

High Performance Storage System User Guide

Table of Contents

| | |
|------------------------------------------------------------------|----|
| Preface | 3 |
| Introduction | 4 |
| Storage Summarized | 5 |
| Overview | 7 |
| User Interfaces | 7 |
| File Transfer Protocol (FTP) | 7 |
| Parallel FTP (PFTP) | 7 |
| LC Local Interfaces | 8 |
| NFT | 8 |
| HSI | 8 |
| Hopper | 9 |
| HTAR | 9 |
| Storage Concepts | 11 |
| Class of Service | 11 |
| Storage Class | 11 |
| Storage Hierarchy | 11 |
| Interface Usage Considerations | 13 |
| File Transfer Protocol (FTP) | 15 |
| Site (and Quote) Commands | 16 |
| CHGRP (Changing a File's Group by Name) | 16 |
| CHMOD (Changing a File's Permissions) | 17 |
| IDLE (Listing/Setting Idle Time) | 18 |
| SETCOS (Specifying a File's Class of Service) | 18 |
| STAGE (Staging a File) | 20 |
| SYMLINK (Creating a Symbolic Link) | 20 |
| UMASK (Changing the Default UMASK) | 21 |
| WAIT (Setting the Desired Wait Options for Migrated Files) | 21 |
| List Directory Extensions | 23 |
| Parallel File Transfer Protocol (PFTP) | 24 |
| Storage Groups | 25 |
| Disclaimer | 26 |
| Keyword Index | 27 |
| Alphabetical List of Keywords | 28 |
| Date and Revisions | 29 |

Preface

- Scope:** This High Performance Storage System (HPSS) User's Guide provides the necessary information for transferring files to or from LLNL's installation of HPSS. This reference manual, adapted with permission from similar publications whose copyright owners are listed in the note below, reflects HPSS use at Livermore Computing, LLNL.
- LC users seeking a concise introduction to local storage features, limitations, and customized storage-support tools should consult EZSTORAGE (URL: <https://computing.llnl.gov/LCdocs/ezstorage>).
- Availability:** Unless otherwise noted, HPSS and its features are available on both the open (OCF) and the secure (SCF) computing facilities at LLNL.
- Consultant:** For help, contact the LC hotline at 925-422-4531 (open e-mail: lc-hotline@llnl.gov, SCF e-mail: lc-hotline@pop.llnl.gov).
- Note:** Copyright (C) 1992, 2009 International Business Machines Corporation, The Regents of the University of California, Los Alamos National Security, LLC, Lawrence Livermore National Security, LLC, Sandia Corporation, and UT-Battelle.
All rights reserved.
HPSS Release 7 February 2009
- Printing:** The print file for this document can be found at:

OCF: <https://computing.llnl.gov/LCdocs/hpss/hpss.pdf>
SCF: http://www.llnl.gov/LCdocs/hpss/hpss_scf.pdf

Introduction

This document provides information for HPSS Release 7.1 and has been modified to reflect local LLNL features. Note: It is not the intent of this document to define the standard commands and subcommands provided by standard FTP. Only interface extensions provided by HPSS are defined within the HPSS User Guide. Note also that HPSS has no interface to support SCP (secure copy), so that you cannot use SCP to store files or retrieve stored files at LLNL. HTAR, which at LC acts as a specialized HPSS interface for storing and retrieving (individual members within) archive (TAR-like library) files, uses parallel file transfer behind the scenes but does not execute the PFTP client. Because "secure FTP" (SFTP) relies on a different server than standard FTP, you cannot use SFTP to store files or retrieve stored files at LLNL.

Refer to the HPSS User's Guide (Release 7.1) for the complete documentation for HPSS as issued by the HPSS Collaboration partners. Refer to the HPSS Installation Guide and the HPSS Management Guide for descriptions of the interfaces provided to HPSS administrators. Refer to the HPSS Programmer's Reference for programming interfaces provided to the end user. Refer to the HPSS Error Messages Manual for a list of all HPSS error and advisory messages that are output by the HPSS software. This documentation is available from the HPSS Collaboration Web Site at <http://www.hpss-collaboration.org/hpss/index.jsp> (URL: <http://www.hpss-collaboration.org/hpss/index.jsp>)

LC users seeking a concise introduction to local storage features, limitations, and customized storage-support tools should consult [EZSTORAGE](https://computing.llnl.gov/LCdocs/ezstorage) (URL: <https://computing.llnl.gov/LCdocs/ezstorage>). Standard FTP features as well as local customizations to support parallel transfers to and from storage are explained thoroughly in LC's [FTP Reference Manual](https://computing.llnl.gov/LCdocs/ftp) (URL: <https://computing.llnl.gov/LCdocs/ftp>). For efficient management of (very large) stored archive files, see the [HTAR Reference Manual](https://computing.llnl.gov/LCdocs/htar) (URL: <https://computing.llnl.gov/LCdocs/htar>). For a user-friendly interface that communicates with HPSS and makes it easy to transfer files and manipulate files and directories using familiar UNIX-style commands, see the [HSI](https://computing.llnl.gov/LCdocs/hsi) (URL: <https://computing.llnl.gov/LCdocs/hsi>) documentation.

Storage Summarized

This section briefly summarizes the chief storage-system constraints and tells how to perform the most important file-storage tasks at LC.

Storage System Constraints:

```

Largest allowed file size: 10 Tbyte (using FTP/NFT interface)
                           68 Gbyte/member           | (using HTAR
                           10 Tbyte/archive           | interface)
Longest file name:         1023 characters
Problem characters in file names:
  Treated as file filters:  ? * {a,b}
  Forbidden FIRST characters: # - ! ~
  Forbidden in any position: * ? [ {

```

Commands for Common File-Storage Tasks:

| TASK | FTP | NFT |
|--------------------------------------------------|-----------------------------------------------|-----------------------------------------------|
| Connect to storage: | ftp storage | nft |
| Make storage directory: | mkdir <i>dr</i> | (same) |
| Change storage directories: | cd <i>dr</i> | (same) |
| Store a file: | put <i>fl</i> | (same) |
| Retrieve a stored file: | get <i>fl</i> | (same) |
| Retrieve from within a stored archive: | See the HTAR Reference Manual | See the HTAR Reference Manual |
| Delete a stored file: | delete <i>fl</i> | (same) |
| List stored files: | dir | (same) |
| Change permissions (to <i>nnn</i>): | quote site chmod <i>nnn fl</i> | chmod <i>nnn fl</i> |
| Start migration of a stored file back from tape: | quote site stage <i>fl</i> | (none) |
| Control file overwriting: | | |
| .. Prevent overwriting | (none) | noclobber |
| .. Allow overwriting | (default) | clobber |
| Can be run visually by using Hopper: | yes | yes |

HPSS has no interface to support SCP (secure copy) or SFTP (secure FTP), so you cannot use SCP or SFTP to store files or retrieve stored files at LLNL. Also, LC's firewall prevents direct FTP connections to storage from any machine outside llnl.gov (see the [FTP section](#) (page 15) below for suggested alternatives). See [EZSTORAGE](https://computing.llnl.gov/LCdocs/ezstorage) (URL: <https://computing.llnl.gov/LCdocs/ezstorage>) for a concise, LC-oriented treatment of local storage features and problems (such as recursive deletions or sharing stored files), as well as an introduction to customized local storage-support tools (such as LSTORAGE, CHMODSTG, CHGRPSTG, and HTAR).

Storage Home Directories:

Regardless of their access software (FTP, NFT), LC users arrive at HPSS in their storage home directory. This always has a pathname of the form

`/users/u[00-54]/username`

where *username* is your LC login name (for example: `/users/u34/jsmith`). This basic directory structure supports customized division into subdirectories (for example, by invoking FTP's or NFT's MKDIR command) as well as access control of stored files through storage groups (page 25).

Overview

The High Performance Storage System (HPSS) provides scalable parallel storage systems for highly parallel computers as well as traditional supercomputers and workstation clusters. Concentrating on meeting the high end of storage system and data management requirements, HPSS is scalable and designed for large storage capacities, and to use network-connected storage devices to transfer data at rates up to multiple gigabytes per second.

User Interfaces

File Transfer Protocol (FTP)

HPSS provides an industry-standard FTP user interface. Because FTP is a serial interface, data sent to a user is received serially. This does not mean that the data within HPSS is not stored and retrieved in parallel. It simply means that the FTP daemon within HPSS must consolidate its internal parallel transfers into a serial data transfer to the user. HPSS FTP performance in many cases will be limited not by the speed of a single storage device, as in most other storage systems, but by the speed of the data path between the HPSS FTP daemon and the user's FTP client. [Note: On all LC production machines, the default FTP client is really the parallel interface PFTP, and all files over 4 Mbyte move to or from storage in parallel by default.]

All FTP commands are supported or properly rejected if the HPSS Parallel FTP daemon does not implement a specific feature. In addition, the ability to specify "Class of Service for" is provided via the quote site or site commands. Additional site command (page 16) options are provided for chgrp, chmod, stage, symlink, umask, and wait.

Different FTP servers process incoming FTP "M" commands (such as MDELETE) differently, with different file-transfer results. The FTP server(s) on HPSS at LC are unusual in interpreting MDELETE filters (such as TEST*) recursively; they remove all (matching) files not only in the current working directory but in the directory children of that directory as well. To minimize the impact of this aggressive interpretation of MDELETE on your stored files, see the "FTP Pitfalls (with Storage)" section of EZSTORAGE (URL: <https://computing.llnl.gov/LCdocs/ezstorage>).

Passive connections are not supported. Also, to avoid confusion, the user may want to explicitly specify the data transfer type of ascii or binary. For additional local interfaces to storage that use the standard FTP daemons underneath, see the LC Local Interfaces (page 8) section. Because the "secure FTP" (SFTP) client talks only to a different (SSHD2) daemon, SFTP is *not* an interface to HPSS at LLNL.

Parallel FTP (PFTP)

[Note: On all LC production machines, the default FTP client is really the parallel interface PFTP, and all files over 4 Mbyte move to or from storage in parallel by default.]

The PFTP client supports normal FTP commands plus extensions (including an LLNL-only PARALLEL toggle command). It is built to optimize FTP performance for storing and retrieving files from HPSS by allowing the data to be transferred in parallel to the client. The interface provided to the user has syntax similar to FTP but with some extensions to allow the user to transfer data to and from HPSS across parallel communication interfaces.

The following constraints are imposed by PFTP.

- Pipes are not supported.
- Passive connections are not supported.
- ASCII transfers are not supported over the parallel interface because ASCII transfers insert characters. This makes it impossible to send the data in parallel. Since extra characters are inserted in the stream, there is no way to resolve data placement. Warning: Some FTP implementations default to ascii. If this is the case, it will be necessary to specify binary by entering the bin command.
- PFTP client access is supported only from nodes that support the HPSS PFTP client software.

LC Local Interfaces

The interfaces to HPSS described below are local to LC and include command-line interfaces as well as graphical interfaces. Each interface has its own documentation in addition to the overview provided in this section.

Note: Additional local tools use FTP daemons for file transfer to storage but offer nonFTP interfaces. One (for workstations) is XDIR; another (for Macintosh computers) is Fetch.

NFT

NFT (Network File Transfer) runs on all open and secure LC production machines (but NOT necessarily on other local machines). NFT relies on standard FTP daemons, and so it inherits many of their file-transfer properties, but its user interface offers several features quite different from FTP:

- A special NFT server preauthenticates all NFT transfers, so all NFT executions are passwordless.
- NFT elaborately tracks and numbers all transfers. It automatically persists if system problems delay storing any file, and it keeps detailed records of your file-storage successes and problems.
- Input from and output to files is easy, and NFT's command syntax (unlike FTP's) lends itself to practical use in scripts and batch jobs.
- Some NFT commands especially facilitate transfers to and from storage (so some users regard NFT as primarily a file-storage rather than a general file-transfer tool). NFT even "routes" all storage-related file transfers automatically so as to benefit from jumbo-frame network connections to storage whenever they are available.

For a concise, task-oriented summary of how to use NFT commands and features, with annotated typical examples, consult the "Using NFT" section of [EZOUTPUT](https://computing.llnl.gov/LCdocs/ezoutput) (URL: <https://computing.llnl.gov/LCdocs/ezoutput>). For a complete analysis of NFT syntax and special features, along with a thorough alphabetical command dictionary, consult the [NFT Reference Manual](https://computing.llnl.gov/LCdocs/nft) (URL: <https://computing.llnl.gov/LCdocs/nft>).

HSI

The Hierarchical Storage Interface (HSI) utility communicates with the HPSS via a user-friendly interface that makes it easy to transfer files and manipulate files and directories using familiar UNIX-style commands. HSI supports recursion for most commands as well as CSH-style support for wildcard patterns and interactive command line and history mechanisms, among its many other features.

For details about how to run HSI and a complete list of its commands, see the [HSI](https://computing.llnl.gov/LCdocs/hsi) (URL: <https://computing.llnl.gov/LCdocs/hsi>) documentation.

Hopper

Hopper is a powerful interactive tool that allows users to graphically move, copy, find, delete, and otherwise operate on files. Operations include changing permissions, checking access, searching using wildcards, synchronizing directories, and many others. Key features include:

- Drag-and-drop interface for file transfers between local and remote hosts or between remote hosts.
- Diverse protocol support, including FTP, SSH, NFT, SFTP, HPSS client API, HTAR, HSI, and others.
- Simultaneous transfers, searches, and other operations.
- Search by content or by filename using wildcards, regular expressions, and other criteria.
- Synchronize directories between any two systems.
- Perform high-level data management tasks, including access checking, permission changing, etc.

More general background information on Hopper is available from the project Web site at <https://computing.llnl.gov/resources/hopper> (URL: <https://computing.llnl.gov/resources/hopper>). See "Getting Started" for instructions on how to download Hopper to your local desktop machine.

HTAR

On LC production machines only, HTAR is a separate, locally developed utility program that serves as a fast, parallel-transfer alternative to the PFTP client for storage access. HTAR combines a flexible file bundling tool with internally managed parallel access to HPSS, to let you store and selectively retrieve even very large sets of files very efficiently. (With HTAR's -F option, you can adapt this tool for file transfers to LC *nonstorage* hosts, too.)

HTAR's enhanced features include:

- Uses a TAR-like syntax and supports TAR-compatible archive files by relying on the POSIX 1003.1 TAR file format.
- Bundles files in memory using multiple concurrent threads and transfers them into an archive file built *directly* in storage by default, to avoid needing extra online disk space.
- Takes advantage of available parallel interfaces to storage to provide fast file transfers (measured at as high as 150 MB, which exceeds 30 times the typical rate for transferring small files separately).
- Uses an external index file to easily accommodate thousands of small files in any archive, and to support retrieval of specified files from *within* a still-stored archive without first retrieving the whole archive from HPSS.
- Imposes no limit on the total size of the archives that it builds (some have reached 200 GB successfully, with up to 1,000,000 members) and accepts input files (archive members) as large as 8 GB.

When HPSS is up and available to users, you can execute HTAR with a command line that has the general form

```
htar action archive [options] [filelist]
```

and the specific form

```
htar -c|t|x|X|K -f archivename [-BdEFhHILmMoOpSTvVwY] [flist]
```

where exactly one action and the *archivename* are always required, while the control options and (except when using *-c*) the *filelist* (or *flist*) can be omitted (and the options can share a hyphen flag with the action for convenience). Users familiar with TAR can guess how to run HTAR from this model (although there are some tricky syntax differences). Others should consult the [HTAR Reference Manual](https://computing.llnl.gov/LCdocs/htar) (URL: <https://computing.llnl.gov/LCdocs/htar>) for usage suggestions, annotated examples, technical tips, full option details, and known problems.

Storage Concepts

This section defines key HPSS storage concepts that have a significant impact on the usability of HPSS.

Class of Service

Class of Service (COS) is an abstraction of storage system characteristics that allows HPSS users to select a particular type of service based on performance, space, and functionality requirements. Each COS describes a desired service in terms of characteristics such as minimum and maximum file size, transfer rate, access frequency, latency, and valid read or write operations. A file resides in a particular COS and the class is selected when the file is created. Underlying a COS is a storage hierarchy that describes how data for files in that class are to be stored in HPSS.

For the FTP and PFTP interfaces, the COS ID may be explicitly specified by using the site setcos (page 18) command. If not specified, a default COS is used. A different NFT SETCOS command (URL: <https://computing.llnl.gov/LCdocs/nft/index.jsp?show=s9.34>) is available that is easy to use in scripts and whose current COS value is easy to check. [See the SETCOS (page 18) section below for a current list of COSs supported at LC and their intended roles.] If the COS ID is explicitly set by using the site setcos command, that COS will be used regardless of file size. [Note: At LLNL, the default COS is in fact dependent on file size except for files written with HTAR.]

A COS is implemented by a storage hierarchy of one to many storage classes. Storage hierarchies and storage classes are not directly visible to the user.

Storage Class

An HPSS storage class is used to group storage media together to provide storage with specific characteristics for HPSS data. The attributes associated with a storage class are both physical and logical. Physical media in HPSS are called physical volumes. Physical characteristics associated with physical volumes are the media type, block size, the estimated amount of space on volumes in this class, and how often to write tape marks on the volume (for tape only). Physical media are organized into logical virtual volumes. This allows striping of physical volumes. Some of the logical attributes associated with the storage class are virtual volume block size, stripe width, data transfer rate, latency associated with devices supporting the physical media in this class, and storage segment size (disk only). In addition, the storage class has attributes that associate it with a particular migration policy and purge policy to help in managing the total space in the storage class.

Storage Hierarchy

An HPSS storage hierarchy consists of multiple levels of storage with each level representing a different storage media (i.e., a storage class). Files are moved up and down the storage hierarchy via stage and migrate operations, respectively, based upon storage policy, usage patterns, storage availability, and user request.

For example, a storage hierarchy might consist of a fast disk, followed by a fast data transfer and medium storage capacity robot tape system, which in turn is followed by a large data storage capacity but relatively slow data transfer tape robot system. Files are placed on a particular level in the hierarchy depending on the

migration policy and staging operations. Multiple copies of a file may also be specified in the migration policy. If data is duplicated for a file at multiple levels in the hierarchy, the more recent data is at the higher level (lowest level number) in the hierarchy. Each hierarchy level is associated with a single storage class.

See the SETCOS (page 18) section below for the intended role of each storage class currently offered at LC.

Interface Usage Considerations

Guidance on when to use a particular HPSS interface or under what conditions a particular interface might be preferable is provided below.

FTP (Note that on all LC production machines, FTP defaults to PFTP automatically):

- Utilizes standard FTP interface. Users and applications familiar with FTP can access HPSS with the standard command set.
- Supports any FTP client platforms.
- FTP commands may be issued from any vendor nodes with an FTP interface. No specialized code is required.

PFTP (Note that on all LC production machines, FTP defaults to PFTP automatically):

- Provides faster file transfers. PFTP is a better performer than FTP because it provides the capability to stripe data across multiple client data ports.
- Supports partial file transfer. PFTP provides options on the pget and pput commands to perform partial file transfers. This would be beneficial to users who want to extract pieces of large files.
- On LC production machines, a locally added PARALLEL command lets you toggle parallel transfers on and off while you use regular FTP commands to actually move files.

NFT:

- Support of queued transfers.
- The NFT server preauthenticates your access to the machines where NFT works; hence, all NFT file transfers are passwordless.
- To take advantage of the significantly faster file transfers to or from HPSS that jumbo-frame network connections enable (especially helpful for transfers between the Lustre parallel file system and storage).
- Elaborate job tracking (e.g., uniquely numbering each job; commands to report job status before, during, and even after completion; user control of NFT's interactive messages about job progress; and ways to create and monitor "sessions" of related jobs).

HTAR:

- File-transfer needs specifically involve placing many files into or retrieving them from a remote archive (TAR-format library).
- Ability to retrieve individual files from a stored archive without moving the whole large archive back to your local machine first.

HSI:

- Interact with HPSS via a friendly interface, making it easy to transfer files and manipulate files and directories using familiar UNIX-style commands.
- HSI supports recursion for most commands (-R parameter), as well as CSH-style support for wildcard patterns, interactive command line and history mechanisms (among its many other features).

Hopper:

- Prefer an interactive tool that allows users to graphically move, copy, find, delete, and otherwise operate on files.
- Additional features such as authentication management, searching, data aggregation, directory manipulation, etc.

File Transfer Protocol (FTP)

FTP is supported from any FTP client platform (but see the limitation in the next paragraph and the warning about HPSS server differences farther below).

WARNING: LC uses its hardware/software security "firewall" to block direct FTP connections from machines outside the llnl.gov domain to LC machines within llnl.gov (including storage). This firewall blocking of incoming FTP connections means that if you have files on any machine outside llnl.gov and you want to store them in HPSS, you must either:

(1) First log on to a machine within llnl.gov, run FTP there to transfer (GET) your files to that within-llnl.gov machine, and then transfer (PUT) them again from there to storage, or

(2) Before you run FTP on your outside-the-firewall machine, get, install, configure, and execute a Virtual Private Network (VPN) client on that machine. Contact the LC Hotline to see if you are authorized to run a VPN client for access to LLNL. A VPN client borrows an llnl.gov IP address for your machine while it runs, and LC has confirmed that if you run VPN and FTP together under Windows, you can directly transfer files to storage.llnl.gov from outside the firewall (no staging to an LC production machine is needed). But you may encounter vendor-compatibility problems with some versions of Windows or with other operating systems. See LC's [Firewall and SSH Guide](https://computing.llnl.gov/LCdocs/firewall) (URL: <https://computing.llnl.gov/LCdocs/firewall>) for full instructions on getting and using VPN to enable FTP.

For information on additional user interfaces that also rely on the FTP daemons underneath (but not on FTP clients), see the [LC Local Interfaces](#) (page 8) section. SFTP is *not* an interface to HPSS on LC machines.

HPSS supports the FTP command set for transferring files to and from HPSS. To use FTP, the user enters the following:

```
ftp storage.llnl.gov
```

At this point, any standard FTP command may be entered. Note: If you receive an error message, contact your HPSS administrator. An error message generally implies that either HPSS is not correctly configured or some HPSS components may not be executing.

Different FTP servers process incoming FTP "M" commands (such as MDELETE) differently, with different file-transfer results. The FTP server(s) on HPSS at LC are unusual in interpreting MDELETE filters (such as TEST*) recursively; they remove all (matching) files not only in the current working directory but also in the directory children of that directory as well. To minimize the impact of this aggressive interpretation of MDELETE on your stored files, see the "FTP Pitfalls (with Storage)" section of [EZSTORAGE](#) (URL: <https://computing.llnl.gov/LCdocs/ezstorage>).

On all LC production machines, typing FTP automatically executes the locally enhanced PFTP (parallel FTP) client, and all files over 4 Mbyte automatically move to or from storage by parallel file transfer (unless you explicitly request otherwise). See the [FTP Reference Manual](#) (URL: <https://computing.llnl.gov/LCdocs/ftp>) for how to force execution of the serial FTP client and for general advice on using parallel FTP service effectively in the LLNL computing environment. Note that HTAR transfers files to or from stored archives (TAR-format libraries) very efficiently in parallel but *without* running the PFTP client.

WARNING: If you work on LC Linux/CHAOS systems, you have access to /usr/kerberos/bin/ftp, but you should instead run /usr/bin/ftp. Under some conditions the former (but not the latter) refuses to log you into HPSS or needlessly asks you to "Please login with USER and PASS".

Site (and Quote) Commands

The site commands listed below are supported in HPSS at LC.

- `chgrp (*)`
- `chmod (*)`
- `idle`
- `setcos (*)`
- `stage`
- `symlink`
- `umask`
- `wait`

NFT (URL: <https://computing.llnl.gov/LCdocs/nft>) offers its own versions of those commands marked (*) above, and the NFT version may be easier to use inside job scripts. Some FTP clients let you type (some of) these commands without a prefix (e.g., `chmod`), some clients expect the `site` prefix (as shown in this manual), and some clients require the longer `quote site` prefix (e.g., `quote site stage`). Only testing will reveal which format your FTP client requires on a specific machine, although the `quote site` long form is always an acceptable fallback.

Details for each supplementary command appear in the following subsections.

CHGRP (Changing a File's Group by Name)

CHGRP is used to change the group name of a file and has the following format:

```
site chgrp grp file
```

where,

grp is the new group name of the file.

file is the name of the file.

The user must belong to the specified group and be the owner of the file, or be the root user. (For an alternative way to change a stored file's group, see the instructions for the CHGRPSTG tool in EZSTORAGE (URL: <https://computing.llnl.gov/LCdocs/ezstorage>.)

Example: The following may be entered to change the group of myfile to group mygroup.

```
site chgrp mygroup myfile
```

CHMOD (Changing a File's Permissions)

CHMOD is used to change the group mode (permissions) of a file and has the following format:

```
site chmod mode file
```

where,

mode is the new octal mode number of the file. Symbolic CHMOD parameters (such as u or g) are not supported by the FTP CHMOD command. For a good short review of the octal permission notation, see [EZFILES](https://computing.llnl.gov/LCdocs/ezfiles) (URL: <https://computing.llnl.gov/LCdocs/ezfiles>). (For an alternative way to change a stored file's permissions (octal or symbolic), see the instructions for the CHMODSTG tool in [EZSTORAGE](https://computing.llnl.gov/LCdocs/ezstorage) (URL: <https://computing.llnl.gov/LCdocs/ezstorage>)).

file is the name of the file.

Mode is constructed from the OR of the following modes:

```
0400 read by owner
0200 write by owner
0100 execute (search in a directory) by owner
0040 read by group
0020 write by group
0010 execute (search in a directory) by group
0004 read by others
0002 write by others
0001 execute (search in a directory) by others
```

Note: The following mode values are not supported:

```
4000 set user ID on execution
2000 set group ID on execution
1000 sticky bit
```

Only the owner of the file or root user can change its mode.

Example: The following may be entered to change the mode of myfile to read, write by owner and group.

```
site chmod 0660 myfile
```

IDLE (Listing/Setting Idle Time)

IDLE is used to determine/change the default idle time (in seconds) and has the following format:

```
quote site idle time
```

where,

time is a number in seconds less than the maximum idle time.

Example: The following may be entered to set the idle time to 1000 seconds.

```
quote site idle 1000
```

Without a size specifier, the current idle time is returned:

```
quote site idle  
Current IDLE time limit is 1000 seconds; max 7200
```

SETCOS (Specifying a File's Class of Service)

SETCOS is used to specify a Class of Service (COS) and has the following format:

```
site setcos cos_id
```

where,

cos_id is the COS identifier (used when creating a new HPSS file during a put operation).
WARNING: HPSS does not verify the *cos_id* that you supply, and it now accepts without error any integer as a *cos_id* (even if no such COS exists), but your file transfer will fail.

COS is used as a means for specifying how the storage system handles the incoming files. For example, COS may determine the number of copies kept or the amount of parallelism or stripe width for a file. If a COS is not specified, a default is used (and using the default COS at LC is recommended for maximum efficiency).

At LC, HPSS currently supports the following COSs on both the OCF and SCF storage systems (although other COS values are also available):

| COS ID | Default for the file size(*) | Copies kept | Slang term |
|--------|------------------------------|-------------|---------------|
| 110 | 0 .GE. file .LT. 4 Mbyte | 2(*) | small (sFTP) |
| 120 | 4 .GE. file .LT. 33 Mbyte | 2(*) | medium (mFTP) |
| 135 | 33 .GE. file .LT. 268 Mbyte | 1 | large (lFTP) |
| 160 | 268.GE. file | 1 | jumbo (jFTP) |
| 200 | dualcopy by request | 2 | |

(*)For files stored using FTP or NFT. All files stored using HTAR, regardless of size, have COS 160 (and hence single-copy storage) by default. COS 200 allows users to overtly request dual-copy storage even for large files or for files of any size stored (for mission critical files only, please).

LC's goal is for the FTP storage server to automatically detect incoming file size (and sending client) and assign a COS appropriately, with no intervention by the user. COS details at LC may change with time to optimize storage system performance.

In the example below, the following command might be entered to put a large file into HPSS with a COS ID 200, to get the protection of dual-copy storage for a large but mission critical file:

```
site setcos 200
```

The NFT SETCOS command (URL: <https://computing.llnl.gov/LCdocs/nft/index.jsp?show=s9.34>) simplifies specifying storage COS within a job script, and you can check the current value with NFT's STATUS command. There are some complexities about previously stored files and nonallowed COS values, however, so consult the NFT Reference Manual (URL: <https://computing.llnl.gov/LCdocs/nft>) for details.

STAGE (Staging a File)

STAGE is used to initiate a stage of a migrated file (e.g., from tape to disk). The user can initiate the stage and then return at a later time to initiate the file transfer using the FTP get or PFTP pget commands:

```
site stage file
```

where,

file is the name of the file.

Example: The following may be entered to stage file /home/smith/myfile.

```
site stage /home/smith/myfile
```

SYMLINK (Creating a Symbolic Link)

SYMLINK is used to create a symbolic link:

```
site symlink path/file link
```

where,

path/file refers to the destination.

link refers to the local file name.

Example: The following may be entered to create a link named sys_passwd in the local directory pointing to /etc/passwd.

```
site symlink /etc/passwd sys_passwd
```

A DIR command will show sys_passwd -> /etc/passwd.

UMASK (Changing the Default UMASK)

UMASK is used to change the default umask:

```
site umask octal-mask
```

where,

octal-mask refers to the octal mask to be applied.

Example: The following may be entered to change the default umask to 2.

```
site umask 0002
```

When issued without an octal value, the active umask is displayed.

WAIT (Setting the Desired Wait Options for Migrated Files)

WAIT is used to notify the HPSS PFTP daemon.

```
site wait option
```

where,

option is one of the following values:

- 1 or inf[inite] wait forever for the file to be staged. Do not return from the get or pget command to complete until the file has been transferred or a transfer error has occurred.
- 0 do not wait for the file to be staged. If the file has been migrated, return the appropriate message and initiate the stage. The user will return later to reissue the get or pget command.
- n* (an integer) wait the specified period (in seconds) for the file requested by a get or pget command to complete. Either transfer the file if the file is staged within the specified period or return a reply to notify the user to try again later.

Example: The following may be entered to wait for files to be staged.

```
site wait -1
```

The following table describes the behavior the user should expect from FTP when issuing the stage/wait commands. Note: Only COSs utilizing the "Stage on Open Background" option will exhibit predictable results.

Stage/Wait Behaviour

| Wait Time | File Condition | Command | Behavior/Message |
|------------------|-----------------------|----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| No Wait | Archived | site stage xyz | "File xyz is being retrieved from archive." |
| No Wait | Not Archived | site stage xyz | "File xyz is currently ready for other processing." |
| Wait <i>nnn</i> | Archived | site stage xyz | Wait for period then receive message: "File xyz is currently ready for other processing." or "File xyz is currently ready for other processing." if the file is staged in the time frame allowed. |
| Wait <i>nnn</i> | Not Archived | site stage xyz | "File xyz is currently ready for other processing." |
| No Wait | Archived | get xyz | "File xyz is being retrieved from archive." |
| No Wait | Not Archived | get xyz | Transfers data as expected. |
| Wait <i>nnn</i> | Archived | get xyz | Wait for period then receive message: "File xyz is being retrieved from archive." or transfers data as expected if file is staged in the time allowed. |
| Wait <i>nnn</i> | Not Archived | get xyz | Transfers file as expected. |

List Directory Extensions

FTP supports the `ls` command to list the contents of a directory. Standard options supported are: `ls`, `ls -l`, `ls -a`, and `ls -F`. In addition to the standard `ls` options generally provided, HPSS also provides a `-lh` option. If `-lh` is specified, then a long directory listing is generated. However, in place of the owner field (field 3) and group field (field 4) listed for the `-l` option, the COS identifier and account code are listed.

Example: `ls -lh`

```
-rw-rw---- 1 1          198      157286400 May 13 1996  TEST
-rw-r--r-- 1 1          160           32768 May 16 1996  prod1
-rw-r--r-- 1 1          160           32768 May 16 1996  prod10
-rw-r--r-- 1 1          160           32768 May 16 1996  prod11
-rw-r--r-- 1 1          160           32768 May 16 1996  prod12
-rw-r--r-- 1 1          160           32768 May 16 1996  prod13
-rw-r--r-- 1 1          160           32768 May 16 1996  prod14
-rw-r--r-- 1 1          160           32768 May 16 1996  prod15
-rw-r--r-- 1 1          160           32768 May 16 1996  prod151
-rw-r--r-- 1 1          160           32768 May 16 1996  prod152
```

For an alternative way to list stored files and their attributes, including recursive listings of all or part of your storage hierarchy easily redirected to a file, see the instructions for the `LSTORAGE` tool in [EZSTORAGE](https://computing.llnl.gov/LCdocs/ezstorage) (URL: <https://computing.llnl.gov/LCdocs/ezstorage>).

Parallel File Transfer Protocol (PFTP)

Whenever you run FTP on LC production machines (OCF and SCF), but not necessarily on special-purpose machines, you are automatically executing a customized PFTP client. For advice on how PFTP interacts with the jumbo-frame (high-capacity) links on some LC machines, and for information on the PARALLEL command added to the local implementation of PFTP, see the "Parallel FTP Service" section of the FTP Reference Manual (URL: <https://computing.llnl.gov/LCdocs/ftp>).

It is recommended at LC to simply use FTP (rather than PFTP) to transfer files to archival storage. The HPSS PFTP client does, however, support many command-line options that, while generally not needed (or used) at LC, can be used to customize PFTP command lines. Refer to the documentation available from the HPSS Collaboration Web site at http://www.hpss-collaboration.org/hpss/users/docs/AdobePDF/7.1/users_guide.pdf (URL: http://www.hpss-collaboration.org/hpss/users/docs/AdobePDF/7.1/users_guide.pdf). In addition, several additional site (and quote) commands (e.g., ACL, BUFSIZE, MINFO) are available with PFTP. These are also described in the aforementioned documentation.

Note: If the message "Load rescue failure: audit info" is received, contact your HPSS administrator. This message generally implies that either HPSS is not correctly configured, or some HPSS components may not be executing.

Storage Groups

A group is a named set of users created by a system administrator to enable easier file sharing among group members. On a production machine, assigning a file to a group (with CHGRP) enables other group members to take advantage of whatever group permissions you have declared (with CHMOD) to list, read, or overwrite the file.

Groups on the HPSS storage systems (open and secure) at LC are obtained from LDAP (lightweight directory access protocol), just as they are on LC production machines. See the general advice about [Using Groups](https://computing.llnl.gov/LCdocs/ezfiles/index.jsp?show=s9) (URL: <https://computing.llnl.gov/LCdocs/ezfiles/index.jsp?show=s9>) in EZFILES.

WHEN YOU STORE OR RETRIEVE:

Your (default) shell level (or directory inherited) group is not conveyed through FTP sessions to remote systems, including to and from storage. So, if you assign a file to a (nondefault) group and later store the file in HPSS, that group assignment (as well as the file timestamp) is lost. HPSS always associates the stored file with your single-member user name group by default. Likewise, if you assign a stored file to a (nondefault) group and later retrieve it, that storage group assignment is lost as well.

CHANGING STORAGE GROUPS:

To enable or limit the sharing of stored files, you can change the group assignment of a file *after* you store it either by using the supplementary FTP command

```
quote site chgrp grpname filename
```

for example

```
quote site chgrp us_cit myfile3
```

or by using the NFT command

```
chgrp grpname filename
```

just as you would use the CHGRP utility on a production machine. You can only assign a stored file to a group to which you belong. Similarly, the supplementary FTP command

```
quote site chmod mode filename
```

will set a stored file's (group) permissions to the octal *mode* that you specify. NOTE: You can also change a stored file's group by using the special CHGRPSTG tool and change its permissions by using the CHMODSTG tool (as well as listing your storage directories and the files they contain using LSTORAGE). Instructions for and examples of these local tools to manage storage groups appear in [EZSTORAGE](https://computing.llnl.gov/LCdocs/ezstorage) (URL: <https://computing.llnl.gov/LCdocs/ezstorage>).

Disclaimer

This document was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor Lawrence Livermore National Security, LLC nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial products, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or Lawrence Livermore National Security, LLC. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or Lawrence Livermore National Security, LLC, and shall not be used for advertising or product endorsement purposes.

Keyword Index

To see an alphabetical list of keywords for this document, consult the [next section](#) (page 28).

| Keyword | Description |
|-----------------------------|-------------------------------------------|
| <u>entire</u> | This entire document. |
| <u>title</u> | The name of this document. |
| <u>scope</u> | Topics covered in HPSS Guide. |
| <u>availability</u> | Where these programs run. |
| <u>who</u> | Who to contact for assistance. |
| <u>introduction</u> | Role and goals of HPSS Guide. |
| <u>ezstorage</u> | Storage tasks and limits summarized. |
| <u>overview</u> | HPSS concepts, requirements, issues. |
| <u>interfaces</u> | Alternative HPSS user interfaces. |
| <u>ftp-summary</u> | FTP as storage interface, intro. |
| <u>pftp-summary</u> | Parallel FTP as storage interface, intro. |
| <u>local-interfaces</u> | LC local interfaces to HPSS. |
| <u>nft-summary</u> | NFT as storage interface, intro. |
| <u>hsi-summary</u> | HSI as storage interface, intro. |
| <u>hopper-summary</u> | Hopper as storage interface, intro. |
| <u>htar-summary</u> | HTAR as storage interface, intro. |
| <u>storage-concepts</u> | Three concepts used by HPSS. |
| <u>class-of-service</u> | "Class of service" (COS) defined. |
| <u>storage-class</u> | Storage classes to group storage media. |
| <u>storage-hierarchy</u> | Stage and migrate between storage levels. |
| <u>interface-comparison</u> | Comparison of storage interfaces. |
| <u>ftp</u> | FTP storage options, details. |
| <u>site-commands</u> | Extra SITE-prefix HPSS/FTP options. |
| <u>chgrp</u> | Change group by using name. |
| <u>chmod</u> | Change access permissions. |
| <u>idle</u> | List/set idle time. |
| <u>setcos</u> | Specify class of service. |
| <u>stage</u> | Prepare migrated file for retrieval. |
| <u>symlink</u> | Create a symbolic link. |
| <u>umask</u> | Change the default umask. |
| <u>wait</u> | Set WAIT option for migrated files. |
| <u>list-storage</u> | Special stored-file reporting option. |
| <u>pftp</u> | Parallel FTP (PFTP) storage. |
| <u>storage-groups</u> | Using, changing storage groups. |
| <u>index</u> | The structural index of keywords. |
| <u>a</u> | The alphabetical index of keywords. |
| <u>date</u> | The latest changes to HPSS Guide. |
| <u>revisions</u> | The complete revision history. |

Alphabetical List of Keywords

| Keyword ----- | Description ----- |
|-----------------------------|--------------------------------------------|
| <u>a</u> | The alphabetical index of keywords. |
| <u>availability</u> | Where these programs run. |
| <u>chgrp</u> | Change group by using name. |
| <u>chmod</u> | Change access permissions. |
| <u>class-of-service</u> | "Class of service" (COS) defined. |
| <u>date</u> | The latest changes to HPSS Guide. |
| <u>entire</u> | This entire document. |
| <u>ezstorage</u> | Storage tasks and limits summarized. |
| <u>ftp</u> | FTP storage options, details. |
| <u>ftp-summary</u> | FTP as storage interface, intro. |
| <u>hsi-summary</u> | HSI as storage interface, intro. |
| <u>hopper-summary</u> | Hopper as storage interface, intro. |
| <u>htar-summary</u> | HTAR as storage interface, intro. |
| <u>idle</u> | List/set idle time. |
| <u>index</u> | The structural index of keywords. |
| <u>interface-comparison</u> | Comparison of storage interfaces. |
| <u>interfaces</u> | Alternative HPSS user interfaces. |
| <u>introduction</u> | Role and goals of HPSS Guide. |
| <u>list-storage</u> | Special stored-file reporting option. |
| <u>nft-summary</u> | NFT as storage interface, intro (local). |
| <u>overview</u> | HPSS concepts, requirements, issues. |
| <u>pftp</u> | Parallel FTP (PFTP) storage options. |
| <u>pftp-summary</u> | Parallel FTP as storage interface, intro. |
| <u>revisions</u> | The complete revision history. |
| <u>scope</u> | Topics covered in HPSS Guide. |
| <u>setcos</u> | Specify class of service. |
| <u>site-commands</u> | Extra SITE-prefix HPSS/FTP options. |
| <u>stage</u> | Prepare migrated file for retrieval. |
| <u>storage-class</u> | Storage classes to group storage media. |
| <u>storage-concepts</u> | Three concepts used by HPSS. |
| <u>storage-groups</u> | Reporting, using, changing storage groups. |
| <u>storage-hierarchy</u> | Stage and migrate between storage levels. |
| <u>symlink</u> | Create a symbolic link. |
| <u>title</u> | The name of this document. |
| <u>umask</u> | Change the default umask. |
| <u>wait</u> | Set WAIT option for migrated files. |
| <u>who</u> | Who to contact for assistance. |

Date and Revisions

| Revision Date | Keyword Affected | Description of Change |
|---------------|-----------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ----- | ----- | ----- |
| 21Sep09 | entire | Complete revision to reflect HPSS 7.1. |
| 27Mar09 | <u>setcos</u> <u>entire</u> | Dualcopy COS ID added. Removed references to NETMON. |
| 24Oct07 | pput pget | Offset options disabled. Offset options disabled. |
| 18Apr07 | <u>nft-summary</u> <u>class-of-service</u> <u>setcos</u> | NFT jumbo-frame routing noted. NFT SETCOS command added. NFT SETCOS compared with FTP SETCOS. Some COS values changed. |
| 06Nov06 | user-id dce-role <u>storage-groups</u> | HPSS DCE support discontinued. HPSS DCE support discontinued. LDAPSEARCH replaces DCECP to report storage groups at LC. |
| 06Jul06 | utility-summary user-id dce-role <u>ftp</u> user-utilities <u>storage-groups</u> | CHACL and LSACL discontinued. Kerberos KINIT replaces DCE login. Discontinued on OCF. /usr/kerberos/bin/ftp warning added. CHACL, LSACL discontinued, deleted. DCECP not on CHAOS or AIX 5.3. |
| 05Apr06 | <u>ezstorage</u> <u>nft-summary</u> organization <u>storage-groups</u> | Storage home directories clarified. LANL, Sandia users can now use NFT. Chapter 6 cross ref added. Form URLs both updated. |
| 25Oct04 | user-utilities chacl lsacl | NFT's ACL commands discontinued. NFT's ACL commands discontinued. NFT's ACL commands discontinued. |
| 14Sep04 | user-utilities chacl lsacl htar <u>introduction</u> <u>ftp</u> | Cross ref added to NFT's ACL commands. Cross ref added to NFT's ACLADD. Cross ref added to NFT's ACLSHOW. Options, relation to PFTP updated. HTAR does not run PFTP client. HTAR does not run PFTP client. |
| 02Dec03 | <u>ezstorage</u> htar | Size and scope details updated. Limits updated, -F option noted. |
| 05May03 | parallel | More automatic parallel transfers. |
| 18Feb03 | <u>introduction</u> <u>ezstorage</u> <u>ftp-summary</u> | SFTP not an HPSS interface. SFTP not an HPSS interface. SFTP uses nonFTP daemon. |

ftp SFTP uses nonFTP daemon.

12Nov02 class-of-service NETMON tracking role noted.
interface-comparison NETMON tracking role noted.
setcos NETMON uses HPSS COSSs to track FTP.

18Jun02 class-of-service Details and roles updated.
setcos COS table revised, expanded to include duplicate copy info.
ftp-summary MDELETE warning added.
ftp MDELETE warning added.
htar Details revised, Y added.
ezstorage Maximum file size clarified.

23Aug01 pftp HTAR as PFTP front end.
htar Brief feature summary added.
introduction HTAR storage role noted.
ezstorage HTAR storage role noted.
index New keyword for new section.

05Jul01 pftp Now default, many local usage details added for all commands.
parallel Local command explained (added).
interfaces FTP/PFTP comparison updated.
ftp Actually invokes PFTP now.
introduction FTP/PFTP interaction noted.
index New keyword for new section.

23May00 scope EZSTORAGE guide noted.
introduction EZSTORAGE guide noted.
ezstorage Cross reference added.
chgrp CHGRPSTG tool in EZSTORAGE noted.
chmod CHMODSTG tool in EZSTORAGE noted.
list-storage LSTORAGE tool in EZSTORAGE noted.
user-utilities LSTORAGE, CHMODSTG, CHGRPSTG noted.
storage-groups New group-relevant tools noted.

02Apr99 ezstorage Firewall now blocks outside access.
ftp Firewall now blocks outside access.

08Feb99 introduction No SCP access to storage.
ezstorage No SCP access to storage.
ftp Firewall alert added.

10Sep98 storage-groups New section on using storage groups.

13Jul98 ezstorage New section summarizes storage.
nft-summary New section introduces NFT.
differences New section contrasts HPSS and UniTree.
index New keywords for new sections.
interfaces Warnings when not at LC.
site-commands Warnings about quote syntax.
setcos Local COSSs listed.
user-utilities Warnings that not generally avail.

| | | |
|---------------|------------------------------------|-----------------------------------------------------------------------------------------|
| 15Jun98 | entire <u>index</u> <u>a</u> | Cross references added throughout. Keyword index added. Alphabetical index added. |
| 11Jun98 | entire | First edition of LC HPSS Guide. |
| EJG (11Sep09) | | |

UCRL-WEB-201755

Privacy and Legal Notice (URL: <http://www.llnl.gov/disclaimer.html>)

EJG (21Sep09) Contact on the OCF: lc-hotline@llnl.gov, on the SCF: lc-hotline@pop.llnl.gov