usually be performed only by the superuser (typically the system administrator).

To display a list of all file systems currently mounted on your local machine, type **mount** at the UNIX prompt. If the file system you want to access has been mounted, you may enter UNIX commands to access files and directories on that system.

### **NFS Commands**

There are no commands unique to NFS. NFS-mounted file systems function in the same manner as a standard UNIX file system. You may copy files to and from the remote file system by simply accessing the appropriate directory.

### **DiskXtender NFS Commands**

Since remote and local files look the same, it is often difficult to tell the difference between files stored on tape and those stored on disk. The primary difference between accessing files on the local file system and accessing DiskXtender via NFS is the speed at which files are retrieved. For example, when you access a file stored on tape, there will be a noticeable increase in access time when compared to the time required to access a file stored on disk.

If the process does not complete within a reasonable time, you may attempt to stop it by using your interrupt key (usually **CONTROL-C**). The file will continue to be cached from tape to disk on the remote machine. Once the file has been cached, the attempted command may be executed again.

You may only interrupt the process if the DiskXtender file system was mounted with version 3 or version 2 as interruptible. If entry of the interrupt command does not effect an interrupt of the process, check with your system administrator for further information.

You may determine if a file exists on disk, and thus available for immediate retrieval, or if the file exists only on the tape by examining the “sticky” bit in the

permission field of a file. The “sticky” bit will be set for all files which exist only on tape. To examine the “sticky” bit, perform a long directory listing with the **ls** command.

For example:

```
% ls -l
-rw-rw-r-T 1 ul user 4165 Feb 23 21:16 file1
```

The use of the “sticky” bit to convey file location with NFS is configurable by the system administrator. If you need to use the “sticky” bit for its intended purpose, or if the “sticky” bit is not behaving as described above, talk with your system administrator.

## **Common NFS Replies and Errors**

NFS uses the same replies and error messages that are displayed by the UNIX file system, with two exceptions. When the link between the local machine and the remote server goes down or when you request access to a file which only resided on tape, you may see the message when the file system is mounted using NFS Version 2:

```
NFS server not responding, will try again
```

When the file system is mounted using NFS Version 3, the following message may be shown.

```
File temporarily unavailable on the server, retrying...
```

The computer will continue to attempt to access the remote server until it is successful. Use the interrupt key if you do not want to wait. In addition, when the file system is mounted using NFS Version 2, commands that reference a file being cached may queue behind the original request that cached the file, causing these commands to appear to hang.

## **Sample NFS Session**

NFS is transparent and uses standard UNIX commands. In the following sample NFS session, */usr/local/bob* corresponds to your local directory, and

*/usr/remote/bob* corresponds to a remote directory that has been mounted on the local directory.

In this section, user entries are denoted by bold type.

```
% mount  
remote: /usr/remote/bob on /usr/mnt/bob nfs (rw)
```

The user gets a list of all file systems mounted and the access permissions for each mounted system. In this example, */usr/remote/bob* on the remote machine is mounted read-write on the local directory */usr/mnt/bob*.

```
% pwd  
/usr/local/bob  
% cd /usr/mnt/bob  
% pwd  
/usr/mnt/bob  
% ls -a  
.  
..  
.trash  
file1  
file2
```

The user is initially in the local directory */usr/local/bob* and then changes to the remote directory through the local mount point */usr/mnt/bob*.

```
% ls -l  
drw----- 2 u2 user 1024 Feb 23 20:00 .trash  
-rw-----T 1 u1 user 4165 Feb 23 21:16 file1  
-rw----- 1 u1 user 2458 Feb 23 21:16 file2
```

The user utilizes the **ls -l** command to view the entries and entry attributes of the current remote directory.

An upper-case “T” at the end of the permissions indicates that the file is on archive media and that the execute bit in the permissions has NOT been set. A



lower-case “t” indicates that the file is on archive media and that the execute bit in the permissions has been set. See the man page for **ls** on your system for a detailed explanation of permission bits and their meaning.

```
% cp /usr/local/bob/file3 file3
```

```
% ls -a
```

```
.
```

```
..
```

```
.trash
```

```
file1
```

```
file2
```

```
file3
```

The user copies the local file *file3* to the remote directory. Since the user is currently in the remote directory, it is not necessary to provide a full pathname for the remote directory. The user checks to see if the file is actually copied into the local directory by listing that directory.

```
% cp file1 /usr/local/bob/newfile
```

```
% cd /usr/local/bob
```

```
% ls
```

```
file3
```

```
file4
```

```
newfile
```

The user copies the remote file *file1* to the local directory */usr/local/bob* and renames the file *newfile*. The user checks to see if the file is actually copied into the local directory by listing that directory.

For more information on NFS, use the **man** utility on your local machine.

```
% man nfs
```

or

```
% man -k nfs
```



## CHAPTER 3 *FTP*

### *ABOUT FTP*

FTP, File Transfer Protocol, is an industry-standard protocol used to transfer files to and from a remote computer system. Both UNIX and non-UNIX local computer systems can access DiskXtender via FTP.

### *LOGGING IN*

You must first log in to your local computer system. At the prompt, type **ftp** and the name or IP address of the remote system you wish to access, and press ENTER (the default port is 21; access via another port will require entry of the port number):

```
% ftp remote_machine [port]
```

Your computer will attempt to access the remote computer and a message will be displayed, indicating if you have successfully connected with the remote FTP server:

```
Connected to remote_machine  
220 UNIX Archive FTP server ready
```

Once connected to the remote computer, you must log into DiskXtender in the same manner you used to log into your local computer system. You will be asked to enter a login name and a password:

```
Name: bob  
Password: XXXXXXXX
```

Your password will not be echoed to the screen.

You cannot access a remote computer with FTP if you have not been given a remote login name and password. You must contact the system administrator of the remote site in order to get access to the remote FTP server.

You will receive a reply stating if you have successfully logged in or if you need to try again. See “Common FTP Replies and Errors” in Appendix B.

When you have successfully logged in, the following FTP prompt displays on the screen:

```
ftp>
```

You may now begin to enter FTP commands.

## *UNIX FTP COMMANDS*

This section lists frequently-used UNIX FTP commands, their definitions, usage and examples. On many systems, you may enter the underlined portion of the command as a shortcut. The examples use the shortcut version of the commands. Commands may only be entered at the FTP prompt. For additional standard FTP commands, see the UNIX man page on FTP.

- append

Appends a local file to a file on the remote machine. If the remote file is left unspecified, the local file name is used.

Usage: `append local-file remote-file`

**Example:** `ap /usr/bob/newfile /users/comp/bob/file1`

- ascii

Sets the file transfer type to ASCII. This is commonly the default, but is not desirable unless transferring ASCII files between machines which have different character sets.

Usage: `ascii`

**Example:** `as`

Binary files should never be transferred while in ASCII mode, otherwise they will be transferred incorrectly. ASCII mode performs character translations of certain characters. These characters occur randomly in a binary file and their translation

may result in the corruption of the binary file. As a rule it is best to always use binary mode.

- binary  
Sets the file transfer type to support binary image transfer. Binary file transfer is faster than ASCII file transfer.  
Usage: binary  
**Example: bi**
- bye  
Terminates the FTP session with the remote server and exits FTP.  
Usage: bye  
**Example: by**
- cd  
Changes the current directory on the remote machine to the directory specified.  
Usage: cd remote-directory  
**Example: cd /users/comp/bob**
- cdup  
Changes the current remote machine directory to its parent directory.  
Usage: cdup  
**Example: cdu**
- close  
Terminates the FTP session with the remote server and returns to the command interpreter (the FTP prompt). Once you close the connection, you may use the **open** command to connect to another remote server.  
Usage: close  
**Example: cl**
- delete  
Deletes the specified remote file from a remote directory.  
Usage: delete remote-file  
**Example 1:del file3**  
**Example 2:del /users/comp/bob/dir1/file1**

- **dir**  
Lists the entries and entry attributes of a directory for the remote machine. If the remote directory is not specified, the current remote directory will be listed.

Usage: dir [remote-directory]

**Example: dir /users/comp/bob**

The DiskXtender **dir** command in FTP provides two extra columns. One column describes the highest layer media on which the file is stored. The other additional column displays what “family” the file belongs to.

- **get**  
Copies a remote file from a remote directory to the local machine. If no pathnames are specified, DiskXtender will copy the remote-file from the current remote directory to the current local machine directory.

Usage: get remote-file [local-file]

**Example 1:ge /users/comp/bob/file1 /usr/bob/newfile**

*/users/comp/bob/file1* is copied from the remote machine to */usr/bob/newfile* on the local machine.

**Example 2:ge file1 newfile**

*file1* is copied from the current remote directory to *newfile* in the current local machine directory.

**Example 3:ge file1 /usr/bob/newfile**

*file1* is copied from the current remote directory to */usr/bob/newfile* on the local machine.

**Example 4:ge /users/comp/bob/file1**

*/users/comp/bob/file1* is copied from the remote machine to */users/comp/bob/file1* on the local machine if the directory */users/comp/bob* exists. If */users/comp/bob* does not exist on the local machine, the command will return an error.

- **hash**  
Toggles the printing of the pound-sign (#) on the screen for each data block transferred. The size of the data block is machine dependent.

Usage: hash

**Example: ha**

- **help**  
Displays information to help you learn more about FTP commands. There are two ways to access **help**. If you type the command **help** by itself, a list of valid FTP commands will display. To get help on a specific command, type **help** followed by the command name.  
Usage 1:help  
**Example 1:he**  
Usage 2:help command-name  
**Example 2:he get**
- **lcd**  
Changes the current directory on the local machine. If no directory is specified, FTP changes to your local home directory.  
Usage: lcd local-directory  
**Example: lcd /usr/bob**
- **ls**  
Lists filenames in a remote directory. If the remote directory is not specified, FTP lists your current remote directory.  
Usage: ls [remote-directory]  
**Example: ls /users/comp/bob**
- **mdelete**  
Deletes multiple files. This command deletes the specified files. If full pathnames are not specified, the default is the current remote directory. Wildcard characters (\*, ?, [ ]) may be used with **mdelete**.  
Usage: mdelete remote-file1 remote-file2 ...  
**Example 1:mde file1 file2**  
*file1* and *file2* are deleted from the current remote directory.  
**Example 2:mde /users/comp/bob/file1 /users/comp/barb/file1**  
*/users/comp/bob/file1* and */users/comp/barb/file1* are deleted.
- **mget**  
Gets multiple files. This command lets you retrieve specified files from one or more remote directories and transfer them to the current local directory. Wildcard characters (\*, ?, [ ]) may be used with **mget**.

Usage: mget remote-file1 remote-file2 ...

**Example 1:** mge file1 file2

**Example 2:** mget /users/comp/bob/file1 /users/comp/barb/file1

- mkdir

Makes a directory on the remote machine. If no pathname is specified, the directory will be inserted into the current remote directory.

Usage: mkdir remote-directory

**Example:** mk /users/comp/bob/dir2

- mput

Puts multiple files. Transfers multiple local files from one or more local directories to the current remote directory. Wildcard characters (\*, ?, [ ]) may be used with **mput**.

Usage: mput local-file1 local-file2 local-file3 ...

**Example 1:** mp file1 file2 file3

**Example 2:** mp /usr/bob/file1 /usr/barb/file1

- open

Establishes a connection to the specified host (remote) FTP server. If an optional port number is specified, ftp will attempt to contact an FTP server at that port.

Usage: open host [port]

**Example 1:** op remote\_machine

**Example 2:** op remote\_machine 1021

- prompt

Toggles interactive prompting. This occurs during multiple file transfers to allow the user to retrieve and store files selectively. If prompting is turned off, an **mget** or **mput** command transfers all files specified. The default prompt value is on.

Usage: prompt

**Example:** prom

- put

Copies a local file from a local directory to the remote machine. If no pathnames are specified, DiskXtender copies the local-file from the current local directory to the current remote machine directory.



Usage: put local-file [remote-file]

**Example 1:pu /usr/bob/file1 /users/comp/bob/newfile**

*/usr/bob/file1* is copied from the local machine to */users/comp/bob/newfile* on the remote machine.

**Example 2:pu file1 newfile**

file1 is copied from the current local directory to newfile in the current remote machine directory.

**Example 3:pu file1 /users/comp/bob/newfile**

file1 is copied from the current local directory to */users/comp/bob/newfile* on the remote machine.

**Example 4:pu /usr/bob/file1**

*/usr/bob/file1* is copied from the local machine to */usr/bob/file1* on the remote machine if the directory */usr/bob* exists on the remote machine. If */usr/bob* does not exist on the remote machine, the command returns an error.

- pwd

Prints the name of the current remote directory.

Usage: pwd

**Example: pw**

- quit

Terminates the FTP session with the remote server and exits FTP. Same command as **bye**.

Usage: quit

**Example: qui**

- quote

Specified arguments or commands are sent verbatim to the remote FTP server. See page 21 for a list of commands that are used with the **quote**.

Usage: quote arg1 arg2...

**Example: quo stage 60 /users/comp/bob/newfile**

- recv

A synonym for **get**.

Usage: recv remote-file [local-file]

**Example: See get.**

- rename  
Renames the file from-name on the remote machine to the file to-name.  
Usage: rename from-name to-name  
**Example: ren /users/comp/bob/newfile /users/comp/bob/oldfile**
- reset  
Clears the reply queue. This command resynchronizes command/reply sequencing with the remote FTP server.  
Usage: reset  
**Example: rese**
- rmdir  
Deletes a directory on the remote machine. If no pathname is specified, it deletes the directory from the current remote directory.  
Usage: rmdir remote-directory  
**Example: rmd /users/comp/bob**
- send  
A synonym for **put**.  
Usage: send local-file [remote-file]  
**Example: See put.**
- status  
Displays the current status of ftp.  
Usage: status  
**Example: sta**
- user  
Identifies you to the remote FTP server as the same or a different user. If you enter only the login-name, you will be prompted for a password and/or account if required. You may enter your login-name and password at the time the command is initially entered.  
Usage: user login-name [password]  
**Example 1:user bob**  
User bob will be prompted to enter a password.

**Example 2:us bob XXXX**

User bob has entered his login-name and his password.

*DISKXTENDER FTP COMMANDS*

DiskXtender expands on FTP features through the use of the **quote** and **site** commands. These commands should not be abbreviated. The **site** command may be left off on most systems and the command is case insensitive on most systems. The additional commands are:

- **help**

The **help** command displays information about the commands on the remote FTP server. There are two ways to access remote **help**. If you type the command **quote help** by itself, a list of valid FTP commands on the remote FTP server will be returned. To get help on a specific remote command, type **quote help** followed by the command name.

Usage 1:help

**Example 1:help**

Usage 2:help command-name

**Example 2:quote help STAGE**

- **allo**

To specify the size of the next file that will be put, use the **allo** command. This command improves the efficiency of binary puts and is used automatically by some ftp clients. The size is in bytes and only applies for the next put.

Usage: quote allo filesize

**Example: quote allo 58732883**

- **chgrp**

To change the group associated with a local file, UNIX uses the **chgrp** command. To change the group associated with a remote file, use the **quote chgrp** command at the FTP prompt.

Usage: quote chgrp group-name filename

**Example: quote chgrp group1 file1**

You must be the owner of the file and a member of the new group (or the superuser) to execute this command.

- **chmod**

To change the permissions on a local file, UNIX offers the **chmod** command. To change the permissions on a remote file, use the **quote chmod** command at the FTP prompt.

Usage: quote chmod permissions filename

**Example: quote chmod 700 /users/comp/bob/file1**

You must be the owner of the file or the superuser to execute this command. The permissions must be in octal notation. Symbolic chmod parameters are not supported with the FTP chmod command. For more information on octal permission notation, see your system man page on **chmod**.

- **chown**

To change the owner of a local file, UNIX offers the **chown** command. To change the owner of a remote file, use the **quote chown** command at the FTP prompt.

Usage: quote chown owner-name filename

**Example: quote chown bob file1**

You must be the superuser to execute this command.

- **direct**

To get a file directly from the storage media, bypassing the disk cache, use the **direct** command. This command is a toggle, with a default value of off. This command is useful if the file being retrieved will not be read again and it is relatively large. To enable direct mode, use the **quote direct** command at the FTP prompt.

Usage: quote direct

**Example: quote direct**

- **GTRSH**

The **GTRSH** (get trash) command displays the current trash can timeout interval. Trash cans are discussed on page 37.

Usage: quote site GTRSH

**Example: quote site GTRSH**

The timeout interval is displayed in minutes.

- **ln**

The **ln** command allows you to create a symbolic link.

Usage: `ln file1 file2`

`file1` is the existing file or directory to be linked and `file2` is the symbolic link.

**Example:** `ln /usr/DiskXtender /usr/bob/UT`

- **NMDUP**

This DiskXtender FTP command lets you see and set the current number of duplicate copies of a file that will be stored on DiskXtender. When this command is executed, every file you put to DiskXtender is duplicated *n* number of times on *n* different tapes. This is a temporary setting. When you end the FTP session, the number of duplicate copies will return to the default value.

Usage: `site NMDUP [n]`

**Example 1:** `site NMDUP`

This command returns the current value of the number of duplicate copies for which the user is set.

**Example 2:** `site NMDUP 5`

The number of duplicate copies is set to 5 for this user.

The maximum number of copies is restricted to 15 or less, as set by your system administrator.

- **PURGE**

The **purge** command lets you indicate to the DiskXtender system that a file may be purged from the disk cache when space is needed. This may be used when it is known that a file will not be accessed in the near future and helps the DiskXtender system keep the files most likely to be accessed on the disk cache.

Usage: `purge file`

**Example:** `purge /users/comp/bob/file1`

- **SETFAM**

This DiskXtender FTP command lets you see and set the family to which files will belong. Each file in a family resides on a tape with files exclusively of the same family. For the duration of the FTP session, or until **SETFAM** is re-executed, all created files will be assigned to family-name or family-number.

If you are not identified in the DiskXtender family configuration file as a member of the designated family, or if you use a family-name or family-number that is not defined in the family configuration file, the files are assigned to the common family. The common family, family 0, is the default family.

Usage: quote site SETFAM [family-name]

**Example 1: quote site SETFAM smith**

Usage: quote site SETFAM [family-number]

**Example 2: quote site SETFAM 1**

**Example 3: quote site SETFAM**

- STAGE

It may take additional time for DiskXtender to retrieve a file stored on tape. If you use the **get** command, you may see a message that the file is being retrieved from the archive. Because the FTP client does not wait until the file is cached from tape, unless you previously executed the **wait** command, you must execute a second **get** command to obtain the file once it has been cached to disk.

To avoid the necessity of issuing two **get** commands or waiting for the file to be cached, use the **quote STAGE** command before the file is actually needed. Once the file has been cached (staged) to disk, it may be acquired with a single **get** command.

Use the **dir** command to see whether the file is currently stored on disk (DK) or ARCHIVE (AR) to determine if it needs to be staged. Files on disk are identified by “DK” while files on tape use “AR” as an identifier. See the sample **dir** command output below.

```
ftp> dir
200 PORT command successful.
150 Opening ASCII mode data connection for /diskxtender/production/bin/ddir (0 bytes).
-rw-----      1  u1  user   AR   common   4165    Feb 23 21:16    banner
-rw-----      1  u1  user   DK   common   2458    Feb 23 21:16    basename
-rw-----      1  u1  user   AR   common    721    Feb 23 21:16    batch
-rw-----      1  u1  user   DK   common  25620    Feb 23 21:16    bc
226 Transfer complete.
1015 bytes received in 6 seconds (0.17 Kbytes/s)
```

Usage: quote stage waittime file

*waittime* is the desired waiting time (locked by the transaction) in seconds. You are given control of the prompt when *waittime* has expired. If you do not want to be locked by the transaction, enter a zero value. Files will be staged to disk regardless of the value of *waittime*. The file parameter represents the name of the file you want to stage to disk.

**Example: quote stage 60 /users/comp/bob/file1**

- STRSH

You may change your trash can timeout interval using the **STRSH** (set trash) command. Once the timeout is changed, it remains at this value after the FTP session is terminated. The timeout is specified in minutes. Trash cans are discussed on page 37.

Usage: quote site STRSH time

**Example: quote site STRSH 60**

- SWAP

The **SWAP** command allows you to request that when a file is retrieved from the tape system that it be retrieved from one of the duplicate copies instead of the primary copy. This command is useful if a tape has become damaged and removed from the DiskXtender system. This command should only be run under the supervision of the system administrator as some sites remove their duplicate copy tape for off-site storage.

Usage: quote swap file

**Example: quo swap /users/comp/bob/file1**

- umask

This command is used to set the file creation mask for default permissions on new files. The default file creation mask is set to 077 at start-up, for owner access but no group or world access.

Usage: quote umask permission-mask

**Example: quote umask 007**

- WAIT

The **WAIT** command lets you toggle waiting on and off when a file is cached from tape to disk, when using **get**, before returning the FTP prompt. The default is off.

Usage: quote WAIT

**Example: quo WAIT**

## *COMMON FTP REPLIES AND ERRORS*

When you enter an FTP command, you receive a corresponding reply that indicates that the command was accepted, rejected or is being processed. An FTP reply consists of a three-digit code followed by a brief description of the result. This section contains an overview of common FTP reply codes for the commands discussed in “UNIX FTP Commands” on page 14. For a complete listing of FTP replies in numeric order, See “FTP Replies” in APPENDIX B.

- **GTRSH**

Accepted: 250 Requested file action okay, request completed.

Rejected: 506 Usage: error.

550 Requested action not taken.

- **NMDUP**

Accepted: 200 Command okay.

- **STRSH**

Accepted: 250 Requested file action okay, request completed.

Rejected: 506 Usage: error.

550 Requested action not taken.



- **append**
  - Accepted: 125 Data connection already opened.
  - 150 File status okay.
  - 226 Closing data connection.
  - 250 Requested file action okay, request completed.
  - Rejected: 421 Service not available, closing control connection.
  - 500 Syntax error, command unrecognized.
  - 501 Syntax error in parameters or arguments.
  - 502 Command not implemented.
  - 530 Not logged in.
  
- **bye or quit**
  - Accepted: 221 System status.
  - Rejected: 500 Syntax error.
  
- **cd or cdup**
  - Accepted: 200 Command okay.
  - Rejected: 421 Service not available.
  - 500 Syntax error.
  - 501 Syntax error.
  - 502 Command not implemented.
  - 530 Not logged in.
  - 550 Requested action not taken.
  
- **chgrp**
  - Accepted: 250 Requested file action okay, request completed.
  - Rejected: 505 No such file or directory.
  
- **chmod**
  - Accepted: 250 Requested file action okay, request completed.
  - Rejected: 505 No such file or directory.

- chown
  - Accepted: 250 Requested file action okay, request completed.
  - Rejected: 505 No such file or directory.
  
- close
  - Accepted: 221 System status.
  - Rejected: 500 Syntax error.
  
- delete or mdelete
  - Accepted: 250 Requested file action okay, request completed.
  - Rejected: 421 Service not available.
  - 450 Requested action not taken.
  - 500 Syntax error.
  - 501 Syntax error.
  - 502 Command not implemented.
  - 530 Not logged in.
  - 550 Requested action not taken.
  
- dir
  - Accepted: 150 Opening data connection.
  - 200 Command okay.
  - 226 Transfer complete.
  - Rejected: 500 Syntax error.

- get, mget, or rec
  - Accepted: 125 Data connection already opened.
  - 150 File status okay.
  - Processing: 550 File being retrieved.
  - Rejected: 421 Service not available.
  - 450 Requested action not taken.
  - 500 Syntax error.
  - 501 Syntax error.
  - 530 Not logged in.
  - 550 Requested action not taken, file unavailable.
  
- ln
  - Accepted: 250 Requested file action okay, request completed.
  - Rejected: 505 No such file or directory.
  - 506 Usage: error.
  
- ls
  - Accepted: 125 Data connection already opened.
  - 150 File status okay.
  - Rejected: 421 Service not available.
  - 450 Requested file action not taken.
  - 500 Syntax error.
  - 501 Syntax error.
  - 502 Command not implemented.
  - 530 Not logged in.

- **mkdir**
  - Accepted: 257 "PATHNAME" created.
  - Rejected: 421 Service not available.
  - 500 Syntax error.
  - 501 Syntax error.
  - 502 Command not implemented.
  - 530 Not logged in.
  - 550 Requested action not taken.
- **open**
  - Accepted: 220 Service ready for new user.
  - 230 User logged in, proceed.
  - Rejected: 421 Service not available.
- **prompt**

No reply code; replies whether interactive mode is on or off.
- **put, mput, or send**
  - Accepted: 125 Data connection already opened.
  - 150 File status okay.
  - Rejected: 421 Service not available.
  - 450 Requested action not taken.
  - 500 Syntax error.
  - 501 Syntax error.
  - 530 Not logged in.
  - 553 Requested action not taken.
- **pwd**
  - Accepted: 257 pathname is current directory.
  - Rejected: 421 Service not available.
  - 500 Syntax error.
  - 501 Syntax error.
  - 502 Command not implemented.
  - 550 Requested action not taken.

- rename

Accepted: 250 Requested file action okay, completed.

Rejected: 421 Service not available.

450 Requested file action not taken.

500 Syntax error.

501 Syntax error.

502 Command not implemented.

530 Not logged in.

532 Requested file action aborted.

533 Request action not taken.

- reset

No reply code if accepted.

Processing: 350 Requested file action pending further information.

Rejected: 500 Syntax error.

501 Syntax error.

502 Command not implemented.

530 Not logged in.

- rmdir

Accepted: 250 Requested file action okay, completed.

Rejected: 421 Service not available.

500 Syntax error.

501 Syntax error.

502 Command not implemented.

530 Not logged in.

550 Requested action not taken.

- stage
  - Accepted: 250 DiskXtender STAGE command successful.
  - Processing: 505 File is being moved from the archive.  
550 File being retrieved.
  - Rejected: 450 Requested action not taken.  
500 Syntax error.  
501 Syntax error.  
550 Requested action not taken, file unavailable.
  
- status
  - Accepted: 211 System status reply.  
212 Directory status.  
213 File status.
  - Rejected: 421 Service not available.  
450 Requested action not taken.  
500 Syntax error.  
501 Syntax error.  
502 Command not implemented.
  
- umask
  - Accepted: 200 Command okay.  
250 Requested file action okay, request completed.
  
- user
  - Accepted: 230 User logged in, proceed.
  - Processing: 331 User name okay, need password.  
332 Need account for login.
  - Rejected: 421 Service not available.  
500 Syntax error.  
501 Syntax error.  
530 Not logged in.

- wait  
Accepted: 258 wait on (or off).

### *SAMPLE FTP SESSION*

This section contains a sample FTP session using a subset of common FTP commands.

In this section, user entries are denoted by bold type.

```
% ftp remote_machine
Connected to remote_machine
200-
UNIX Archive FTP Server(DiskXtender Version(2.3)) active.
200 UNIX Archive FTP server ready.
NAME: bob
331 Password required for bob
PASSWORD: XXXXXX
230 bob logged in
```

User *bob* establishes connection with the remote server by using FTP and logging in.

```
ftp> pwd
257/users/comp/bob is current directory
ftp> ls
200 PORT command successful
150 Opening ASCII mode data connection
.
..
.trash
file1
```

```
file2
file3
226 Transfer complete
ftp> bin
200 Type set to I
```

User *bob* asks for the current remote directory which is */users/comp/bob*. Then he lists the contents of the directory. Last, he turns on the binary file transfer mode.

```
ftp> mput file4 file5
mput file4? y
200 PORT command successful
150 Opening BINARY mode data connection for file4
226 Transfer complete
local: file4 remote: file4
13883 bytes sent in 0.09 seconds
mput file5? y
200 PORT command successful
150 Opening BINARY mode data connection for file5
226 Transfer complete
local: file5 remote: file5
9670 bytes sent in 0.074 seconds
```

User *bob* utilizes the **mput** command. This command copies local files into the current remote directory. Because the prompt option is on by default, user *bob* is



asked to verify the file transfer before each file is transmitted. A status message is sent back for each request.

```
ftp> dir
200 PORT command successful.
150 Opening ASCII mode data connection for /diskxtender/production/bin/ddir (0 bytes).
-rw-----      2   u1   user   DK   common   1024   Feb 10 10:76   .trash
-rw-----      1   u1   user   AR   common   2458   Feb 20 15:16   file1
-rw-----      1   u1   user   DK   common   721    Feb 20 15:16   file2
-rw-----      1   u1   user   AR   common  25620   Feb 20 15:16   file3
-rw-----      1   u1   user   DK   common   8445   Feb 23 16:11   file4
-rw-----      1   u1   user   DK   common  28708   Feb 23 16:12   file5
226 Transfer complete.
1015 bytes received in 6 seconds (0.17 Kbytes/s)
```

User *bob* utilizes the FTP **dir** command to view the entries and entry attributes of the current remote directory.

The **dir** listing contains an extra column for the media in the highest layer of the hierarchy on which each file is stored (see column 5). Note that *file1* and *file3* do not exist on the disk cache.

```
ftp> get file2 newfile
200 PORT command successful
150 Opening BINARY mode data connection for file2
226 Transfer complete
local: newfile remote: file2
1964 bytes received in .09 seconds
ftp>quit
221 Goodbye
```

User *bob* copies the remote file *file2* from the current remote directory to the local machine and renames the file *newfile*. After the transfer is successfully completed, *bob* ends the FTP session.

For more information on FTP, use the **man** utility on your local machine.

**% man ftp**

## CHAPTER 4 *SPECIAL FEATURES*

### *TRASH CANS*

The trash can is a DiskXtender safety feature that enables you to easily recover files deleted from DiskXtender directories.

As a DiskXtender user, you have a directory named *.trash* in your home directory. When you delete a file from your DiskXtender directory structure, the file is removed from that directory and moved into your *.trash* directory.

When this occurs, DiskXtender renames the files to avoid name conflicts. Two character strings are added as extensions to each file. The first is the date the file was moved into *.trash* and the second is a global counter that ensures that multiple files with the same name that are deleted simultaneously from different directories are distinguishable.

For example, the *.trash* filename given to a file originally named *file1* would appear as:

```
file1#04-01-96#01:09:30PST#0064
```

Unlike UNIX file names, DiskXtender filenames are not limited in length.

A file will remain in *.trash* until a predefined timeout period elapses. The initial timeout period is set by the system administrator. When the timeout period is reached, the file is automatically deleted from the *.trash* file. This delete is final.

There are two methods, described below, for retrieving files from *.trash*.

1. Use the FTP **rename** or NFS **mv** command to move the file out of the trash can and into another directory, possibly giving the file a different name in the process.

2. Use FTP or NFS commands to retrieve a file from the trash can before the timeout period expires, thus recovering from file deletes. You may also list the contents of the trash can directory and delete entries from the trash can. Any delete from the *.trash* directory is final. See the following table for trash can operations.

Operation	FTP Commands	NFS Commands
List the contents of <i>.trash</i>	dir	ls or ls -la
Retrieve files from <i>.trash</i>	get	cp or mv
Delete entries in <i>.trash</i>	del	rm

**TABLE 1. Trash can operations**

The following commands cannot be performed on the *.trash* directory.

- You cannot delete the *.trash* directory itself.
- You cannot move or rename the *.trash* directory.
- You cannot rename entries in *.trash*, since DiskXtender uses the date (first ASCII string extension) to determine when the timeout period has expired. You may, however, **rename** or **mv** entries out of the trash can to another directory.
- You cannot use the **put** (FTP), **mv** or **cp** (NFS) commands to place files directly into the *.trash* directory.

## APPENDIX A *GLOSSARY*

<b>ASCII</b>	The ASCII character set, which encodes numeric, alphabetic, and special characters in binary format. Also a mode for moving files with FTP.
<b>binary</b>	Machine-readable data. Also a mode for moving files with FTP.
<b>caching</b>	The act of automatically copying a file from a lower layer in the hierarchy to a higher layer. Caching only occurs after migration and purging have occurred. When a file is cached, there are at least two copies of the file. One copy remains on the layer from which it was cached. The other copy is on the layer to which it was cached. Caching can be initiated through FTP by use of the <b>quote stage</b> commands or by trying to read data from a file with NFS.
<b>directory</b>	A cataloging structure consisting of human-oriented names, uniquely identified by a pathname.
<b>file</b>	An ordered set of computer data of arbitrary length that is uniquely identified by a pathname.
<b>fileserver</b>	A process that creates and controls access to files.
<b>login</b>	The process of granting a user access to a computer.
<b>migration</b>	The process of copying a file from a higher layer to a lower layer in the DISKXTENDER hierarchy, for example, from disk to tape.
<b>mount</b>	A command that attaches or maps a remote file system into the local file system so that it can be accessed transparently. When the <b>mount</b> command is entered without parameters, it displays all mounted remote file systems.
<b>off-line</b>	A storage medium that requires human intervention for access.
<b>on-line</b>	A storage medium (typically magnetic disk or robotic tape) that does not require human intervention for access.
<b>pathname</b>	The character string used to identify a file or directory
<b>purging</b>	The act of removing files from a higher layer in the hierarchy at some time after they have been migrated to a lower layer. Purging is an automated system deletion activated by parameters configurable by the system administrator, and is used to satisfy the need for additional storage space on the higher layer.
<b>reply</b>	An acknowledgment, positive or negative, sent from the file system to the user in response to commands sent by the user.
<b>retrieve</b>	A request to obtain a file.

---

## Glossary

---

<b>trash can</b>	DISKXTENDER mechanism for allowing retrieval of files that are mistakenly or accidentally deleted by the user from their directory structure.
<b>user</b>	Person or process that accesses a computer system.

## APPENDIX B *FTP REPLIES*

<b>Numeric Response</b>	<b>ASCII response</b>
120	Service ready in nnn minutes.
125	Data connection already open, transfer starting.
150	File status okay, about to open data connection.
200	Command okay.
202	Command not implemented, superfluous at this site.
211	System status, or system help reply.
212	Directory status.
213	File status.
214	Help message.
220	Service ready for new user.
221	Service closing control connection.
225	Data connection already open, no transfer in progress.
226	Closing data connection.
230	User logged in, proceed.
250	Requested file action okay, completed.
257	Pathname is current directory.
258	Command on (or off).
331	User name okay, need password.
332	Need account for login.
350	Requested file action pending further information.
421	Service not available, closing control connection.
425	Can't open data connection.
426	Connection closed, transfer aborted.
450	Requested file action not taken.
451	Requested action aborted. Local error in processing.
452	Requested action not taken. Insufficient storage space.

500	Syntax error, command unrecognized.
501	Syntax error in parameters or arguments.
502	Command not implemented.
503	Bad sequence of commands.
504	Command not implemented for that parameter
505	No such file or directory. File is being moved from the archive.
506	Usage: error.
507	STAGE: invalid waittime.
530	Not logged in.
532	Need account for storing files.
533	Requested action not taken.
550	Requested action not taken. File unavailable. File being retrieved.
551	Requested action aborted. Page type unknown.
552	Requested file action aborted. Exceeded storage allocation.
553	Requested action not taken. File name not allowed.



---

# INDEX

---

## A

accessing DiskXtender 8  
append 14, 27  
ascii 14  
ASCII file transfer 15

## B

binary 14, 15, 34, 1  
binary file transfer 34  
bye 15, 19, 27

## C

caching 3, 5, 1  
cd 10, 11, 15, 27  
cdup 15, 27  
chgrp 21, 27  
chmod 22, 27  
chown 22, 28  
close 15, 28  
cp 11, 38

## D

delete 15, 28, 37, 38  
dir 1, 2, 4, 8, 9, 10, 11, 15, 16, 17, 18, 19, 20, 23, 24, 27, 28, 29, 30, 33, 34, 35, 37, 38, 1,  
2, 1, 2  
DiskXtender 1, 4, 5, 8, 13, 18, 21, 23, 24, 37, 38, 1, 2

## F

features 21  
file system 3, 7, 8, 9, 10, 1  
FTP 3, 5, 7, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 33, 35, 36, 37, 38, 1  
ftp 13, 14, 18, 20, 24, 33, 34, 35, 36

## G

get 3, 14, 16, 17, 19, 21, 22, 24, 25, 29, 35  
GTRSH 22, 26

## H

hash 16  
help 17, 21, 1

---

---

---

## **L**

lcd 17  
ln 23, 29  
logging in 33  
ls 9, 10, 11, 17, 29, 33, 38

## **M**

mdelete 17, 28  
mget 17, 18, 29  
migration 3, 4, 1  
mkdir 18, 30  
mount 8, 10, 1  
mput 18, 30, 34

## **N**

NFS 3, 5, 7, 8, 9, 11, 37, 38, 1  
NMDUP 23, 26

## **O**

open 15, 18, 30, 1

## **P**

prompt 8, 13, 14, 15, 18, 21, 22, 25, 30, 34  
purging 3, 4, 1  
put 3, 18, 19, 20, 23, 30, 38  
pwd 10, 19, 30, 33

## **Q**

quit 19, 27, 35  
quote 19, 21, 22, 23, 24, 25, 26, 1

## **R**

rename 20, 31, 37, 38  
reset 20, 31  
rmdir 20, 31

## **S**

s 1, 2, 3, 4, 7, 8, 10, 24, 35  
send 20, 30  
SETFAM 23, 24  
site 14, 21, 22, 23, 24, 25, 1  
stage 3, 19, 25, 32, 1  
status 20, 27, 28, 29, 30, 32, 1  
STRSH 25, 26

## **T**

trash can 3, 22, 25, 37, 38, 2

---

---

---

## **U**

umask 25, 32

user 1, 2, 3, 4, 5, 7, 9, 10, 11, 18, 19, 20, 23, 24, 30, 32, 33, 34, 35, 37, 1, 2, 1

## **W**

wait 23, 24, 33

waittime 25, 2