HSI Reference Manual

Hierarchical Storage Interface, Version 11.1, 09 August 2024

Revision History	
09 August 2024	H

Table of Contents

Release information. Version 11.1, 09 August 2024

Copyright notification. Copyright © 1998-2004 Gleicher Enterprises, LLC.

Trademark usage. High Performance Storage System is a trademark of International Business Machines Corporation.

IBM is a registered trademark of International Business Machines Corporation.

IBM, DB2, DB2 Universal Database, AIX, pSeries, and xSeries are trademarks or registered trademarks of International Business Machines Corporation.

AIX and RISC/6000 are trademarks of International Business Machines Corporation.

UNIX is a registered trademark of the Open Group.

Linux is a registered trademark of Linus Torvalds in the United States and other countries.

Kerberos is a trademark of the Massachusetts Institute of Technology.

Java is a registered trademark of Oracle and/or its affiliates.

ACSLS is a trademark of Oracle and/or its affiliates.

Microsoft Windows is a registered trademark of Microsoft Corporation.

DST is a trademark of Ampex Systems Corporation.

Other brands and product names appearing herein may be trademarks or registered trademarks of third parties.

About this book. The *HSI Reference Manual* is structured as follows:

- *Chapter 1: Introduction* - Provides a high-level introduction to HSI.
- Chapters 2 through 15 describes concepts, configuration, and usage information.
- *Chapter 16: HSI keywords* - defines the keywords used for many HSI commands.
- *Chapter 17: HSI commands (alphabetical list)* - provides a list of the commands in alphabetical order.
- *Chapter 18: HSI commands (by function)* - provides a list of the commands grouped by function, in tabular form.
- *Chapter 19: Command reference* - Defines in detail each HSI command.

Typographic and keying conventions. This document uses the following typographic conventions:

Example commands that should be typed at a command line will be proceeded by a percent sign ("%") and be presented in a boldface courier font:

```
% sample command
```

Any text preceded by a pound sign ("#") should be considered comment lines:

```
# This is a comment
```

Italic   *Italic* words or characters represent variable values to be supplied.

[ ]   Brackets enclose optional items in syntax and format descriptions.

{ }   Braces enclose a list of items to select in syntax and format descriptions.

Chapter 1. Introduction

HSI is intended to provide a simple-to-use command line interface to the High Performance Storage System (HPSS). It has evolved over the years from a simple HPSS Client API application that only ran on DCE/Encina-based AIX platforms to its current version, which uses its own libraries and a client-server architecture to provide HPSS access from almost all current UNIX-based systems, including Cygwin running on Windows-based platforms.

HSI was designed primarily as an alternative to the original PFTP that was part of the HPSS offering. It provides a number of features that are useful both to the user community and to HPSS system administrators, such as wildcard pattern matching, recursion for most commands, the ability to connect to multiple HPSS systems in a single HSI session and move files between systems using 3rd party copy protocols, and automatic optimization of tape mounts for operations such as "change COS" and "get".

Since different sites have different authentication requirements, HSI supports several different methods for verifying the end user's identity. The HPSS site administrator can choose at compile time which authentication method(s) should be enabled. The administrator also chooses a "default" mechanism to use - users can override the default authentication method at run time via a command line option (**-A**), or by specifying the method in

their private $HOME/.hsirc file. Only authentication mechanisms that are supported on both the client and server can be used.

HSI uses strong encryption provided by the OpenSSL libraries for passing sensitive information, such as passwords, across the network.

File transfers automatically make use of HPSS parallel file transfers without requiring any special action on the user's part. Configuration files allow HPSS site administrators to tune network settings, specify a list of local network interfaces to be used on particular hosts, and to specify a restricted port range for sites that use firewalls in front of client hosts. HSI also supports a "firewall" mode for sites that cannot open a port range, which allows HSI to transfer data to HPSS via a store-and-forward mechanism to the HSI Gateway Server.

For sites that have a requirement for high-speed parallel transfers between cluster filesystems such as IBM's GPFS and Lustre, HSI supports the use of the multinode Transfer Agent. By using the Transfer Agent, HSI file transfers are not restricted to rates that can be achieved by a single node, but can scale up by using multiple nodes. Rates of over 2 GB/second from HPSS have been demonstrated using HSI with the Transfer Agent.

HSI continues to evolve to support new HPSS features and to accommodate the diverse needs of the user community. Suggestions from the HSI user community for new features or enhancements of existing features are always welcomed.

Chapter 2. Authentication Methods

HSI supports several different methods by which a user can authenticate his or her self. The methods that are enabled at a site are defined by the HPSS administrator when the HSI package is compiled.

The **-A** command line option can be used to specify the authentication mechanism to be used, for example: `hsi -A kerberos …`

The following authentication methods are available:

combo   (previously known as "dce") For this method, a user name and password combination are specified. For backward compatibility, the mechanism can still be specified as "dce"; however, this is deprecated and will be removed in a future release.

keytab   For this method, a Kerberos keytab file or a UNIX keytab file is read by the HSI library and passed to the HSI Gateway Server over the link after first encrypting the contents. On the server side, the file is decrypted and verified.

Kerberos   This mechanism uses the MIT Kerberos libraries to verify the user's existing credentials. A Kerberos keytab file can also be specified for Kerberos authentication. If so, the **kinit** program is first used to obtain the user credentials, and then authentication proceeds normally.

ident   This mechanism uses the trusted server protocol to obtain the user's identity.

gsi   This mechanism uses the Globus GSI protocols to obtain the user's identity.

Chapter 3. Restricted TCP ports and restricted Mover hosts

3.1. HSI port ranges and reserved ports

The HSI package makes use of reserved ports as follows:

- For initial connection to the HSI Gateway Daemon (HSIGWD) by the HSI and HTAR Clients. This is the port on which **inetd** or **xinetd** listens for connections for starting up the HSIGWD. The default port is 1217, and is the port assigned by the Internet Assigned Numbers Authority (IANA) for this server. On the HSIGWD system, the normal /etc/services and inetd or xinetd.d setup is used for launching the program in response to connections from clients.
- For authentication by mechanisms that use a private socket for protocols. Currently Kerberos and Globus GSI require private sockets.
- For I/O transfers on client hosts that use firewalls to restrict inbound connections or that simply want to use a certain range of TCP ports for HSI/HTAR transfers.
- For I/O transfers that set HSI "firewall" mode for data transfers. In this mode, the HSIGWD server listens for connections from the client, and uses store-and-forward mode for transferring data.

The HPSS_PORT_RANGE environment variable is used by the HSI package for defining a range of ports to use for all connections except the initial connection to the HSIGWD. The syntax is:

HPSS_PORT_RANGE=*start-end*

where *start* and *end* are the beginning and ending port number in the range.

Note

Older versions of HPSS and HSI/HTAR (and PFTP) prior to HPSS 7.5 supported two other environment settings for this purpose:

RPC_RESTRICTED_PORTS=ncacn_ip_tcp[*start-end*]

HPSS_PFTPC_PORT_RANGE=ncacn_ip_tcp[*start-end*]

The RPC_RESTRICTED_PORTS setting was part of the Distributed Computing Environment (DCE) that was required for HPSS versions prior to 6.0. The HSI and HTAR client libraries still support these as of HPSS 7.5.1, but they are deprecated and will be removed in a future release. They are only checked for if the HPSS_PORT_RANGE is *not* found in the environment.

The HPSS client libraries used by the HSIGWD through HPSS 7.5.2 have removed the HPSS_PFTPC_PORT_RANGE setting but still allow RPC_RESTRICTED_PORTS if HPSS_PORT_RANGE is not found in the environment.

On the HSIGWD server, the port range can be specified either in `/var/hpss/etc/env.conf`, where it applies to all applications and servers that run on the machine, or via the normal xinetd stanza setting for the HSIGWD, for example: `env +=HPSS_PORT_RANGE=20100-30100`.

3.2. HSI restricted Mover hosts

HSI and HTAR support the ability for the HSI/HPSS administrator to specify which HPSS mover hosts at a site are allowed to connect in order to perform I/O. This optional feature was added to avoid problems that can occur during transfers when site security port scans take place. It is described in detail in *Chapter 14, HPSS.conf settings (client)*. Normally, when this feature is needed at a site, the HPSS.conf file is set up by the site administrators when HSI/HTAR are installed on a client system.

Chapter 4. HSI command line

The HSI command line has the following format:

```
hsi [options] [command [;command [;…]]]
```

If the optional command-string is specified, HSI is said to be running in one-liner mode. In this mode, HSI will execute the command-string (which may consist of multiple semicolon-separated commands), and then terminate.

Command line options are as follows:

-help
> Displays a built-in synopsis of HSI usage. If used as the only command line option, HSI exits instead of entering interactive mode.

-?
> Same as **-help**

-a *accountID | accountName*
> Specifies the account ID or account name to be used for creating new files and directories.

-A *auth_mechanism*

> Specifies the authentication mechanism to be used when HSI starts up. Only authentication mechanisms that are available on both the client and HSI Gateway Server can be used successfully. Valid mechanisms are:

> **combo**   User name and password will be used. If the site uses RSA SecurID one-time passwords, the password is the current OTP setting.

> > **Note**

> > The deprecated "dce" authentication mechanism is still recognized as an alias for the "combo" mechanism, but its use should be discontinued, as it will be removed in a future release.

> **kerberos** Specifies that MIT Kerberos will be used for authentication.

> **keytab**   Specifies that a UNIX or Kerberos-style keytab will be used for authentication.

> **gsi**   Specifies that the Globus GSI certificate protocol will be used.

local   (AIX only) This protocol is used for sites that allow users to authenticate using their AIX (or LDAP) passwords on the HSI Gateway Server machines.

-c *cred_file*
> Specifies the credentials cache file to be used if Kerberos authentication is used.

-d *debug_level*
> Specifies the debugging level to enable when HSI starts up. The *debug_level* is an integer value between 0 and 4. Level 0 (the default) specifies no debugging, levels 1-4 specify successively higher debug levels. In general, the HSI debugging output is useful only to the developers for troubleshooting problems.

-e
> "Echo" flag. If enabled on the command line, or by the **echo** command, HSI will display command lines that are read from IN files.

-E *editor-style*
> Specifies the editing style to be used when editing command lines. The *editing-style* is either "vi" or "emacs". The default, if not specified in the .editrc file, is **-E vi**. Note that this option is effective only if HSI was built with line-editing enabled.

-G *globus_grid_proxy_path*
> If Globus GSI authentication is used for authentication, (for example, if **-A gsi** is specified), this option can be used to specify an alternative path to the proxy created by the grid-proxy-init command.

-g *GID | groupName*
> Specifies the group ID or group name to be used after login has completed. [HSI Version 3.5.4 or later]

-h *hostname*
> Directs HSI to attempt to connect to *hostname*, which can be either a name or an IP address.

-k *keytab_path*
> If keytab authentication is used for authentication, the path to the keytab file can be specified by this option. *Note*: the keytab file will not be used by HSI if it has other than <owner> permissions.

-l *login_name*
> This option can be used to specify the user's login name. The normal default is to use the local login name for the client host on which HSI is being run. The default login name can also be specified in the user's .hsirc file, or as a pattern in the global hsirc file for the client host.

-O *pathname*

> This option is intended for use by applications that run HSI via the **system** or **popen** calls. It causes HSI to:

> 1. Force one-liner mode (no input will be read interactively via stdin).
> 2. Write all output that would normally be written to the TTY or stderr to the file specified by pathname.

> The option takes effect immediately, so it should normally be the first option on the command line if it is used, in order to capture any messages related to parsing the command line. It also causes the "quiet" flag to be set (disabling extraneous messages from being displayed, such as the message-of-the-day), and the "verbose" command response flag to be disabled. In addition, it causes all interactive file-transfer progress messages to be disabled.

-o

> Overrides the <no-login> flag, if set by the HPSS administrator. This option can only be used by root. It is deprecated, and will be removed in a future release.

-P

> Popen command mode. This option is similar to the **-O** option, but causes all output to be directed to stdout, which is normally redirected to a process that starts HSI with the popen(3) system call. It also results in setting "quiet" (no extraneous messages) mode, disabling verbose response message, and disabling interactive file transfer messages.

-p *port*

Specifies the port number to be used when connecting to the HSI Gateway Server. This option is only effective if the **-h** *hostname* parameter is also specified.

-q

Enables "quiet" mode. In this mode, extraneous messages such as the interactive file transfer progress messages, as well as the startup "Message of the Day" are not displayed.

-s *site_name*

Specifies the global or private hsirc stanza name to be used. The *site_name* must match the name of one of the stanzas in either the global hsirc file, or the user's private $HOME/.hsirc file.

-V

Prints HSI version information without requiring a connection to the HSI Gateway Server. If used as the only command line option, HSI exits instead of entering interactive mode.

-v

Enables verbose mode for command output. Verbose mode is used for printing markers to the listable output file when performing operations such as recursive puts or gets.

Chapter 5. FTP compatibility and differences

HSI supports several of the commonly used FTP commands, including **dir**, **get**, **ls**, **mdelete**, **mget**, **put**, **mput**, and **prompt**, with the following differences:

- HSI transfers are always in "binary" mode.
- the "m" commands cause a prompt to be issued. Use the *Section 19.72, "PROMPT command"* toggle to disable or reenable prompting.
- The **dir** command is an alias for **ls** in HSI. The **ls** command supports an extensive set of options for displaying files, including wildcard pattern-matching, and the ability to recursively list a directory tree.
- The "put" and "get" family of commands support recursion.
- There are "conditional put" and "conditional get" commands (**cput** and **cget**).
- The syntax for renaming local files when storing files to HPSS or retrieving files from HPSS is different than FTP. With HSI, the syntax is always "local_file : HPSS_file", and multiple such pairs may be specified on a single command line.

With HSI, the local filename is specified first both on a **put** command and on a **get** command. For example, when using HSI to store the local file `file1` as HPSS file `hpss_file1`, then retrieve it back to the local filesystem as `file1.bak`, the following commands could be used: `put file1 : hpss_file1 get file1.bak : hpss_file1`

With FTP, the local filename is specified first on a **put** command, and second on a **get** command. For example, when using FTP to store the local file `file1` as HPSS file `hpss_file1`, then retrieve it back to the local filesystem as `file1.bak`, the following commands could be used: `put file1 hpss_file1 get file1.bak hpss_file1`

The "m" prefix is not needed for HSI commands; all commands that work with files accept multiple files on the command line. The "m" series of commands are intended to provide a measure of compatibility for FTP users.

Chapter 6. Path names

6.1. Syntax for renaming files in put and get commands

The syntax for renaming files in **put** or **get** commands is always:

local_file : *HPSS_file*

Whitespace is required before and after the colon ":" character.

Multiple such pairs of pathnames may be freely intermixed with other files that are not being renamed, for example:

```
put file1 : hpss_file1 file2 file3 : hpss_file3 file4 file5 file6
```

This would result in storing HPSS files `hpss_file1`, `file2`, `hpss_file3`, `file4`, `file5`, and `file6`.

Similarly:

```
get file1.bak : hpss_file1 file2.bak : hpss_file2 file3 : file3 file4 file5.bak : file5
file6
```

would result in retrieving local files `file1.bak`, `file2.bak`, `file3`, `file4`, `file5.bak`, and `file6`.

6.2. Using put and get commands with just one pathname

When the "local : hpss" rename syntax is *not* used to specify both a local path and an HPSS path on a **get** or **put** command, such as this one:

```
put /glade/user/someuser/nidas/2012/2012.tar.gz
```

HSI assumes that the path given is the HPSS pathname, and it uses the last component — in this case, "200201.tar.gz" — as the local pathname, which would be equivalent to:

```
put 200201.tar.gz : /glade/user/mizukami/nidas/2002/2002.tar.gz
```

6.3. Using UNIX PIPES

See *Chapter 9, Using PIPEs for input and output* for using PIPEs to read from or write to HPSS files.

See *Chapter 5, FTP compatibility and differences* for other differences between FTP and HSI commands.

See *Section 9.2, "Using UNIX PIPE commands for input and output"* for information on using UNIX pipe commands for storing or retrieving HPSS files.

Valid characters for a path parameter are ASCII characters greater than or equal to 0x20 (space) and less than or equal to 0x7E (tilde).

However, HSI uses the space character as a separator, so its use is discouraged; generally, pathnames containing spaces must be quoted. Also note that HPSS and HSI allow pathnames to contain non-printable characters. However, in HPSS the optional configuration setting, "Object names can contain unprintable characters" must be turned on. This option is normally disabled, so that HPSS also restricts pathnames to printable ASCII characters (0x20 - 0x7E).

Wildcard characters are asterisk (*), question mark (?), caret (^), open and close square brackets ([]), and open and close curly brackets ({}).

They may be used in all commands that reference either local or HPSS paths, including the "local : HPSS" form. Note that names containing wildcard characters must resolve to a single node for some commands, such as **set** DIR*n*=*pathname*, or **mv** *path path…npath*, in which *npath* must resolve to a single directory node.

See *Chapter 15, HSI special topics* for information on working with filenames that contain whitespace and wildcard characters.

6.4. UNIX-style pathname prefixes

The standard UNIX pathname prefix characters tile (~), period (.), and double period (..) are also allowed on all file name/HPSS-path name references.

Chapter 7. HSI Checksum Feature

HSI Versions 4.0.1.1 and later have the capability of storing checksums as part of the HPSS metadata for files, of verifying the checksums during file retrievals, and listing the checksum hashes.

7.1. Checksum hashing algorithms

HSI supports the following hashing algorithms:

- md5
- sha1
- sha224
- sha256
- sha384
- sha512
- crc32
- adler32

The default algorithm is defined when the HSI package is built, but can be overridden in the public and private hsirc files and via command line options on the **put** and **hashcreate** commands, as described below.

7.2. Transfer Rate Performance Impact

The checksum algorithms that are used are very CPU-intensive. Although the checksum code is compiled with a high level of compiler optimization, transfer rates can be significantly reduced when checksum creation or verification is in effect. The amount of degradation in transfer rates depends on several factors, such as processor speed, network transfer speed, and speed of the local filesystem.

7.3. New Commands

The following new commands were added as part of the checksum feature. The minimum abbreviation is followed by the remainder of the full command in square brackets.

hashcr[eate] Creates a checksum hash for existing HPSS files.

hashdel[ete] Deletes checksum hash for files.

hashli[st]   List the checksum hash for files.

lshash   This is an alias for the **hashlist** command.

hashver[ify]   Verifies checksum hash for files.

The usage syntax for all of these commands can be obtained interactively by issuing the command name by itself. For example: `? hashdel Usage: hashdel[ete] [−A] [−R] path … −A : display absolute pathname for files −R : [standard option]recursively delete hash entries for files in the specified path(s)`

7.4. New hsirc options

The following new options have been added to the global and private hsirc files:

cksum_enabled = **on|off**
> (default = off) Automatically enables (**on**) or disables (**off**) checksum creation for **put** commands and checksum verification for **get** commands.

cksum_type = *algorithm*
> Specifies the checksum hashing algorithm to use when creating new checksums for **put** or **hashcreate** commands. Valid checksum values are shown in *Section 7.1, "Checksum hashing algorithms"*, above. If not specified, the build-time default algorithm specified by the HSI administrator is used. The release default is `cksum_type = md5`.

These options can be specified in either the global or site stanza, or both, within the hsirc files. Site-stanza settings take precedence over global stanza settings, if both are specified. In addition, the private .hsirc file settings for these options take precedence over the host-global hsirc file in $HPSS_CFG_FILE_PATH/hsirc, if one is found on the HSI client machine during startup.

7.5. New Command Line Options for put and get commands

The following command line options have been added to the **put** and **get** commands:

-c on|off   Enables or disables checksum creation and verification for the **put** and **get** family of commands, overriding the build time option and hsirc setting.

-H   Specifies the checksum hashing algorithm to use (**put** command). For **get** commands, the hashing
algorithm   algorithm stored in HPSS metadata is used.

Chapter 8. Access Control Lists (ACLs)

HPSS Access Control Lists (ACLs) were based on the Access Control Lists for the Distributed Computing Environment (DCE). A good source of detailed information about DCE ACLs is at http://pubs.opengroup.org/onlinepubs/9656999/chap8.htm#tagcjh_09.

HSI added support for HPSS ACLs in Version 4.0.1.3. There are two new commands to support this feature:

chacl - creates, deletes and modifies ACL entries.

lsacl - lists ACL entries.

The purpose of ACLs is to define who can use an object, and what kinds of access are allowed for the object. All HPSS files and directories have ACLs that are mapped into the standard UNIX permissions (read, write, execute for owner, group, and other). In addition, there may be "Extended ACLS" that allow specific users, groups, or other principals in the local realm (DCE "cell") and in foreign realms to access an object.

As currently implemented in HPSS, adding, changing, or deleting an ACL requires either owner permissions or group write permission to the object.

An ACL contains one or more "ACL Entries", each of the form *type:key:perms*, where:

type   Is one of the following: user_obj, group_obj, other_obj, user, group, foreign_user, foreign_group, foreign_other, any_other, mask_obj, unauthenticated, user_obj_delegate, group_obj_delegate, other_obj_delegate, user_delegate, group_delegate, foreign_user_delegate, foreign_group_delegate foreign_other_delete, and any_other_delegate

       The most commonly used types in HPSS are user_obj, group_obj, other_obj, and the permissions for these types are mapped to the standard UNIX "owner", "group" and "other" permissions.

       mask_obj ACL entries are used to calculate the *effective permissions* for an object, similar to the umask that is used for UNIX permissions. The reference above contains some examples that illustrate the use of the mask_obj entry.

key   Is of the form *principal@realm*

       Different types of ACLs may not require the principal or the realm. Where a realm is required, it can be either a name, such as ibcg.gov or an ID such as 77233323.

       User and group *principals* can be specified as either a name, such as "rheinlein" or a numeric ID, such as 40149. Internally in HSI, principals are translated to binary values by means of the HPSS registry, but are displayed as names in the **lsacl** command.

perms Is a string representing the permissions, from the set "rwxcid". These are, respectively, "read", "write", "execute", "control", "insert" and "delete". In the **chacl** command, the permission string characters can be omitted or represented as a hyphen (-) in the permission string, when not wanted. For example, the following are equivalent: `r--c` and `rc`

       As an example, consider this command: `? lsacl * a_drive user_obj:user1:rwxc-- group_obj:user1:r-x--- other_obj:r-x--`

All HPSS objects contain an ACL that contains user_obj, group_obj and other_obj ACL entries. Directories also contain special "Initial Object" and "Initial Container" ACL entries that specify the default permissions to be placed on newly created objects (such as files) or subdirectories. These automatically propagate downward in the tree as new leafs and branches are added; this is called "ACL inheritance".

Here is an example of the object entry, and the Initial Object and Initial Container entries on a directory: `? ls - ld bpf drwxr-x--- 3 user1 user1 512 Mar 15 2011 bpf ? lsacl bpf bpf user_obj:user1:rwxcid`

```
group_obj:user1:r-x--- other_obj:------ ? lsacl -ic -io bpf [IC] bpf user_obj:user1:rwxcid
group_obj:user1:rwx-id other_obj:rwx-id [IO] bpf user_obj:user1:rwxc-- group_obj:user1:rwx---
other_obj:rwx---
```

Chapter 9. Using PIPEs for input and output

HSI supports the use of UNIX "pipes" when either storing or retrieving files, using the standard rename syntax:

get *local_File* : *HPSS_File* put *local_File* : *HPSS_File*

9.1. Using stdin and stdout for input/output

For **put** commands, if *localFile* is a minus sign ("-"), then HSI reads input from its stdin. (The transfer progress display must be turned off for this to work.) This is intended for use in shell script pipelines, such as: `tar cf -`
`. | hsi` *options* `"put - : someTarFile"`

Note

HTAR provides the same capability as the above, but is generally much faster.

However, if used interactively, HSI can also be used to read from the TTY to pipe a file into HPSS. In this case, a single *<ctrl-D>* (CTRL key and "d" key pressed simultaneously) can be used to terminate the input (take care not to use more than one *<ctrl-D>*, as this is also a way to exit HSI). For example: `0:[hpss-`
`nccs]/home/user1/temp->put - : ToDo-Today`

Here is my TO-DO list *(Lines entered from the TTY)*: wake up work eat go to bed *<ctrl-D>*

`put <stdin> : /home/user1/ToDo-Today ( 78 bytes, 0.0 KBS (cos=5081))`

For **get** commands, if *localFile* is a minus sign, then HSI pipes the output to its stdout. Again, this is intended for use in shell script pipelines, such as: `hsi` *options* `get - : someTarFile | tar xf -`

As explained in *[Chapter 15, HSI special topics](#)*, this is one way to view the contents of HPSS files without first copying them to a local file, for example, using the above HPSS file: `get - : ToDo-Today`

Here is my TO-DO list: wake up work eat go to bed

`get <stdout> : /home/user1/ToDo-Today (2010/12/20 19:47:38 78 bytes, 221.7 KBS )`

9.2. Using UNIX PIPE commands for input and output

For both storing and retrieving, if the first character of *localFile* is a vertical bar ("|"), HSI interprets the remainder of *localFile* as a pipe command after skipping any leading white space. The pipe command can contain one or more shell commands, such as: `put "| cat file1.c file2.c " : oneCFile` or `put "| ls -l`
`| awk '{print $9}' " : listOfFilenames` or `get "|tar xf - " : tools.tar` or `get "|less" :`
`bigDocument` *(see notes 3 and 4, below)*

Notes:

1. The *localFile* must be enclosed in quotes if it contains whitespace characters. In addition, if this form is used on a shell command line, and *localFile* contains whitespace characters, then the quoted *localFile* string must itself be enclosed in a separate set of quotes in order to ensure that HSI sees the quotes around

localFile when it parses the get or put options. Normally a single quote is used to quote the double-quoted *localFile* string.

For example, this will work: `hsi` *options* `put '"| ls -a"' : listingFile`

but this will fail: `hsi` *options* `put "| ls -a" : listingFile`

The reason is that in the first form, the shell parses the command line and passes the string `| ls -a` as a parameter to HSI, including the double quotes, so HSI sees the command parameters as:

    a. the string (including quotes) `| ls -a`
    b. the colon string (:)
    c. the string listingFile

In the second form, the shell removes the double quotes, and HSI sees the options as five strings:

    a. `|`
    b. ls
    c. -a
    d. :
    e. listingFile

In the second form, HSI interprets the first and second strings as local filenames, and the last three strings as the *local : HPSS* form, with local file -a and HPSS file.

2. To disable HSI file transfer progress messages, which can overwrite parts of the last line of output on the TTY, use either the **progress** command toggle, or the **-q** (quiet) option on the HSI command line.
3. When using interactive utilities such as "less", it is usually, although not always, necessary to use "exit" (or "end", "quit", or whatever the utility recognizes as a command to terminate) in order to exit the utility, even if HSI is able to pipe the entire file to the utility.

Important

HSI transfers entire files, using the parallel transfer features of HPSS in order to achieve the fastest possible transfer rates. When reading HPSS files to stdout, or to a local pipe command, it is possible to cause a transfer to abort if the file is longer than two data buffers (8 MB by default), and data is not consumed at a fast enough rate. This is normally not a problem for utilities such as " " or "md5", but could potentially be a problem for TTY-oriented utilities such as "less", "more", "tail", etc.

For information on how HSI chooses an HPSS Class of Service when reading from a UNIX pipe, see *Chapter 10, HSI COS selection with and without piped files*.

Chapter 10. HSI COS selection with and without piped files

This section describes the way that HSI chooses a Class of Service when storing files that are passed in via a pipe command.

As described in *Chapter 9, Using PIPEs for input and output* and *Chapter 15, HSI special topics*, sections, HSI supports the ability to read data from a pipe when writing to HPSS.

If the HPSS Class of Service has been set via the "**set cosid** = *cos_id*" command or as a keyword on the **put** command, then HSI causes that COS ID to be used when the file is written. However, many HPSS sites do not

reveal the Class of Service IDs that have been defined to their user community, and instead rely on HSI (and HTAR) to choose the correct COS based on information that is defined by the site's HPSS administrator. This is the case when automatic COS selection is in effect, which is the default when HSI is first started up.

The current class of service ID can be shown by using the **set** command with no options; for example: `0: [/home/user1]: set ... cosid = "auto" ...`

When automatic Class of Service selection is in effect, usually the COS is chosen based primarily on the source file size and number of copies desired. The HPSS administrator has the ability to set other criteria to restrict COSs to particular users, groups, or accounts, and to create sets of COSs called "named COS lists" that the user can specify via the **[coslist-key]** command, or list via the **lscos -n** command. In addition, starting with HSI Version 5.0.2.p11 (HPSS Version 7.4.3) and HSI Version 6.0.0.p11/6.2.0.p1 (HPSS Version 7.5.2/7.5.3), the user can specify a default COS to be used when HSI is first started. However, when the file size is unknown at the beginning of the transfer, HSI does one of the following:

- Does nothing and lets HPSS choose the initial COS, or
- Chooses an initial COS that the site's HPSS administrator has defined with the "default auto" setting in the server-side COS configuration file.

There are two cases that can occur at the end of the transfer when the file size was initially unknown:

- If the entire file fits within the first I/O buffer, before any data has been written to the file, HSI reselects the COS based on the amount of data in the buffer, then writes the data to the sink HPSS file using the new COS that was just selected. The amount of data that can fit in the buffer is determined by the default buffer size that was chosen when HSI was built. The release value is 8 MB (8,388,608 bytes), but this can be varied both at build time, or via the **[iobufsize-key]** keyword.

- If the file does not fit within a single I/O buffer, then after the transfer has successfully completed, using the COS that was initially chosen by HPSS or using the "default auto" COS, HSI does the following:

    - Reselects the COS based on the final sink file size in HPSS.
    - Internally issues a "Change Class of Service" to cause HPSS to move the file from the initial COS to the final COS.

The amount of time that it takes for HPSS to internally move the file from the initial COS to the final COS varies. There are a number of factors that influence the amount of time that this takes, such as number of Change-COS streams that are active, site I/O activity, size of the file, amount of free space on the final COS storage class, etc.

The number of possible Change-COS streams is defined by the site when HPSS is configured - there can currently be up to 32 active streams. For each Change COS issued internally by HSI for piped files, **stream 0** (the default) is used. Within each stream, the files are processed in FIFO order. See *Section 19.9, "CHCOS command"* for more information on this topic.

Chapter 11. Customizing the HSI prompt

The default HSI prompt string ("? " for the initial line, and "> " for continuation lines) may be customized by setting the **PS1** and **PS2** keywords, or by specifying the PS1 and PS2 settings in the global or private hsirc files. For example, setting the strings as follows: `PS1 = "%d[%H]%w3->"` `PS2 = "[%H]continue: "` would produce a prompt string similar to this for the initial prompt: `R:[hcdp01]/home/user1->` and a prompt similar to this for continuation lines: `[hcdp01]continue:`

Within a prompt string, characters are copied literally, except for substitutable parameters of the form *%x*, where *x* is one of the following:

C   DCE Cellname (currently recognized, but unused).

c   Current connection ID.

d   Current logical drive ID in the form A:.

D   System date in yyyy/mm/dd format.

h   Canonical hostname for current connection.

H   Remote HPSS hostname for current connection.

I   Remote HPSS IP address for current connection.

L   Current login principal.

s   Site name (non-DCE version).

N   Current command number (1:N).

T   System time in hh:mm:ss format.

W   Current HPSS working directory for connection, with optional number of directory paths specified by numeral following the %W.

w*n*   Current HPSS working directory, with "…" substitution in long directory paths. *n* is the number of directories to try to display on the right if the prompt is truncated to promptlen characters.

%   % character

Chapter 12. Using HSI with multiple HPSS systems

This section contains information on the following topics:

- Using HSI with multiple HPSS systems that are linked together via Junctions and Cross Cell authentication.
- Using HSI to access multiple HPSS systems.
- Opening connections to more than one HPSS system during a single HSI session.
- Description of the *Logical Drive* pathname syntax that HSI uses to associate files with a connection during a session.
- List of commands that are used to open, close, and switch between active sessions.
- Examples of storing, retrieving, and copying files while connected to multiple HPSS systems.

12.1. Using HSI to access multiple HPSS systems

HSI has the ability to establish concurrent connections to more than one HPSS system at more than one site (or to the same HPSS site) within a single session. After the sessions are opened, normal HSI commands can be used to work with any of the sessions, as described below. Files can be copied between the sites using one of the following mechanisms:

- either third-party transfers, which do not require the data to flow through HSI memory.
- First-party transfers, in cases where firewalls or other network connectivity problems prevent the movers from the source HPSS system from connecting to the HSI Gateway Server at the sink HPSS system.

The command to copy files between sites is described below. There are three requirements that must be met in order to use the Multi-HPSS feature:

1. You must have an account on the each of the HPSS systems that you wish to access.
2. There must be network connectivity between the client system on which HSI is executing and the HPSS sites that you wish to work with.
3. In order to transfer files using the 3rd-party mechanism, there must be network connectivity between the mover nodes on the source HPSS system, and the HSI Gateway server process on the sink HPSS system.

When working with multiple HPSS systems, HSI treats each connection as a "logical drive", borrowing from the familiar legacy of the early personal computers. In this environment, there are three key concepts:

1. The notion of a *current connection*. This is initially established when the first connection is opened as HSI begins executing. It can change during execution as connections to other HPSS systems are opened, and as you switch between logical drives. Each connection has its own separate *context*, containing items such as:

   - The address of the HPSS server host.
   - The logical drive letter that you (or HSI) assigned for the connection.
   - Your home current directory.
   - The current working directory.
   - The classes of service that are available on the HPSS system.

2. The use of *logical drive* notation as a command prefix, and to reference files and directories on a particular HPSS system. Logical drive notation simply consists of a non-case-sensitive *drive letter* followed by the colon (:) character (with no intervening whitespace). For example: `D:` followed by the command or pathname, with no intervening spaces. For example, **d:ls** would switch to connection D: and list the files in the current working directory.

   The command `ls C:*.c` would list all of the files ending with the characters ".c" on the HPSS system associated with logical drive C:.

   `get fromHPSS1 : D:file1 fromHPSS2 : F:file2` would fetch the file `file1` from the HPSS system associated with drive D:, renaming it to the local file `fromHPSS1`, and similarly would fetch the file `file2` from the HPSS associated with drive F:, renaming it locally to `fromHPSS2`.

Note that as of HSI Version 2.4, the colon character (":") used to separate the local and HPSS pathnames, must be surrounded by whitespace, unlike earlier versions of HSI. This requirement was added in order to avoid ambiguity in the inclusion of colon characters in pathnames. You can tell which HSI version you are using by typing: `version`

Drive letters are not case sensitive, so, the following are equivalent:

- `f:get myfile`
- `F:get myfile`

It is not necessary to use this notation; commands and pathnames without a drive prefix always reference the current connection. If the multi-HPSS feature of HSI is not used, commands and pathnames can be used just as in earlier versions of HSI.

Depending upon the context, the use of a standalone drive letter, such as `b:` or `ls C:` may be interpreted as either of the following:

- Set the *current connection* to the HPSS system associated with the logical drive (first example, above), or
- Reference the current working directory for the HPSS system associated with the logical drive (second example, above).

In commands where a filename is required, the use of a standalone drive letter will cause a usage error, for example: `get P:`

3. Third-party copies. When copying files between HPSS systems, the default is to use a third-party copy mechanism so that the data flows directly between HPSS systems, without having HSI in the middle of the transfer. See *Section 19.16, "CP and CCP commands"* command for more details. For some situations, such as copying files through a firewall, it may be necessary to use a store-and-forward mechanism to read data from either a local file or from an HPSS system inside the firewall, and then transfer it to an HPSS system outside the firewall. See *Section 19.30, "FIREWALL command"* for more information on setting up store-and-forward transfers.

12.2. Commands used for Multi-HPSS HSI Sessions

Commands used to access multiple HPSS systems are shown in the table below, and are also described in more detail in *Chapter 19, Command reference*. The minimum abbreviation for each command is shown in the text preceding the "[" character.

Command	Description
open or **connect**	Establishes a new connection to an HPSS system.
close	Closes an existing connection to an HPSS system.
lscon[nections] or **showc[on]**	Shows a list of currently established connections, including the home directory and the current working directory for the connection

lssites	Displays a list of site names that have been defined in the global .hsirc startup file, or in your private .hsirc file.
drive:	As a standalone command, this changes the current active connection.
drive:command	Changes the current active connection, then issues the specified command in the context of the new connection.
setdrive	Changes the logical drive letter assigned to a connection

12.3. Example scenario for a multi-HPSS session

A typical interactive scenario for using HSI to communicate with multiple HPSS systems is described below.

1. *Set the prompt string* so that we can tell which logical drive is our current active connection, and what our current working directory is within that connection. We want our prompt string to look like this:
   ```
   drive_letter:[hostname]working_directory->
   ```

   To accomplish this, the PS1 variable can be set in the hsirc file, or interactively as follows:

   ```
   PS1 = "%d[%H]%w3->"
   ```

   which will yield a prompt string similar to this:

   ```
   C:[hpss]/home/user1->
   ```

   The "%w3" will cause long working directory pathnames to be displayed by truncating to at most three subdirectories on the right-hand part of the prompt string, and replacing characters in the middle of the pathname with the string "…"; for example:

   ```
   S:[hpss05i]/home…/tools/scripts/backman->pwd … pwd0:
   /home/user1/HPSS/tools/scripts/backman
   ```

2. *Establish initial connection*

   If the site administrator has set up the global hsirc file, or you have customized your private .hsirc file in your home directory, then all that is required is to startup HSI with no options (to connect to the site-defined default HPSS system), or `hsi -s` *sitename* to connect to the site called *sitename*. Note that this is a case-sensitive option, which must exactly match the sitename contained in the hsirc stanza.

   If the **-site** option is not used on the command line, the following optional parameters may be used to connect to any HPSS system that you have an account on: `hsi -h` *hostname* `-p` *port* `-l` *login_name* `-A` *authentication method* `-k` *keytab filename*

   The format of the login name varies, depending on the authentication method that you have chosen. For the *combo* method (**-A** combo) and the keytab method (**-A** keytab), the login name is your principal name on the HPSS system to which you are connecting. At most sites, this will be the same as your login name on any other machines at that site, but check with the HPSS administrator at the site to make sure.

For Kerberos authentication (**-A** kerberos), the login name is normally your principal name and the Kerberos Realm name, which is generally in uppercase; for example: `-A kerberos -l` <u>yourname@REALMNAME</u>`.EDU`

Normally, the site administrator will set up an HSI wrapper script containing all the information regarding hostnames, ports, and authentication mechanism(s) to be used, so that the optional parameters shown below are usually not needed.

3. *Establish connection to another site*

As above, if the site administrator has set up a global hsirc file, or if you have customized your private .hsirc file to add the appropriate information, the command: `open -s` *new_sitename* or `open -d` *drive_letter:*

can be used to establish a connection to the HPSS system designated by *new_sitename* or *drive_letter:*.

The **lssites** command can be useful when running interactively, if you need help remembering the exact sitename(s) or drive letter(s) that have been set up by the HPSS administrators. For example, a user might have the following sites defined:

```
R:[hcdp01]/home/user1\->lssites
Site, Logical Drive....site1, "G:"
Site, Logical Drive....site2, "P:"
Site, Logical Drive....site3, "O:"
Site, Logical Drive....site4, "N:"
Site, Logical Drive....site5, "R:"
Site, Logical Drive....site6, "T:"
Site, Logical Drive....site7, "I:"
```

Note that in this next example, user1 is logged into HPSS at site5, and is going to open a connection to site7:

```
R: [hcdp01]/home/user1->cd HPSS/tools/scripts/backman
R:[hcdp01]/home/user1->open -d I: -A combo
Password: password
I:[hpss]/home/user1->
```

Display the list of active sites after opening the second connection:

```
R:[hpss]/home/user1->lscon
List of Currently Active Connections
Current default connection handle: 3
         Han-     Remote          Remote         HSM  Fire- I/O    Ctl   Data Port Cmd
Drive   dle    IP Address       Hostname      Level wall Mode   Port   Min  Max Count
 ----------------------------------------------------------------------------------
R:   1 <IPv4 addr8>   hcdp01.site8    6.2.1   off normal  1217      0 65535 22
HomeDir: /home/user1
pwd0: /home/user1/HPSS/tools/scripts/backman
->I:   3 <IPv4 addr7> hpssb1.site7    6.2.1   off normal  1218      0     0 13
HomeDir: /home/user1
pwd0: /home/user1


C:[hpss]/home/user1->
```

At this point, standard HSI commands can be used to list directories, store, retrieve, and copy files using the logical drive pathname syntax, etc.

In the next example, we want to store a copy of the local file `good_stuff` on one of the HPSS systems, and then replicate it on the other, in order to provide disaster recovery. We will do this by using the **cp**

command, so that the data flows directly between the two HPSS systems, without having HSI in the middle of the transfer.

First, store the file on logical drive R: in the directory `big_project`, which is an existing subdirectory directly underneath our home directory on both HPSS systems:

```
R:[hpss]/home/user1\->put good_stuff : s:~/big_project/good_stuff
put  good_stuff : /home/user1/temp/good_stuff ( 1083916 bytes, \
     397.5 KBS (cos=6))
```

Next, copy the file from the S: drive to the I: drive:

```
R:[hpss]/home/user1->cp s:~/temp/good_stuff i:~/temp/good_stuff
cp  R:/home/user1/temp/good_stuff to I:/home/user1/temp/good_stuff \
     (2001/09/16 23:36:18 1083916 bytes, 481.1 KBS (cos=12)
```

At this point, we are ready to close out the connection(s) and terminate the session(s). This can be done by simply exiting out of HSI. If we want to continue to work in one of the connections, but have no need to keep the connection open for the other, then the **close** command can be used; for example:

```
R:[hpss]/home/user1->close R:
I:[hpss05i]/home/user1->
```

Notice that when the current active connection is closed, HSI switches to one of the other active connections, in this case, the only other active connection was drive I:. If an attempt is made to close this connection, the following error message will be displayed:

```
I:[hpss05i]/home/user1/temp->close i:
*** Cannot use <close> for the only open connection.
 Use  (or one of its aliases) to terminate HSI
```

4. *Logging HSI sessions*

When HSI execution has terminated, a logfile containing a time-stamped record of all commands will be present in the hsilog file, which is normally written in the home directory. This can be controlled by using the HSI **log** command, or by setting the HSI_LOGFILE environment variable as follows:

If using CSH: setenv HSI_LOGFILE *somepath*

If using KSH export HSI_LOGFILE=*somepath*

If using SH (this will also work for KSH and BASH): export HSI_LOGFILE HSI_LOGFILE=*somepath*

somepath can be the reserved string none to disable automatic creation of a logfile. It can also contain metacharacters of the form *%X* as follows:

%H Expanded to your local home directory pathname.

%N Expanded to the non-canonical local hostname; that is, the hostname without any domain name. For example, "snuffles.site8" would become "snuffles".

%P Expanded to the current HSI process ID.

%U Expanded to the local login name.

The default setting is usually determined by the HPSS administrator, and set in the HSI wrapper script. If the path already exists when HSI attempts to open the local logfile, HSI will try to append to it.

Chapter 13. HSI environment variables

HSI and its support libraries use several environment variables to control the program behavior. The method of setting these variables depends upon the shell that is in effect:

For *csh,* environment variables are set by using the syntax: `setenv` *environment_variable=value*

For **sh**, **ksh**, or **bash**, environment variables can be set by using the syntax: `export` *environment_variable* *environment_variable=value*

For **ksh** or **bash**, the two lines can be combined into one: `export` *environment_variable=value*

The following system or HPSS-defined environment variables are currently used by the HSI package:

ENVIRONMENT
> This variable is checked to determine whether HSI is being run in a batch environment. If set to "BATCH", HSI assumes that the job is not being run interactively.

HOME
> This is the home directory for the local logged-in user.

KRB5CCNAME
> If using Kerberos authentication (see *Chapter 2 "Authentication methods")*, this specifies where the credentials cache file should be written or looked for.

The following environment variables are defined for the HSI package:

HPSS_EGD_PATH
> This is the path to the Entropy Gathering Daemon, used for OpenSSL calls.

HPSS_HOSTNAME
> This variable is used when determining which network interface to use for connections from the HPSS movers. It can be either a host name or an IP address. It can be overridden by the HPSS.conf "PFTP Client Interfaces" stanza. If not specified, then HSI uses the interface associated with the hostname command.

HSI_NOLOGIN_FILE
> This is checked by HSI during startup to determine whether the site admins have disabled access to HPSS.

HPSS_PFTPC_PORT_RANGE=ncacn_ip_tcp[*start-end*]

> This variable is used to define a range of restricted ports that HSI will use when opening sockets for inbound connections from movers, or for authorization protocols such as Kerberos.

> **Note**

> In HSI Version 5.0.0 and later the new and preferred syntax for a restricted port range is: HPSS_PORT_RANGE=*start_port-end_port*. The older form is deprecated, but will continue to be supported for now.

HPSS_USE_NETRC_FILE

If this variable is defined, HSI will attempt to read a .netrc file to obtain the password to be used for either the *local* (**-A local** command line option) or *combo* (**-A combo** command line option) authentication method.

HPSS_API_DEBUG=*debug_mask*

(HSI 3.4.3 and later) If this environment variable is set, *debug_mask* defines the logging bitmask for the HSI Gateway Library, which corresponds to the priority levels defined for syslog(1). This value is an integer value between 0 and 255, and may be specified as a normal integer, an octal value (with a leading 0), or a hexadecimal value (with a leading *0x*). The default value is 255 (0xFF). Bit positions are as follows:

2^0 [LOG_EMERG] - system is unusable 2^1 [LOG_ALERT] - action must be taken immediately 2^2 [LOG_CRIT] - critical conditions 2^3 [LOG_ERR] - error conditions 2^4 [LOG_WARNING] - warning conditions 2^5 [LOG_NOTICE] - normal but significant condition 2^6 [LOG_INFO] - informational 2^7 [LOG_DEBUG] - debug-level messages

This environment variable must be set in order to enable any debug logging. Normally, this variable will be set and the default (255) will be used for HPSS_API_DEBUG.

HPSS_API_DEBUG_PATH=path
If set, contains the pathname used for the debug file. This file is appended to unless the "overwrite" option is set in HPSS_API_DEBUG_FLAGS. If the pathname is "stdout" or "stderr", the log is written to stdout or stderr.

HPSS_API_DEBUG_FLAGS

If set, contains a list of one-character flags used to control the logging as follows:

w If set, the log file is truncated to zero length or created for writing.

a (default) If set, the log file is created if it does not exist, or positioned to the end of the file, and all writes occur at the end of the file.

f If set, causes the log file to be flushed after each call. Note that this happens automatically if writing to stderr or stdout.

m Trace inbound and outbound messages.

p If set, causes the PID to be included in message prefix.

r If set, cause realtime (hi-res) time to be included in the message prefix.

R If set, disable hi-res time in the message prefix.

t   (default) If set, causes time from ctime (**time()**) to be included in the message prefix.

T   If set, disables "t" flag.

- Unrecognized flags are silently ignored.
- If there are duplicate or conflicting flags, the last one encountered is used.

HSI_HISTFILE
   When HSI is built with command-line editing enabled, this variable is used to specify the local file where command history is saved and restored in future sessions.
HSI_HISTSIZE
   When HSI is built with command-line editing enabled, this variable is used to override the default number of command lines that are saved and displayed with the history command.

Chapter 14. HPSS.conf settings (client)

The following HPSS.conf settings are used by the HSI and HTAR clients:

PFTP Client Interfaces stanza:

Network Options stanza:

Restricted Addresses stanza:

This top-level stanza in the client-side HPSS.conf has the following form:

Restricted Addresses = {

hostname or *hostpattern* … = {

IPV4 or IPV6 address[/*mask*]

IPV4 or IPV6 address[/*mask*]

…

}

hostname or *hostpattern* … = {

IPV4 or IPV6 address[/*mask*]

IPV4 or IPV6 address[/*mask*]

…

}

…

}

hostname: the client hostname on which HSI/HTAR is executing. This can be a wildcard pattern, and multiple patterns can be specified. If the "gethostbyname" call returns a fully qualified domain name, then the domain name must also be matched in the hostname or hostpattern.

IPV4 addresses have the CIDR form: *x.x.x.x*[/*mask*] or x.x.x[/*mask*]

IPV6 addresses have the CIDR form: x:x:x:x[/*mask*]

x : octet

mask: number of significant bits in the address

Chapter 15. HSI special topics

This page describes some of the ways in which HSI has been used to perform non-obvious functions.

15.1. How to pipe an HPSS file into "cmp", "head", "less", etc.

It is often desirable to retrieve a file from HPSS immediately after storing it, in order to compare the HPSS file with the original local file, or to view the contents of an HPSS file using a tool such as "less". The most obvious way to do this is to retrieve the HPSS file to a local file, and then use the standard operating system utilities to operate on the local file.

However, HSI provides the ability to pipe the output of a **get** command into a pipeline of one or more operating system utilities without having to write the file back to the local filesystem. This can help to avoid problems with disk quotas, insufficient local filesystem space, non-writable filesystems, etc. and can also be significantly faster, depending on the filesystem type, than copying the file to a local filesystem and then operating on it.

Pipe syntax is described in *Chapter 9, Using PIPEs for input and output*.

Class of Service selection for Piped input files is described in the *Chapter 10, HSI COS selection with and without piped files*.

Here's an example of using a pipe interactively to run the "cmp" utility to compare a local file with an HPSS file:
`put my_local_file : my_HPSS_file get "|cmp – my_local_file" : my_HPSS_file`

In this example, `|cmp – my_local_file` is the local file to which the HPSS file called `my_HPSS_file` is to be renamed, using the `local : HPSS` rename syntax. Since the local filename contains spaces, it must be enclosed in quotes. In addition, since the local filename starts with "|", HSI interprets the remainder of the local filename as a pipe command, and forks a shell to receive the output of the command. Note that if this command form is used in a shell command (that is, in a script), the double quoted local file string must itself be enclosed in single quotes, as explained in the notes in *Chapter 9, Using PIPEs for input and output*.

In this case, the "cmp" command interprets the first argument, "-", as standard input. Other utilities that read from standard input may use a different parameter to represent standard input, or may simply read from standard input if no filename arguments are provided.

The **-q** command line option, or the **progress** command toggle, can be used to disable file transfer progress messages that can interfere with the output of utilities such as "head".

Note

It is necessary to type "exit" when piping output into some utilities, such as "head", in order to exit from the utility back to the HSI command prompt, even if the entire HPSS file is piped into the utility in less than one full screen.

15.2. How to retrieve a set of files in optimal tape/position order

By default, the **get** command will retrieve the specified list of files in tape and position order. Also note that it may be necessary to use the **-A** option if auto-scheduling has been disabled at the site. However, this only applies to a single **get** command; if multiple **get** commands are issued, each command is independent of any others, and only the list of files specified by the single **get** command will be optimized for retrieval.

The easiest way to optimize a single **get** command is to use the "here-document" form of the command, by creating an IN file that looks like this:

```
get << EOF
file1
file2
file3
...
EOF
```

15.3. Working with pathnames containing special characters

A common problem is working with filenames containing spaces and special characters that are normally used for pattern-matching. An example of this is shown below:

```
Run-202 Future Trends in Automobile Design.pdf
```

Pathnames that contain whitespace characters can be enclosed in single or double quotes, or the whitespace characters can be escaped by backslashes. For example:

```
ls "Run-202 Future Trends in Automobile Design.pdf"
```

or

```
ls Run-202\ Future\ Trends\ in\ Automobile\ Design.pdf
```

A more difficult situation is a filename that contains characters that are normally used for filename globbing. For example:

```
Robotics[102]: Design and Development
```

This pathname contains both spaces and square brackets, which are normally used as pattern-matching characters. To avoid pattern-matching by the shell (ksh, csh, …), filenames can be enclosed in single quotes; however, using single quotes for this pathname does not work as it would with the shell, because HSI's command parser does not currently distinguish between single and double quotes. (Note: this will be corrected in a future version).

The simplest way to deal with filenames that contain wildcard characters is to disable filename globbing by issuing the **glob** command. This command is a toggle - each time it is issued, its state switches between "enabled" and "disabled". The command may be issued any number of times within a single HSI session. The initial state when HSI starts up is "enabled". When filename globbing is disabled, HSI does not check for wildcard characters when it is matching filename patterns; instead, filenames are matched literally, character-for-character. For the example above, the following could be used:

1. disable filename globbing: `glob`

2. Issue command(s). Note that quotes are still needed because of whitespace characters in the HPSS pathname. `ls -l "Robotics[102]: Design and Development" get roboticsCourse : "Robotics[102]: Design and Development"`
3. Re-enable filename globbing: `glob`

Chapter 16. HSI keywords

The following keywords are available in almost every HSI command. They are set by one of the following mechanisms:

1. Setting a value for the session: **set** *keyword=value* command. Note: The **set** command string can optionally be omitted, as HSI assumes any command of the form *keyword = value* is a **set** command.
2. Setting a value for the current command: *command keyword=value* [*options*] …

Keywords, and their minimum abbreviation, are shown in the table below.

Keyword	Function
ba[ckup] = on | off	Automatically renames existing file on get/put. Default is "on".
col[umns] = *numeric value*	Specifies number of columns per line. This value is used by the **ls** command when determining the number of entries per line to list. It is changed whenever the screen size changes.
copies = *n*	Number of copies to store. The default is set by the HPSS administrator in the server configuration file.
cos[id] = **auto** | *cosid*	Class of services to use. Default is "auto", which automatically selects the class of service, based upon file size, etc. as determined by the HPSS administrator. Use the **lscos** command to see the classes of service defined for the site; be aware that these are highly site-dependent.
coslist = *name*	Class of Service list to use. Default is none. Named Classes of Services are groups of COSs that are defined by the HPSS administrator. Use **lscos -n** to display a list of named COSs for the current active connection.
dcreate[mode] = *octal_value*	Octal mode to use when creating directories. This value is defined in the global or private hsirc file's dcreate_mode setting, or to 0777 by default. The current umask value is used in conjunction with the mode setting in order to yield the effective permissions that are set.

Keyword	Function
dir*n* = *pathname*	*n* is omitted (meaning dir0) or it is 0 to 9). The default pathname is your HPSS home directory. A keyword of **dir0** or **dir** assigns the current working directory. **dir*n*** sets up a shorthand name for the directory as *n*', which can then be used in place of the directory, for example: dir1 = /home/users/jane/projects ls 1' which is equivalent to: ls /home/users/jane/projects
family[id] = *numeric_value*	Set default file family ID to be used when creating new files. The default is either 0, or the value defined in the global or private .hsirc file "family_id" setting.
fcreate[mode] = *octal_value*	Octal mode to use when creating files. This value is defined in the global or private hsirc file's "fcreate_mode" setting, or to 0777 by default. The current umask value is used in conjunction with the mode setting in order to yield the effective permissions that are set.
iob[ufsize] = *numeric_value*	I/O buffersize to use when transferring files. The default is set by the HPSS administrator when HSI is built, normally 8 MB. The *numeric_value* can optionally be followed by any of the following multipliers, with no intervening whitespace: "k", "kb", "m", "mb", "g", "gb", "t", "tb", "p", "pb" for kilobytes/megabytes/gigabytes/terabytes/petabytes. The maximum buffersize is normally 32 MB, but is defined at build time by the HPSS administrator.
lines = *numeric_value*	Sets the number of lines per page for the terminal. This value was used for a curses-based help package for a previous version of HSI, but is currently unused.
promptl[en] = *numeric_value*	Sets the prompt length in characters.
promptd[irlen] = *numeric_value*	Sets the maximum number of characters to use when expanding the \fI%w\fP metacharacter in the command prompt. This is the number of directory components to include from the right-hand side of the pathname if the prompt must be truncated to **promptlen** characters.
PS1 = *string*	Specifies the prompt string to use for initial command prompting. See the "Prompt String" section for details on the options for *string*. The default setting for this value is "?" or the value of the PS1 setting in the global or private hsirc file.

Keyword	Function
PS2 = *string*	Specifies the prompt string to use for continuation lines. See the "Prompt String" section for details on the options for *string*. The default setting for this value is ">" or the value of the PS2 setting in the global or private hsirc file.
pwid[**th**] = *numeric_value*	Sets the parallel stripe width to use when transferring files. HSI automatically determines the optimum stripe width to use, so this option is currently unused, and may be removed in a future release.

Chapter 17. HSI commands (alphabetical list)

Command	Minimum Abbreviation	Notes
add	**a**	Alias for "mkdir".
abort	**ab**	Abort or cancel current active file transfer.
adopt	**ado**	Replace Active Keyset values with a saved Keyset.
annotate	**anno**	Set annotation for a namespace object.
bell	**bell**	Toggle sounding a bell when interactive input is needed.
bye	**bye**	Alias for "end".
copy	**c**	Alias for "cp".
cd	**cd**	**cd[0..9]** : Change HPSS default directory.
cdls	**cdl**	Equivalent to "cd" followed by "ls".
cget	**cget**	Conditional get.

Command	Minimum Abbreviation	Notes
chacct	**chacc**	Change account ID for one or more files.
chacl	**chacl**	Create, update, or delete Access Control List entries.
chcos	**chc**	Change class-of-service for one or more files.
chgrp	**chgrp**	Change group ID for HPSS objects.
chmod	**chmod**	Change permissions for HPSS objects.
chown	**chown**	Change owner ID for HPSS objects.
close	**close**	Close HPSS server connection.
connect	**conn**	Alias for "open".
cp	**cp**	Copy one or more existing HPSS files to another HPSS file.
cput	**cput**	Conditionally put files into HPSS.
crename	**cren**	Conditionally rename and optionally delete files.
debug	**debug**	Set debug level.
delete	**del**	Alias for "rm".
dir	**dir**	Alias for "ls".
du	**du**	Tally kilobytes in files or directories.

Command	Minimum Abbreviation	Notes
dump	**dump**	List object Nameserver/BFS metadata.
echo	**echo**	Toggle the display of command input from IN files.
end	**end**	Terminate execution.
erase	**erase**	Alias for "rm".
exit	**ex**	Alias for "end".
expdelete	**expdel**	Deletes expiration time for files.
expfind	**expfind**	Find files whose expiration time matches the specified criteria.
expls	**expls**	List expiration time for files.
expset	**expset**	Set expiration time for files.
find	**find**	Find objects that match specific criteria.
firewall	**fire**	Set store-and-forward transfer mode thru firewall.
free	**fr**	Delete a saved Keyset from HPSS.
get	**g**	Copy a file from HPSS to the client.
give	**give**	Give one or more files to another user.
glob	**glob**	Toggle on/off filename wildcard pattern-matching.

Command	Minimum Abbreviation	Notes
groups	**gro**	List group membership.
help	**help**	Explain HSI commands.
history	**hist**	Display or edit command history.
hashcreate	**hashcr**	Create checksum hash for one or more HPSS files.
hashdelete	**hashdel**	Delete checksum hash for one or more HPSS files.
hashlist	**hashli**	List checksum hash for one or more HPSS files.
hashverify	**hashver**	Verify checksum hash for one or more HPSS files.
id	**id**	Display current login identity info.
idletime	**idle**	Set interactive session timeout value.
in	**in**	Read commands from a file.
keep	**k**	Store current Active Keyset values into a saved Keyset.
list	**l**	Alias for "ls".
lcd	**lcd**	Local "change directory".
lcdls	**lcdl**	"cd" followed by "ls" command.
lls	**lls**	Local "ls" command.

Command	Minimum Abbreviation	Notes
lmd	**lmd**	Alias for "lmkdir".
lmkdir	**lmkd**	Make a local directory.
ln	**ln**	Create symbolic or hard link.
log	**log**	Write HSI log output to a file.
lpwd	**lpwd**	Local "print working directory".
ls	**ls**	List HPSS objects.
lsconnections	**lscon**	Display active HPSS server connections.
lsacl	**lsacl**	List Access Control List (ACL) entries.
lscos	**lscos**	List Class of Service entries.
lsexp	**lsexp**	Alias for "expls".
lsfset	**lsfset**	Alias for "lsfilesets".
lsfilesets	**lsfiles**	List HPSS filesets.
lshash	**lshash**	Alias for "hashlist".
lsjunctions	**lsjunc**	List nameserver junctions.
lssites	**lssi**	List site entries from one or more hsirc files.

Command	Minimum Abbreviation	Notes
md	**md**	Alias for "mkdir".
mdelete	**mdel**	Delete one or more files with interactive override.
mget	**mget**	Get one or more files with interactive override.
migrate	**mig**	Migrate files from disk cache to tape.
mkdir	**mkd**	Make one or more HPSS directories.
move	**move**	Alias for "mv".
mput	**mput**	Put files to HPSS with interactive override.
mv	**mv**	Move objects within the HPSS directory structure.
newacct	**newa**	Set account ID to use for current session.
newgrp	**newgrp**	Set new current group for session [HSI Version 3.5.4]
open	**open**	Open a new HPSS server connection.
out	**out**	Write HSI listing messages to a file.
put	**p**	Put a file into HPSS, replace any existing file.
plock	**plo**	Lock one or more files in disk cache.
punlock	**punl**	Unlock files that were previously locked in disk cache.

Command	Minimum Abbreviation	Notes
progress	**progress**	File transfer progress display on/off toggle.
prompt	**prompt**	Toggle on/off for prompting for "mget/mput/mdel" commands.
purge	**purge**	Purge one or more files from HPSS disk cache.
pwd	**pwd**	Print current HPSS working directory.
quit	**q**	Alias for "end".
r	**r**	Alias for "hist -s" command.
rd	**rd**	Alias for "rmdir".
replace	**rep**	Alias for "put".
recv	**rec**	Alias for "get".
remove	**rem**	Alias for "rmdir".
rename	**ren**	Rename HPSS objects.
rm	**rm**	Remove files.
rmdir	**rmdir**	Remove directories.
runique	**runique**	Toggle creating local files with unique names.
save	**s**	Alias for "cput".

Command	Minimum Abbreviation	Notes
send	**send**	Alias for "put".
set	**set**	Set or list variables in the active keyset.
setcon	**setc**	Set current active HPSS server connection.
setdrive	**setdr**	Set logical drive letter for an HPSS connection.
show	**show**	Show keysets.
showcon	**showc**	Alias for "lsconnections".
store	**st**	Alias for "put".
stage	**sta**	Stage one or more files onto HPSS disk cache.
su	**su**	Assume the identity of another HPSS user.
sudo	**sudo**	Issue commands as another HPSS user.
sunique	**sunique**	Toggle creating HPSS files with unique names.
switch	**swi**	Alias for "setcon".
touch	**tou**	Update access times for, or create, HPSS objects.
umask	**umask**	Set file creation mask for remote files.
verbose	**verbose**	Toggle "verbose" mode.

Command	Minimum Abbreviation	Notes
version	**vers**	Show HSI version info.
whoami	**whoami**	Display current HPSS user identity.

Chapter 18. HSI commands (by function)

The following reference tables for HSI commands are arranged by general function.

18.1. HPSS File and Directory Commands

Command	Function
cd	Change current directory.
cdls	Change current directory and list contents.
chacct	Change account code or name for a file.
chgrp	Change group ownership of a file or directory.
chmod	Change permissions of file or directory.
chown	Change ownership of a file or directory.
cp	Copy a file within HPSS.
get cget mget	Copy one or more HPSS-resident files to local files.
erase	Alias for "rm".
find	Search for and list namespace objects matching selection criteria.

Command	Function
give	Give one or more files to another user.
ln	Create symbolic or hard link from one HPSS object to another.
ls	List namespace objects.
mdelete	Delete files with optional interactive override. See "rm".
mget	Retrieve HPSS files with optional interactive override. See "get".
mkdir	Create an HPSS directory.
mput	Copy one or more local files to HPSS with optional interactive override. See "put".
mv	Rename an HPSS file.
put cput mput	Copy one or more local files to HPSS.
pwd	Print current directory.
rm	Remove one or more files from HPSS.
rename	Rename an HPSS node.
rd	Alias for "rmdir".
rmdir	Delete an HPSS directory.
touch	Create an empty HPSS file or update an HPSS file timestamp.

18.2. Local File and Directory Commands

Command	Function
!command	Issue shell command.
in	Read commands from a local file.
lcd	Change local directory.
lcdls	Change local directory and list contents.
lls	List local directory.
lmd	Alias for "lmkdir".
lmkdir	Make a local directory.
log	Write all HSI commands and responses to a local log file.
lpwd	Print current local directory.
out	Write HSI output to a local file.

18.3. Access Control List (ACL) Commands

Command	Function
chacl	Create, update, or delete Access Control List entries.
lsacl	List Access Control List entries.

18.4. Accounting/Metadata Commands

Command	Function
annotate	Add commentary to HPSS metadata for a file or directory.
chacct	Change the account ID of a file or directory.
chcos	Change Class of Service for an HPSS file.
crename	Conditionally rename an existing HPSS file.
du	Summarize space usage.
dump	Display nameserver and bitfiles server metadata for a namespace object.
expdelete	Delete the file expiration time for a file.
expfind	Find files whose expiration time matches specified criteria.
expls	Display expiration time for files.
expset	Set expiration time for HPSS files.
groups	Display HPSS group membership.
hashcreate	Create checksum hashing information for an existing HPSS file.
hashdelete	Remove checksum hash information from HPSS metadata.
hashlist	Display checksum hashing information for an HPSS file.
hashls	Alias for "hashlist".
hashverify	Verify checksum hashing information for an HPSS file.

Command	Function
id	Display current login or su/sudo identity.
lscos	Display contents of server-side Class of Service file.
lsfset	Alias for "lsfilesets".
lsfilesets	List filesets that have been created by the HPSS administrator.
lshash	Alias for "hashlist".
lsjunctions	List junctions and their related filesets that have been created by the HPSS administrator.
newacct	Change active account ID or account name.
newgrp	Change active group ID or group name.
su	Assume another user's HPSS identity.
sudo	Issue HSI commands using another user's HPSS identity.
umask	Set file creation mask.
whoami	Display current login or su/sudo identity.

18.5. File and Directory Administration Commands

Command	Function
migrate	Copy HPSS files from lower to higher levels of a hierarchy.
plock	Make files non-purgeable from disk cache for HPSS-admin specified time.

Command	Function
punlock	Make files purgeable from disk cache.
purge	Delete data from level 0 of a hierarchy.
stage	Copy files to level 0 of a hierarchy.

18.6. Multi-HPSS Commands

Command	Function
close	Close a server connection.
connect	Alias for "open" command.
lsconnections	Display a list of active server connections.
lssites	Display a list of sites parsed from the hsirc file(s).
open	Open a new server connection.
setconnection	Alias for "switch" command.
setdrive	Change the logical drive letter associated with a connection.
switch	Switch active server connection. Note that this implicitly happens if the logical drive letter is specified in a standalone command, such as B:.

18.7. HSI Settings Commands and Keywords

Command	Function
set *key=value*	Set value for an internal HSI variable. **Note:** any command of the form *key=value* is treated as a set command.

Command	Function
adopt	Replace current set of keywords with those from a named keyset.
free	Delete a saved keyset.
keep	Save current set of keyword values.
show	Display saved keysets.

Keyword	Value
backup	Enable (**on**) or disable (**off**) renaming existing files on **get**/**put** commands.
columns	A numeric value to specify the number of columns per line. Default = "auto-adjust".
copies	Set number of copies (normally either 1 or 2) to use when selecting a Class of Service.
cosid	Set Class of Service ID to request for **put** and **cp** commands.
coslist	Set Class of Service "named cos list" to use when selecting a Class of Service.
dcreatemode	An octal value specifying permissions to set (combined with **umask**) when creating a directory.
dir*n*	Set working directory 0 - 9 to *pathname*.
familyid	Set numeric file family ID to request when creating new files.
fcreatemode	An octal mode to use, combined with **umask**, when creating new files.
iobufsize	Buffer size to use when transferring files.

Keyword	Value
lines	Number of lines per page to use for the terminal.
promptlen	Maximum number of characters to use for interactive prompt.
promptdirlen	Maximum number of characters to use for directory components of path for prompt.
PS1	Prompt string to use for interactive input prompt.
PS2	Prompt string to use for continuation line input prompt.
pwidth	Parallel stripe width to use when transferring files.

18.8. Miscellaneous HSI Commands

Command	Function
bell	Toggle for sending the BEL (\007) character to the prompt for input.
echo	Toggle for echoing command input.
firewall	Enable/disable use of store-and-forward I/O through a firewall.
glob	On/off toggle for enabling/disabling pattern matching for filenames.
help	Display help information for HSI commands.
history	Display command line history.
idletime	Set idle timeout value.
progress	Toggle for enabling/disabling file transfer progress display.

Command	Function
quit exit end	Terminate HSI.
prompt	Toggles HSI prompting for **cget**, **mget**, **mput**, and **mdelete**.
runique	[future] Toggle for creating local files with unique names on for **get** command.
sunique	[future] Toggle for creating remote files with unique names on for **put** command.
verbose	Toggle for enabling/disabling verbose output mode.
version	Display HSI version and build information.

Chapter 19. Command reference

This chapter describes each HSI command, its parameters and options, and examples and notes about its use.

19.1. ADOPT command

Synopsis
>   **ado**[**pt**] [*keywords*] *keyset*

Description
>   This command replaces the current set of keyword values with those from the named keyset.

HSI Version
>   All

Aliases
>   None

Options

>   **-l**   Adopt local keyset

>   **-h**   Adopt HPSS keyset

Keywords
>   None

Usage Notes
>   If the keyset name is a minus sign (-), then the system default keyset is used.

Examples
>   adopt myFavoriteKeyset

Related Commands

19.2. ANNOTATE command

Synopsis
> **anno**[**tate**] [**-R**] [**-e**] [**-d**] [**-f**] [**-A** "string"] *path* …

Description
> This command adds annotation text to existing files or directories.

HSI Version
> All

Aliases
> None

Options

-A	Annotation string (including metacharacters). An empty string is treated as if the **-e** option was specified
-R	Recursively traverse directories in the specified path(s)
-e	Erase any existing annotation
-d	Operate only on directory objects
-f	Operate only on file objects

Keywords
> None

Usage Notes

1. The annotation string may be a maximum of 255 characters. It is silently truncated during expansion if it exceeds this length.

2. The annotation string may contain backslash-escaped characters which are replaced with their normal ASCII equivalents as follows:

   - \a alert → bell character
   - \n newline
   - \f formfeed
   - \r carriage return
   - \t horizontal tab
   - \v vertical tab
   - \\ backslash
   - \? question mark

- \' single quote

- \" double quote

  Note that two consecutive backslash characters are required in order to enter one of the above, since the HSI parser treats the first backslash as a "quote" character while initially scanning the command. The second backslash and the character following are then evaluated by the code within the annotate command.

Examples

1. Add annotation to the file "test.results" and then list the annotation: `annotate –A "Results of test \\nRun on 09/01/2001" test.results ls –A test.results`
2. Remove annotation from all directory objects in the directory tree "old.experiment": `annotate –eRd old.experiment`

Related Commands

    Section 19.53, "LS command" *(-A option)*

    Section 19.63, "MKDIR command" *(-A option)*

    Section 19.75, "PUT command" *(-A option)*

19.3. BELL command

Synopsis
    `bell`
Description
    This command is the toggle for sending the BEL (007) character to prompt for input.
HSI Version
    All
Aliases
    None
Options
    None
Keywords
    None
Usage Notes
    The BEL character is not sent unless stderr is connected to a terminal. It is only sent when input from the terminal is required, not after every file transfer.
Examples
    `bell`
Related Commands
    None

19.4. CD command

Synopsis

    cd[0..9] [**drive:**] *path*

    cd -?

Description

This command changes working directory DIR0-9.

HSI Version

    All

Aliases

    None

Options

> **0..9**   Working directory number. If the directory number is not specified, then "0" is used.

> **drive:**   Specifies the logical drive connection to be used. Note that although the path for the connection is changed, this command does not change the active connection. Use the **drive** (or **connection** or **switch**) command to change the active connection.

> **-?**   Print usage message.

Keywords

    DIR

Usage Notes

1. **cd** with no parameters is semantically equal to **cd0 ~**. The following are exactly equivalent:
   `cd0  cd  cd ~`
2. The user must have execute permission for the target directory (and all component directories in the full path), or the command will fail.
3. HSI extends the standard UNIX notion of working directory to provide 10 directory paths, each of which can contain a separate pathname.
4. A shorthand notation (called prime notation) is used to reference each working directory, using the form: n'> where n is a number between 0 and 9. Directory 0 (0') always refers to the current working directory.
5. Wildcards may be used to pattern match the directory name, so long as the pattern matches exactly one name.

Examples

1. Change to the "Projects" directory within the user's home directory: `cd ~/Projects`
2. Another way to do the same thing, using prime notation. The **set** command changes the value of working directory 1. The **cd** then uses the contents of working directory 1 to change the contents of working directory 0, which is the default working directory. `set d1=~Projects;cd 1'`

Related Commands

[**Section 19.5, "CDLS command"**](#)

[**Section 19.46, "LCD command"**](#)

19.5. CDLS command

Synopsis

    cdl[**s**] [*options*] [*path …*]

Description

This command executes the **cd** command, followed by the **ls** command, to combine the operations of changing to a new directory and listing it into a single command.

HSI Version
>   All

Aliases
>   None

Options

>   **-a**   List all entries, including "hidden" files whose names begin with ".".

>   **-c**   Use time of last modification for sorting (deferred).

>   **-d**   If file is a directory, list its name instead of its contents.

>   **-l**   Long list format.

>   **-p**   Put a slash after each name if the file is a directory (deferred).

>   **-r**   Reverse alpha or age sort order, as appropriate (deferred).

>   **-s**   Display size as well as name if -1 (numeral 1) option used.

>   **-u**   Use time of last access for sorting instead of last modification (deferred).

>   **-x**   Multicolumn output format, with entries sorted across page (deferred).

>   **-A**   Print annotation info.

>   **-C**   Multicolumn output format, with entries sorted down the columns (deferred).

>   **-F**   Puts a forward slash (/) after directory filenames, or asterisk (*) if executable (deferred).

>   **-H**   Print headings on long listings.

>   **-O**   Print unordered "-l" or "-1" format listings.

-P   Print one line with position/volume info.

-R   Recursively list directories.

-U   Print HPSS-specific information.

-V   Print volume for first tape level.

-X   Print extended volume info (for all levels).

-1   (Numeral 1) forces one name-per-line list.

Keywords
> DIR

Usage Notes
> None

Examples
> None

Related Commands

> **Section 19.4, "CD command"**

> **Section 19.53, "LS command"**

> **Section 19.46, "LCD command"**

> **Section 19.47, "LCDLS command"**

> **Section 19.48, "LLS command"**

19.6. CGET command

Synopsis
> **cget**[] [**-A**] [**-B** | **-b**] [**-C**] [**-F** *on|off*] [**-h**] [**-k**] [**-L** *path*] [**N**] [**-O** *tuple*] [**-P** | **-p**] [**-Q**] [**-R**] [**-S**] [**-T** *on|off*] [**-t**] [**-U** | **-u**] *path* …

Description
> This command conditionally retrieves a copy of a file from HPSS to your local file space on the host system only if a local copy does not already exist or the HPSS file is newer than an existing local file.

HSI Version
> All

Aliases
> None

Options

> **-A**   Enable auto-scheduling for retrievals in order to optimize tape mounts.

-B | -b   Backup option. Renames any existing local file by appending "~".

-C   Purge source file from disk cache after file(s) are copied successfully.

-F   Enable (**on**) or disable (**off**) the use of HPSS Local File Mover I/O.

-k   Keep partially transferred files if errors are encountered.

-h   Symlink option. If specified, create local symlink if HPSS symlink. If not set, reads through HPSS symlinks when copying files to local filesystem.

-L   Specifies pathname to a file containing a list of file segments to be retrieved. See the usage notes
path   for a description of the segment syntax.

-N   Disable auto-scheduling of retrievals.

-O   Partial file transfer specification. See the usage notes below for the format of the "tuple"
tuple   parameter. Multiple **-O** options can be specified for a command.

-p   Preserve timestamp. Attempts to copy the HPSS file's timestamp to the local file.

-Q   If running as local root user, attempts to preserve the HPSS owner and group on the local file.

-R   Recursively copy directories.

-S   Disable staging of the HPSS files; read directly from tape.

-T   Enable (**on**) or disable (**off**) the use of the HPSS Transfer Agent for I/O.

-t   Retransmit request. Appends to the local file, using the local files size as the starting offset within the HPSS file.

-U | -u   Update option. Only copy HPSS file to the local file if the HPSS file timestamp is newer.

Keywords
    BACKUP DIRn
Usage Notes

1. This command will overwrite an existing local file if the HPSS file has a more recent modification timestamp. Note that **cget** with **-t** does not take timestamps into account; rather, it acts as an unconditional "get" without checking to see if the HPSS file is newer than the local file.
2. The **-R** keyword can be used to recursively retrieve directory trees from HPSS, creating local subdirectories as needed.
3. Auto-scheduling is a mechanism used on file retrievals to optimize tape mounts, by organizing file retrievals so that all of the files that are on HPSS disk cache can be retrieved in the foreground, while files that are on tape are staged onto disk in the background. The staging is organized so that all of the files to be retrieved on a single tape virtual volume are scheduled together, sorted by their position on the volume.
4. See *Chapter 5, FTP compatibility and differences* for information on valid characters and renaming files on **get** and **put** commands.
5. See *Chapter 9, Using PIPEs for input and output* for information on piping files into or out of HPSS.
6. See *Chapter 3, Restricted TCP ports and restricted Mover hosts* for information on dealing with firewalls.

7. Partial Transfers:

    HSI can perform partial transfers either by specifying the **-t** parameter, which is normally used in error recovery situations, or by explicitly reading sections of a file, as specified by segment tuples. Segment tuples are specified by one or more **-O** options, or by reading a file of tuples from the file specified by the **-L** parameter.

    The syntax for segment tuples is:

    ```
    sourceOffset:sinkOffset:length
    ```

    Each of these numbers may optionally contain a case-insensitive multiplier suffix of the form "k", "kb", "m", "mb", "g", "gb", "t", "tb", "p", or "pb", for kilobytes, megabytes, gigabytes, terabytes, and petabytes.

    The current position of the source or sink file can be represented by the special string "CP" or "0CP".

Examples
    cget –R someFiles Project
Related Commands

Section 19.29, "FIND command"

Section 19.32, "GET command"

Section 19.17, "CPUT command"

Section 19.75, "PUT command"

19.7. CHACCT command

Synopsis
    chacc[t] [**-R**] *newacct path* …

Description

This command changes the Account ID of a file or directory.

HSI Version

All

Aliases

None

Options

-R	[standard option] Recursively traverse directories in the specified path(s).

newacct New account ID to be used.

path One or more pathnames to which the command applies.

Keywords

DIRn

Usage Notes

1. The account ID is an integer value. The required parameter *newacct* refers to account IDs as known to the HPSS system, not to local account IDs (except by coincidence). The list of HPSS account IDs is maintained by the HPSS system administrator.

Examples

    chacct 200 some_file

Related Commands

Section 19.11, "CHMOD command"

Section 19.12, "CHOWN command"

Section 19.10, "CHGRP command"

Section 19.66, "NEWACCT command"

19.8. CHACL command

Synopsis

chacl *action-option* [**-A**] [**-ic**] [**-io**] [**-Q**] [**-R**] *filelist*

Description

This command creates, updates, or deletes HPSS Access Control Lists.

The CHACL command is used to change the Access Control List for files and directories.

An Access Control List is composed of entries of the form:

    typ:usr:perms[,typ:usr:perms…]

where *typ* is the entry type. The most common entry types are:

- user
- group

- user_obj
- group_obj
- other_obj

Other less common types, which are described in the Access Control List section of this manual, include "foreign_user", "foreign_group", "foreign_other", "any_other", "mask_obj", "unauthenticated", "user_obj_delegate", "group_obj_delegate", "other_obj_delegate", "user_delegate", "group_delegate", "foreign_user_delegate", "foreign_group_delegate", "foreign_other_delete", and "any_other_delegate".

HSI Version
    4.0.1.3 and later

Aliases
    None

Options

-c	Clears all access permissions other than standard owner/group/other permissions.
-f *fname*	Copy the ACL control list from HPSS file *fname* to all files in filelist.
-r *aclentry*	Removes the specified *aclentry* list.
-u *aclentry*	Updates (or adds) the specified *aclentry* list.
-A	Display absolute pathnames in messages (default: list relative pathnames).
-ic	If specified, the ACL entry is for the directory's Initial Container ACL.
-io	If specified, the ACL entry is for the directory's Initial Object ACL.
-Q	Quiet mode - do not display normal messages for successful operations.
-R	Recursively operate on subdirectories.

Keywords
    DIR

Usage Notes

1. The **chacl** command is only available when communicating with an HPSS server.
2. Setting, updating, or deleting an ACL can be done by the object's owner without requiring any special permission (other than *read* if the hashcreate command is used). For all others, it requires either "group" or "other" *write* permission on the object.

Examples

1. Create an ACL to give read access to user bob and alice on file fileA: chacl –u user:bob:r,user:alice:r fileA
2. Remove user joe from the ACL for files whose names end with "NoJoe": chacl –r user:joe: *.NoJoe
3. Add an Initial Object ACL to directory OpenDir to allow read/execute access for user mike and group goodguys: chacl –ic –u user:mike:rx,group:goodguys:rx OpenDir

Related Commands
> **Section 19.54, "LSACL command"**

19.9. CHCOS command

Synopsis
> **chc[os]** [-A] [-h] [-i *subsys*] [-N] [-R] [-s *streamID*] [-T *max_threads*] [-Z] *new_cos|auto path* …
> [<<MARKER]

Description
> This command changes the Class of Service of one or more HPSS files.

HSI Version
> All

Aliases
> None

Options

-A	Enable stage optimizations (default).
-h	Follow symbolic links during recursion. The default behavior is to not follow symbolically-linked directories.
-i *subsys*	(HPSS 7.1+) Specify subsystem ID to use for multiple COS Change threads. Default is 0 - use the current subsystem.
-N	Disables auto-scheduling of files (by Volume and Position). This option normally should not be used, to avoid unnecessary tape mounts.
-R	Recursively traverse directories in the specified path(s).
-s *streamID*	Specifies COS Change stream ID (0 ..n) to use.
-T *max_threads*	(HPSS 7.1+) Set max COS Change threads to use. Default is to use all threads available for the subsystem. See Usage Notes below.
-Z	Internal autoscheduler debug flag. Displays scheduler results but doesn't issue the HPSS **chcos** call(s).

new_cos   Specifies the COS ID to be used, which must be one which you are eligible to use for the HPSS connection(s) specified by the "path" parameter(s) (use **lscos** to see a list of COS IDs). Use of `chcos 0` is always permitted; it removes files from the BFS chcos queue.

auto   Use auto-COS selection to choose the new COS ID, based upon current settings for # copies, account ID, group ID, and user ID.

MARKER   Specify HERE-document terminator sentinel (see options, below)

- MARKER may optionally be preceded by one or more whitespace characters following the here document sentinel "<<".
- Lines following the MARKER, which must be the last item on the line, will be interactively prompted for with "FILENAME". Each line will contain pathnames (which may include wildcards).
- The list is terminated by a line containing MARKER as the first token on the line.

Keywords
     None
Usage Notes

1. If *new-cos* is specified as auto, then each file object is tested to see in which COS it best fits based on criteria defined in the HSI COS file or in the HPSS system configuration. Otherwise HSI will attempt to change COS to that specified in the *new-cos* parameter.
2. The **-T** option is normally not useful except to throttle the number of threads that are used for **chcos** operations. Use of this option can cause **chcos** requests to be queued unnecessarily.

3. The shell-style HERE-document syntax can be used to specify an interactive list of paths, or can be used in an IN file: **Interactive mode:** `hsi chcos auto file1 file2 << EOF FILENAMES: file3 file4 FILENAMES: file5 FILENAMES: EOF`

   Using an IN File: `hsi in stageInput`

   Contents of file "stageInput": `chcos auto file1 file2 << EOF file1 : hpss_file1 file2 : hpss_file2 file3 file4 file5 : hpss_file5 EOF`

4. For HPSS Version 7.1 and later, HSI attempts to use multiple background "change cos" threads in the Core Server by first organizing all files by virtual volume and position, and then issuing requests for each change-cos thread so that:

   a. All requests for a particular tape VV are issued to the same change-cos thread
   b. As many change-cos threads as are configured are used concurrently.
5. HSI will query the current or specified (**-i** option) subsystem to determine the maximum number of change-cos threads that are available at the start of each new command, so that HPSS administrator changes to the number of available COS-change threads are automatically taken into account.

Examples

1. Recursively change all files to COS ID 2: `chcos -R 2 some_directory`
2. Change all files in the current directory from 1-copy to 2-copy, based on size. Note that files that are already in the correct COS will not be changed. `set copies=2; chcos auto *`

Related Commands

Section 19.56, "LSCOS command"

19.10. CHGRP command

Synopsis
> **chgrp** [**-h**] [**-R**] *group_owner path* …

Description
> This command changes group ownership of HPSS nodes.

HSI Version
> All

Aliases
> None

Options

> **-h** For symlinks, change ownership of symlink instead of object which it references. *Note: HPSS does not provide this capability, so symlinks are silently ignored if this option is specified.*

> **-R** [standard option] Recursively traverse directories in the specified path(s).

Keywords
> DIRn

Usage Notes

> 1. The required "groupname" parameter refers to a group name as known to the HPSS system, not to groups on the local host, except by coincidence. The HPSS group list is maintained by the HPSS system administrator.
> 2. The **-R** option can be used to recursively change the group ownership of all nodes within a directory tree.

Examples
> ```
> chgrp —R staff /usr/local/*
> ```

Related Commands

Section 19.11, "CHMOD command"

Section 19.12, "CHOWN command"

19.11. CHMOD command

Synopsis
> **chm**[**od**] [**-d**] [**-f**] [**-h**] [**-H**] [**-R**] *mode path* …

Description
> This command changes permissions associated with HPSS nodes.

HSI Version
> All

Aliases
> None

Options

-d   Operate only on directory objects.

-f   Operate only on file objects.

-h   Suppress mode change for a file or directory pointed to by a symbolic link.

-H   If the **-R** option is specified, symbolic links on the command line are followed. Default is to not follow symlinks for any path specified on the command line. Symbolic links encountered during recursion are never followed.

-R   [standard option] Recursively traverse directories in the specified path(s).

mode   Octal value or symbolic mode string. Run `man chmod` for more details.

path   Files and directories whose permissions are to be changed.

Keywords
          DIRn
Usage Notes

1. The mode of each named file is changed according to *mode*, which may be absolute or symbolic. An absolute mode is an octal number constructed from the OR of the following modes:

   0400 read by owner 0200 write by owner 0100 execute (search in directory) by owner 0070 read, write, execute (search) by group 0007 read, write, execute (search) by others

   A symbolic mode has the form: [who]*op*permission[,[who]*op*permission] …

   The who part is a combination of the letters *u* (for user's permissions), *g* (group) and *o* (other). The letter *a* stands for all; that is, *ugo*. If who is omitted, the default is *a* but the setting of the file creation mask (see **umask**) is taken into account.

   Note: Unlike previous versions of HSI, only one *mode* string can be specified, which may consist of multiple comma-separated sections, with no intervening whitespace. For example, the following is correct: `chmod -R go-rwX,u=rwX` *path* …

   whereas this will fail: `chmod -R go-rwX u=rwX` *path* …

   The string "u=rwX" will be interpreted as a pathname in the second example.

   Op can be "+" (plus) to add permission to the file's mode; "-" (minus) to take away permission; and "=" (equal) to assign permission absolutely (all other bits will be reset).

   Permission is any combination of the letters r (read), w (write), x (execute), X (set execute only if file is a directory or some other execute bit is set), s (set owner or group id) and t (save text - sticky).

Letters u, g, or o indicate that permission is to be taken from the current mode. Omitting permission is only useful with = to take away all permissions.

The octal-constant value is AND-ed with a mask of 0777; values greater than 0777 are silently truncated.

When the **-R** option is given, **chmod** recursively descends its directory arguments setting the mode for each file as described above. When symbolic links are encountered, their mode is not changed and they are not traversed.

Examples

    chmod 740 Project/file∗

    chmod u=rwx g+w o=g ∗.c

Related Commands

> **Section 19.10, "CHGRP command"**

> **Section 19.12, "CHOWN command"**

19.12. CHOWN command

Synopsis
> **chown** [**-h**] [**-R**] *Owner*[:|.*Group*] *path* …

Description
> This command changes the owner for HPSS nodes.

HSI Version
> All

Aliases
> None

Options

> **-h**   Changes the ownership of an encountered symbolic link and not that of the file or directory pointed to by the symbolic link. This applies to all files and directories encountered during recursion if the **-R** option is specified.

> ### Note

> This option, if specified, currently causes symbolic links to be ignored with a warning message, as HPSS does not provide the capability to change the ownership of a symlink.

> **-R**   Descends directories recursively, changing the ownership for each file. When a symbolic link is encountered and the link points to a directory, the ownership of that directory is changed but the directory is not further transversed.

> *Owner*   A username or UID found in the DCE registry.

Group   Optional group name or GID found in the DCE registry. If this form is used, there must not be whitespace between the *Owner* and *Group* names. Either ":" or "." may be used; for example, "owner:group" or "owner.group".

Keywords
    DIRn
Usage Notes

1. The required parameter "ownername" refers to login names as known to the HPSS system, not to local usernames (except by coincidence). The HPSS list of usernames is maintained by the HPSS system administrator.
2. The **-R** option can be used to recursively change the ownership of all nodes within a directory tree.

Examples
    chown –R nastran /usr/local/Nastran
Related Commands

Section 19.10, "CHGRP command"

Section 19.11, "CHMOD command"

19.13. CLOSE command

Synopsis
    close [*connectionID|drive: …*]
Description
    This command closes an HPSS server connection.
HSI Version
    All
Aliases
    None
Options

connection ID Numeric connection ID.

drive: Logical drive prefix, including colon (":") character.

Use **lsc**[**onnections**] for list of current open connections. If no connections are specified, the current connection is closed. It is an error to attempt to close the last active connection.

Keywords
    None
Usage Notes

1. An HPSS connection can be closed by specifying the virtual drive letter of the remote HPSS system. The last open connection cannot be closed via this command; instead, the **end** command or one of its aliases must be used to terminate the program.

Examples
    close A:

Related Commands

> **Section 19.68, "OPEN command"**

> **Section 19.90, "SWITCH command"**

> **Section 19.55, "LSCONNECTIONS command"**

19.14. COPIES keyword

Format
> copies = *n*

Description
> Set a number of tape copies for files written to HPSS.

HSI Version
> All

Aliases
> None

Options
> None

Keywords
> None

Usage Notes

> 1. Number of copies is used during automatic COS selection (see *Section 19.15, "COSID keyword"*). This is normally "1" or "2", although some sites may allow up to "5". The default number of copies is defined at compile time, and may be overridden by the setting of the copies parameter in the global section of the `hsirc` file.
> 2. The number of copies defined for a Class of Service can be displayed by using the **Section 19.56, "LSCOS command"** command.

Examples

> 1. Set the default number of copies for the session: `copies = 1; put myfile`
> 2. Override the session setting for one command: `put copies=2 myfile`

Related Commands

> **Section 19.56, "LSCOS command"**

> *Section 19.15, "COSID keyword"* keyword

19.15. COSID keyword

Format
> cos[id] = {*numeric cosid* | **auto**}

Description
> Set HPSS Class of Service to be used for **put** and **cp** commands.

HSI Version
> All

Aliases
> None

Options
> None

Keywords
　　None
Usage Notes

1. *numeric cosid* can be any value that is defined for the site (use the **Section 19.56, "LSCOS command"** command to obtain a list of valid IDs for the current connection). If **auto** is specified (this is the default setting), HSI will automatically select the Class of Service based upon the number of copies (see the *Section 19.14, "COPIES keyword"* keyword), the file size, and the UID, GID, and Account ID. The HPSS administrator may optionally restrict access to certain Classes of Service to particular users or groups. Any access that the user is prohibited from using are marked with "N" in the **lscos** output.

Examples

1. Set automatic COS selection for the session: `cos = auto; put myfile`
2. Override the session COS setting for the current command: `put cos=20 myfile`

Related Commands
　　Section 19.56, "LSCOS command"

19.16. CP and CCP commands

Synopsis

cp *options sourceFile targetFile* **cp** *options path … TargetDirectory*

ccp *options sourceFile targetFile* **ccp** *options path … TargetDirectory*

Description
　　This command copies an HPSS-resident file to another HPSS-resident file, or copies HPSS file(s) and directories to another HPSS directory.
HSI Version
　　All (the **ccp** command was added in HSI Version 4.0.1.3)
Aliases
　　(for **CP**): **copy**
Conditional Copy command (CCP) description

- The **ccp** command will only copy a source file to the target file if:

  1. The destination file does not already exist
  2. OR the destination file is older than the source file.

- The **ccp** command is identical in function to the **cp** command, except as follows:

  1. **ccp** automatically enables the **-p** (preserve timestamp of source file) and **-U** (update) options.
  2. The **-f** (force) option is not allowed.

Options

- **-**　Cache purge option. If specified, purge source files from HPSS disk cache after a successful copy.
- **C**　Normally used only when it is expected that files will only be read once (or infrequently), to help optimize HPSS disk cache use.

-c   The object ACL of the source object is copied to the destination object.

-f   Force removal of the target file instead of renaming (same as "backup=off").

-h   Copy symbolic links (default is to read through symlinks).

-i   Prompt before copying files (interactive mode).

-ic   The Initial Container ACL for the source directory is copied to the target directory.

-io   The Initial Object ACL for the source directory is copied to the target directory.

-m   Interhpss copy method (**local** or **server**). Default is **server**.

-n   Only copy files modified within the specified number of days.

-p   Preserve time stamp of source file.

-R   Recursively copy source directories.

-S   Disable staging of the source file (attempts to read directly from tape). Users having trouble retrieving files over 25 GB in size may wish to use the **-S** option for the **get** or **mget** commands. The **-S** option was added to allow users to disable staging files from tape, specifically for this type of situation.

-x   Use extended I/O calls. The "which" parameter is *source*, *sink*, or *both*. This option is normally only used when "firewall" mode is in effect, and designates whether firewall-type I/O should be used when reading (*source*) files, writing files (*sink*) or both (*both*). One scenario where this is useful is copying files between HPSS systems, when one or the other HPSS systems lives behind a firewall.

Keywords
    DIRn, Class of Service (COS)
Usage Notes
    In the second form, each file is copied to the subdirectory using the same filename as the original.

Examples

1. Create a copy of an existing file "file1": cp file1 file2
2. Create a copy of an existing file (file.old) in Class of Service 22: cp cos=22 file.old file.new

Related Commands

Section 19.8, "CHACL command"

Section 19.65, "MV command"

Section 19.30, "FIREWALL command"

19.17. CPUT command

Synopsis
{**cput** | **s[ave]**} [**-A** *string*] [**-B** | **-b**] [**-d**] [**-F** *on|off*] [**-h**] [**-M** *mode*] [**-n**] [**-P** | **-p**] [**-Q**] [**-R** | **-r**] [**-T** *on|off*] [**-t**] [**-U** | **-u**] [**-X** *max*] [*path …*]

Description
This command conditionally stores a file if the HPSS file does not exist or the local file is newer.

HSI Version
All

Aliases
save

Options

-A   Annotation string for HPSS file (type annotate ? for more details).

-B | -b Backup option. Renames existing HPSS file by appending "~".

-d   Remove local files after successful transfer to HPSS.

-F   Enable (**on**) or disable (**off**) use of the HPSS Local File Mover, overriding any configuration settings in the HSIRC file(s).

-h   Symlink option. If specified, create symlink in HPSS if local symlink. If not set, reads through local symlinks when copying files to HPSS.

-M   Specifies octal mode to use for file creations.
mode

-n   Only put files modified within last *days* number of days.
days

-P   Create intermediate HPSS subdirectories for the file(s) if they do not exist.

-p   Preserve timestamp. Attempts to copy local file's timestamp to HPSS file.

-Q   If running as the HPSS "root" user, preserve the local owner/group for HPSS files and directories that are created.

-R | -r   Recursively store directories.

-T   Enable (**on**) or disable (**off**) use of the HPSS Transfer Agent, overriding any configuration settings in the HSIRC file(s).

-t   A "re-put" operation. Restart a previously failed operation, using the size of the existing HPSS file as the starting offset. This overrides the **-U** flag, and it also overrides the **cput** command.

-U | -u   Update option. Only copy file to HPSS if local file modification timestamp is newer.

-X   Set maximum concurrent transfers.
max

Keywords
    BACKUP DIRn
Usage Notes

1. This command will overwrite an existing HPSS file if the local file has a more recent modification timestamp. Note that **cput** with **-t** does not take timestamps into account; rather, it acts as an unconditional "retransmit" without checking to see if the local file is newer than the HPSS file.
2. The **-R** option may be specified to recursively CPUT a directory tree. In this case, HPSS subdirectories are automatically created as needed while traversing the local directory tree.
3. See *Chapter 5, FTP compatibility and differences* for information on valid characters and renaming files on **get** and **put** commands.
4. See *Chapter 9, Using PIPEs for input and output* for information on piping files into or out of HPSS.
5. See *Chapter 3, Restricted TCP ports and restricted Mover hosts* for information on dealing with firewalls.

Examples
    None
Related Commands

Section 19.75, "PUT command"

Section 19.6, "CGET command"

Section 19.32, "GET command"

19.18. CRENAME command

Synopsis
    cren[**ame**] [**-d**] [**-ocos** *cosid*[,*cosid* …]] [**-ncos** *cosid*[,*cosid* …] [**-s**]] *new_path orig_path*
Description

This command conditionally renames an existing file by swapping attributes with a replacement file. It exists to assist with the administrative problem of copying files from one media type to another, or from one class of service (COS) to another, while preserving the owner/group ID, account ID, and modification times of the existing file. The scenario in which this command was designed to be used is as follows:

1. The administrator uses the HPSS root account to make a copy of a user file, perhaps to a hidden directory, or perhaps to a file in the same directory with a different suffix.

2. The **crename** command is used to verify that both the original and copied files meet the mandatory and optional criteria described below. If so, then:

   a. The owner, group, account and permissions and Nameserver attributes (utime fields) of the original file are copied to the new file.
   b. The original file is temporarily renamed by appending a tilde ("~") character.
   c. The new file is renamed to the original filename.
   d. The original file is optionally deleted if all went well.

This command, in conjunction with the **cp** command, provides approximately the same functionality as **chcos**; however, since multiple HSI sessions can be run in parallel, it is often faster to use **cp** and **crename** than to use the HPSS single-threaded **chcos** capability.

HSI Version
    All
Aliases
    None
Options

   -d   Delete original file if the rename operation succeeds.

   -s   Suffix string to append to orig_path when renaming it. Default is "~".

   -   Filter option. If specified, the Class of Service of file *orig_path* must be one of the comma-
   ocos   separated COS IDs specified, or the command will fail.

   -   Filter option. If specified, the Class of Service of file *new_path* must be one of the comma-
   ncos   separated COS IDs specified, or the command will fail.

Keywords
    None
Usage Notes

    1. Both files must reside in the same HPSS.

2. Both files must exist, or the command will fail.
3. Both files must be exactly the same size.
4. The **crename** command *currently does not preserve HPSS ACLs*.

Examples

1. Create a copy of an existing file in Class of Service 22: `cp cos=22 file.old file.new`
2. Conditionally rename the new file so that it replaces the existing file, while preserving timestamps, as well as owner, group, and account IDs. The `file.new` will only replace `file.old` if it was written to COS 22, and is exactly the same size as `file.old`: `crename –ncos 22 file.new file.old`

Related Commands

Section 19.77, "RENAME command"

Section 19.12, "CHOWN command"

Section 19.11, "CHMOD command"

Section 19.7, "CHACCT command"

19.19. DEBUG command

Synopsis
  debug *level*
Description
  This command sets the debug level.
HSI Version
  All
Aliases
  None
Options

  level : A number in the range 0 to 5. 0 disables debug messages, levels 1-5 enable increasingly higher levels of debug, roughly corresponding to the following:

  1   Event and high-level information messages

  2   Normal debug messages that help track the flow of command processing

  3   Higher level of debug used for inner loops, table traversals, etc.

  4   I/O debug - mover message traces

  5   Trace-level debug

> *on* Equivalent to "debug 1"

> *off* Equivalent to "debug 0"

Keywords
> None

Usage Notes
> None

Examples
> `debug on`

Related Commands
> None

19.20. DIRn keyword

Format
> DIR*n* = *path*

Description
> Set working directory 0-9.

HSI Version
> All

Aliases
> None

Options
> None

Keywords
> None

Usage Notes

> 1. The *path* parameter may contain wildcards as long as it resolves to a single HPSS directory node name.
> 2. *This command is deprecated, and will be removed in a future version.*

Examples
> `dir0=/home/kericson`

Related Commands
> None

19.21. DU command

Synopsis
> **du** [**-?**] [**-a**] [**-b** *date*] [**-e** *date*] [**-k**] [**-s**] [**-w** *n*] [*path* …]

Description
> This command gives the number of bytes contained in all files and, recursively, directories within each specified directory or file name. If name is missing, "." is used.

HSI Version
> All

Aliases
> None

Options

-a   If specified, causes an entry to be printed for each file.

-b   If specified, only counts files written on or since the specified date. The date is of the form "yyyy/mm/dd".

-e   If specified, only counts files written before or on the specified date. The date is of the form "yyyy/mm/dd".

-k   If specified, prints values in kilobytes (1k = 1024). Default is 512 byte blocks.

-s   If specified, causes only the grand total to be printed.

-   If specified, only counts files written within the last *n* days.
w

Keywords
    None
Usage Notes

1. If neither **-s** nor **-a** is specified, an entry is printed for each directory. An error occurs if both **-w** and (**-b** or **-e**) is specified in such a way that it is impossible for any files to qualify.

Examples
    du −k
Related Commands
    None

19.22. DUMP command

Synopsis
    dump [-A] [-x] [-j] *path* …
Description
    This command dumps name server and bitfile server metadata about file(s).
HSI Version
    All
Aliases
    None
Options

-A   Display absolute pathnames (default = relative).

-x   Dump extended attributes for object.

-j   Display output in JSON format.

Keywords
    None
Usage Notes

   1. This command returns a lot of detailed information about the file(s) queried. The **-x** and **-j** option
      uses metadata resources intensively.

Examples
    dump my_file
Related Commands
    Section 19.53, "LS command"

19.23. ECHO command

Synopsis
    echo
Description
    This command is an on/off toggle for echoing command input.
HSI Version
    All
Aliases
    None
Options
    None
Keywords
    None
Usage Notes

   1. The internal "echo" flag is tested when determining whether to display lines that are read from "IN"
      file.

Examples
    echo
Related Commands
    None

19.24. END command

Synopsis
    {**end** | **bye** | **q[uit]** | **ex[it]**}
Description
    This command terminates execution of HSI.
HSI Version
    All
Aliases
    bye, quit, exit
Options
    None
Keywords
    None

Usage Notes

1. A non-zero exit code is returned reflecting the highest severity of error encountered during the session. If no errors occurred, then an exit code of zero is returned.

Examples
    q
Related Commands
    None

19.25. EXPDEL command

Synopsis
    expdel[**ete**] [**-A**] [**-R**] *path* …
Description
    This command deletes any File Expiration time that is set for the specified file(s).
HSI Version
    HSI Version 6.0.0 patch 4 and later
Aliases
    None
Options

-A	Displays absolute pathnames for files whose expiration time is deleted. Default is to display relative pathnames.
-R	Recursively deletes expiration times for files contained in subdirectories.

Keywords
    None
Usage Notes

1. HPSS administrators may disable the file expiration feature; if so, this command will have no effect.
2. HPSS administrators may also restrict the use of this command.

Examples

1. Recursively remove expiration time for all files that match the pattern "oldfile?", such as oldfilea, oldfileb, and oldfilec, and for files contained in subdir1 and all of its subdirectories: expdel –R oldfile? subdir1

Related Commands

Section 19.26, "EXPFIND command"

Section 19.27, "EXPLS command"

Section 19.28, "EXPSET command"

19.26. EXPFIND command

Synopsis

expfind [**-A**] [**-b** *beginTime*] [**-d** *days*] [**-e** *endTime*] [**-R**] *path* …

Description

This command finds and lists files whose expiration time meets the specified selection criteria.

HSI Version

HSI Version 6.0.0 patch 4 and later

Aliases

None

Options

-A   Displays absolute pathnames for files whose expiration time is deleted. Default is to display relative pathnames.

-b   Specifies beginning time in range.

-d   Find files that will be expiring within the specified number of days from today, starting at 00:00:00 today.

-e   Specifies ending time in range.

-R   Recursively searches subdirectories specified by path.

Keywords

None

Usage Notes

1. HPSS administrators may disable the file expiration feature; if so, this command will have no effect.
2. If no path(s) are specified, files in the current directory are searched.
3. The time value for the **-b** and **-e** option is of the form: YYYY-MM-DD[-hh:mm:ss] YYYY : year MM : month (1 -12) DD : day [1 - n, where n depends on the month (and year, for leap years) hh:mm:ss - optional hour:minute:second substring. If omitted, it is treated as 00:00:00.
4. If **-b** is specified but **-e** is omitted, then files whose expiration time is greater than or equal to *beginTime* are listed.
5. If **-e** is specified but **-b** is omitted, then files whose expiration time is less than or equal to *endTime* are listed.

Examples

1. Find all files in the current directory that will be expiring within the next 10 days: `expfind -d 10`
2. Recursively find all the files in subdirectory "subdir1" that will be expiring after 12 January 2020: `expfind -R -b 2020-01-12 subdir1`
3. Find all files in the current directory that will be expiring between 16 May 2018 and 25 August 2019: `expfind -b 2020-01-12 subdir1`
4. Find all files in the current directory and all of its subdirectories that will be expiring between noon on 1 June 2017 and 4 July 2017 at 3 PM: `expde -R -b 2017-06-01-12:00:00 -e 2017-07-04-15:00:00`

Related Commands

19.27. EXPLS command

Synopsis
> **expls** [**-A**] [**-R**] [**-v**] *path* …

Description
> This command lists the expiration time for the specified file(s).

HSI Version
> HSI Version 6.0.0 patch 4 and later.

Aliases
> **lsexp**

Options

> **-A**   Displays absolute pathnames for files. Default is to display relative pathnames.

> **-R**   Recursively list expiration times for files contained in subdirectories.

> **-v**   Verbose mode. If specified, additional information (owner, group, time that expiration was set) are displayed.

Keywords
> None

Usage Notes

> 1. If no paths are specified, files in the current directory are listed.
> 2. HPSS administrators may disable the file expiration feature; if so, this command will have no effect.

Examples

> 1. Recursively list expiration time for all files that end in `.o`: `lsexp *.o`
> 2. Recursively list expiration times for files contained in `/pub/project1` and all of its subdirectories. Use verbose mode in order to see the owner and group. `expls −Rv /pub/project1`

Related Commands

19.28. EXPSET command

Synopsis

expset [**-A**] [**-b** {**c** | **m** | **n** | **t**}] [**-c**] [**-d** *days*] [**-h** *hours*] [**-m** *minutes*] [**-s** *seconds*] [**-y** *years*] [**-R**] [**-t** *expireTime*] [**-v**] *path* …

Description

This command sets the expiration time for the specified file(s).

HSI Version

HSI Version 6.0.0 patch 4 and later.

Aliases

None

Options

-A Displays absolute pathnames for files whose expiration time is deleted. Default is to display relative pathnames.

-b Set baseTime type as: **c** : File creation time **m** : Modification time **n** : Now (time at which the **expset** command is issued) **t** : Today at 00:00:00.

-c Clear expiration time.

-d Set number of days in expiration time calculation.

-h Set number of hours in expiration time calculation.

-m Set number of minutes in expiration time calculation.

-s Set number of seconds in expiration time calculation.

-R Recursively deletes expiration times for files contained in subdirectories.

-t Set expiration time in the form YYYY-MM-DD or optionally, YYYY-MM-DD-hh:mm:ss, where: YYYY is the year MM is the month number (1-12) DD is the day number (1-31, depending on the month and leap year) hh is the hour (0-23) mm is the minute (0-59) ss is the second (0-59)

-v Enable verbose listing mode.

-y Set number of years in expiration time calculation.

Keywords

None

Usage Notes

1. If **-b** is not specified, "**-b** *n*" is assumed (base time is current time when command is issued).
2. Options **-h**, **-m**, **-s**, **-d**, and **-y** may be combined to specify base time + years/days/hours/minutes/seconds.
3. Options **-c** and **-t** may not be combined with any other options.
4. HPSS administrators may disable the file expiration feature, and may also restrict the use of this command.

Examples

1. Set expiration time for all files that match the pattern ∗.c to their creation time + 2 years, 8 days, and 2 minutes: `expset −b c −y 2 −d 8 −m 2 ∗.c`
2. Recursively set expiration time for all files in the directory `OldProject` to now + 30 days: `expset − R −d 30 OldProject`
3. Recursively set the expiration date for all files in the directory ~Joe to 5 PM on 1 January 2020: `expset −R −t 2020−1−1−17:00:00 ~Joe`

Related Commands

Section 19.25, "EXPDEL command"

Section 19.26, "EXPFIND command"

Section 19.27, "EXPLS command"

19.29. FIND command

Synopsis
   find *pathname-list expression*
Description
   This command recursively descends the directory hierarchy for each pathname in the pathname-list, seeking files that match a boolean (logical) expression written in the primaries given below. In the descriptions, the argument n is used as a decimal integer where +n means more than n, -n means less than n, and n means exactly n.
HSI Version
   All
Aliases
   None
Options

   -cos *cosid*
      True only for plain files whose Class of Service ID matches that specified by *cosid*.
   -name *filename*
      True if the *filename* argument matches the current file name. Wildcard characters are allowed, with or without surrounding quotes (either single quotes or double quotes).
   -perm *onum*
      True if the file permission flags exactly match the octal number *onum*. Only the lowest 3 octal digits are significant.
   -prune
      Always true. Has the side effect of pruning the search tree at the file. That is, if the current path name is a directory, **find** will not descend into that directory.
   -type *c*

True if the type of the file is *c*, where *c* is one of:

- d for directory
- f for plain file
- l for symbolic link

-user *uname*
: True if the file belongs to the user *uname*. If *uname* is numeric and does not appear as a login name in the HPSS password database, it is taken as a user ID.

-nouser
: True if the file belongs to a user not in the password database.

-group *gname*
: True if the file belongs to group *gname*. If *gname* is numeric and does not appear as a login name in the HPSS group database, it is taken as a group ID.

-nogroup
: True if the file belongs to a group not in the group database.

-size [+|-] *n*

- If "+" (plus) is specified, true if the filesize is greater than *n* bytes;
- If "-" (minus) is specified, true if the filesize is less than *n* bytes;
- Otherwise, true if the file is exactly *n* bytes.

-atime *n*
: True if the file has been accessed in *n* days. Note that the access time of directories in *path-name-list* is changed by the **find** command.

-mtime *n*
: True if the file has been modified in *n* days.

-ctime *n*
: True if the file has been changed in *n* days. "Changed" means that the file's "time of last metadata update" field has changed.

-print
: Always true; the current pathname is printed.

-ls
: Always true; causes the current pathname to be printed together with its associated statistics. These include (respectively) inode number, size in kilobytes (1024 bytes), protection mode, user, group, size in bytes, and modification time. If the file is a symbolic link the pathname of the linked-to file is printed preceded by "→".

-newer file
: True if the current file has been modified more recently than the argument file.

-depth
: Always true; causes descent of the directory hierarchy to be done so that all entries in a directory are acted on before the directory itself.

Keywords
: None

Usage Notes
: None.

Examples
: ```
find . /usr/local/bin -name *.doc -print
```

**Related Commands**
: None

# 19.30. FIREWALL command

**Synopsis**

**fire**[**wall**] [**-on** | **-off**] [**-c**] [**-s**] [*drive* …]

**Description**

This command causes HSI to use store-and-forward I/O when transferring files through a firewall.

**HSI Version**

All

**Aliases**

None

**Options**

**-on**  Enables firewall mode for the specified connection(s).

**-off**  Disables firewall mode for the specified connection(s).

**-c**  Client side will create the private socket used for data transfers and server will issue the "connect" call.

**-s**  Server side will create the private socket used for data transfers, and client will issue the "connect" call (default).

*drive*  Logical drive(s) to which the command applies.

**Keywords**

None

**Usage Notes**

1. If no logical drives are specified, the options apply to the current connection.
2. If neither **-on** nor **-off** option is used, current settings will be displayed for the specified drive(s).
3. When firewall mode is in effect, auto-scheduling is disabled for the **get** family of commands.

**Examples**

1. Specify that all I/O to logical drive D: should use store-and-forward I/O via private data socket. By default, the server will create the private data socket, and the client will issue a "connect": `firewall –on d:`
2. Disable store-and-forward I/O for logical drives D: and E:, and list the current setting for drives F: and G: `firewall –off D: E:; firewall f: G:`

**Related Commands**

**Section 19.59, "LSSITES command"**

**Section 19.55, "LSCONNECTIONS command"**

**Section 19.16, "CP and CCP commands"**

**Section 19.75, "PUT command"**

# 19.31. FREE command

**Synopsis**

 **fr**[**ee**] [**-l**] [**-h**] *keyset*

**Description**

 This command deletes a saved keyset.

**HSI Version**

 All

**Aliases**

 None

**Options**

 **-l** Keyset lives in a local file (not currently implemented).

 **-h** Keyset is an HPSS-resident file (default).

**Keywords**

 None

**Usage Notes**

 1. This command is deprecated and will be removed in a future release.

**Examples**

 None

**Related Commands**

 **Section 19.1, "ADOPT command"**

 **Section 19.45, "KEEP command"**

 **Section 19.85, "SHOW command"**

# 19.32. GET command

**Synopsis**

 **g**[**et**] [**-A**] [**-B** | **-b**] [**-C**] [**-c** *on|off*] [**-F** *on|off*] [**-h**] [**-k**] [**-L** *path*] [**-N**] [**-o**] [**-O** *srcofs:sinkofs:length*] [**-P** | **-p**] [**-Q**] [**-R** | **-r**] [**-S**] [**-T** *on|off*] [**-t**] [**-U** | **-u**] [**-X** *max*]

**Description**

 This command retrieves a copy of a file from HPSS to your local file space on the host system.

**HSI Version**

 All

**Aliases**

 **recv**

**Options**

 **-A** Enable auto-scheduling for file retrievals. This is the default unless auto-scheduling has been disabled. Auto-scheduling results in optimizing tape mounts and tape positioning for retrieval of files that live on tape.

 **-B** | **-b** Backup option. Renames any existing local file by appending "~". (This option may not be used with **-L**, **-t**, or **-O**.)

**-C**            Cache purge option. If specified, purge files from HPSS disk cache after a successful **get**. Normally used only when it is expected that files will only be fetched once (or infrequently), to help optimize disk cache use.

**-c**            Enable (**on**) or disable (**off**) verification of HPSS checksums that are stored separately from the file data in HPSS. This option overrides both the default compile-time option, and the hsirc setting for enabling or disabling checksums. Checksums are created either by storing files with checksums enabled (see the **put** command), or by using the **hashcreate** command to generate a checksum for an existing HPSS file.

**-F**            Enable (**on**) or disable (**off**) use of the HPSS Local File Mover for file transfers, overriding the configuration option(s).

**-h**            Symlink option. If specified, create local symlink if HPSS file is a symlink. If not set, reads through HPSS symlinks when copying files to local filesystem.

**-k**            Keep partially transferred files if errors are encountered. Default is to attempt to delete any partially transferred local file if errors occur.

**-L** *path*     pathname of a file containing lists of partial file offsets, in the form specified by **-O** (see below). Multiple entries per line may be specified. **-O** option(s) are processed before any **-L** entries.

**-N**            Disable auto-scheduling of retrievals. Default is to schedule retrievals in an optimal way so as to minimize HPSS tape mounts.

**-O**            Partial file transfer. *srcofs* is the source file offset, *sinkofs* is the sink file offset, *length* is the transfer length. Missing options use defaults (src: 0, sink: 0, length: entire file). "CP" or "0CP" (case-insensitive) can be used to specify "current position". Multiple **-O** options may be specified, and are processed in order. Offsets may specified with case-insensitive suffixes "k", "kb", etc.

**-P**            Reserved for future use.

**-p**            Preserve timestamp. Attempts to copy HPSS file's timestamp to local file.

**-Q**            If running as local root user, attempts to preserve the HPSS UID and GID on the local file(s).

**-R | -r**      Recursively fetch directories.

**-S**      Disable staging of the source file (attempts to read directly from tape). Users having trouble retrieving files over 25 GB in size may wish to use the **-S** option for the **get** or **mget** commands. The **-S** option was added to allow users to disable staging files from tape, specifically for this type of situation.

**-T**      Enable (**on**) or disable (**off**) the use of the HPSS Transfer Agent for I/O, overriding any configuration options.

**-t**      A "re-get" option. Restart a previously failed operation. This is equivalent to "-O sizeof(localfile):sizeof(localfile):0". Note: **-t** may not be used with **-O** or **-L**.

**-U | -u**      Update option. Only copy file from HPSS if local file timestamp is older.

**-X** *max*      (HSI Version 4.0.1.2 and later) Set maximum concurrent transfers for this command. This may only be used to reduce, not increase, the value set by the HPSS administrator.

**MARKER**  Provides sh-style HERE-document syntax for specifying filenames.

- MARKER may optionally be preceded by one or more whitespace characters following the here document sentinel "<<".
- Lines following the MARKER, which must be the last item on the line, contain pathnames (which may include wildcards).
- The list is terminated by a line containing MARKER as the first token on the line.

**Keywords**
    BACKUP DIRn
**Usage Notes**

1. If either **-O** or **-L** is specified, the offsets apply only to the first file transferred.
2. See *Chapter 5, FTP compatibility and differences* for information on valid characters and renaming files on **get** and **put** commands.
3. See *Chapter 9, Using PIPEs for input and output* for information on piping files from HPSS into an operating system utility or to the TTY.
4. See *Chapter 3, Restricted TCP ports and restricted Mover hosts* for information on dealing with firewalls.
5. The shell-style HERE-Document syntax can be used to specify an interactive list of paths, or can be used in an IN file: **Interactive mode:** `hsi get file1 file2 << EOF FILENAMES: file3 file4 FILENAMES: file5 FILENAMES: EOF`

   **Using an IN File:** `hsi in stageInput`

**Contents of file "stageInput":** `get file1 file2 << EOF file1 : hpss_file1 file2 : hpss_file2 file3 file4 file5 : hpss_file5 EOF`

**Examples**

1. Get an hpss file call "hpss_file_name" to a local file called "local_file_name": `get local_file_name : hpss_file_name`
2. Recursively retrieve all of the files in the current directory that end with `.c`, and all of the files in the directories `Project1` and `Project2`: `get -R *.c Project1 Project2`
3. Get the file "precious_data", verifying the checksum that was previously created by storing it with **put -con** *precious_data*: `get -con precious_data`
4. Get the file `myHPSSfile.tar.gz` to a local pipeline which uncompresses and untars the file: `get "| gunzip | tar xvf - " : myHPSSfile.tar.gz`

**Related Commands**

**Section 19.29, "FIND command"**

**Section 19.6, "CGET command"**

**Section 19.17, "CPUT command"**

**Section 19.75, "PUT command"**

# 19.33. GIVE command

**Synopsis**

**give** [**-f**] [**-q**] *destuser filelist* **give** [**-l** | **-n**] *destuser* [*filelist*] **give -u** *destuser* [*filelist*]

The first form copies [or re-copies, with **-f**] all the files in *filelist* to *destuser*'s spool area. The second form lists files that have been given to *destuser* that are still in the spool area. The third form unlinks files from the giver for *destuser* that still exist in the spool area.

**Description**

This command gives HPSS file(s) from one HPSS user to another.

Files are copied to a spool area designated by the HSI administrator, and owned by the destination user. There is a separate subdirectory in the spool area destination user, and a separate subdirectory for each "giver" within the destination user's directory.

Files that have been given to a different user can be listed (**-l** option) or removed by the original giver (**-u** option). The destination user must issue a **mv** or **rename** command to move the file(s) from the spool area to a different directory that they have write access to.

The path to the global spool area is defined in a configuration file at each site, if this feature is enabled; contact the site's HSI administrator to obtain the pathname to the global spool area.

**HSI Version**
4.0.1.3 and later.
**Aliases**
None
**Options**

*destuser*   Numeric user ID or user Name to which files will be given.

**-f**        Force overwrite - files with conflicting names will be overwritten.

**-h**        Help - prints help for this command.

**-l**        List the files that have been given to *destuser*. If *filelist* is specified, restrict the list to the files in *filelist*.

**-n**        Like the **-l** option, except use a narrow list format.

**-q**        Quiet mode - suppress normal output about files that are copied.

**-u**        Unlink option - causes files in *destuser*'s spool area, from this user, to be unlinked.

*filelist*    a space-delimited list of one or more files to be copied.

**Keywords**
    None
**Usage Notes**

1. Subdirectory paths are NOT created in the spool area.
2. Wildcard patterns for filenames in the spool directory (that is, for the **-u** option) are NOT currently expanded.
3. The **give** command currently does not copy files between HPSS subsystems.
4. For symbolic links in *filelist*, the link contents are used as the file path.
5. This command is only available when communicating with an HPSS server. It is not available in the non-HPSS gateway.

**Examples**

1. Give files F1 through F3 to user bob: `give F[1-3] bob`
2. List files given to user bob by this user: `give -l bob`
3. Un-give file F2 that was previously given to user bob: `give -u bob F2`
4. Un-give all files that were previously given to user bob: `give -u bob`

**Related Commands**
    None

# 19.34. GLOB command

**Synopsis**
    **glob**

**Description**
   This command toggles enabling wildcard pattern-matching for filenames. The normal mode is "enabled".
**HSI Version**
   All
**Aliases**
   None
**Options**
   None
**Keywords**
   None
**Usage Notes**

> 1. If globbing is disabled, all pathnames that are specified must exactly match, and wildcard characters
>    (*, ?, {, }, [, ], and \) are treated as normal characters in pathnames.

**Examples**
   `glob`
**Related Commands**
   None

# 19.35. GROUPS command

**Synopsis**
   **gro[ups]**
**Description**
   This command displays HPSS group membership.
**HSI Version**
   All
**Aliases**
   None
**Options**
   None
**Keywords**
   None
**Usage Notes**

> 1. Displays integer ID and name of all HPSS groups of which you are a member.

**Examples**
   `groups`
**Related Commands**
   None

# 19.36. HASHCREATE command

**Synopsis**
   **hashcr[eate]** [**-A**] [**-C**] [**-F on|off**] [**-H** *type*] [**-h**] [**-N**] [**-R**] [**-S**] [**-T on|off**] *path* … [**<<MARKER**] …
**Description**
   This command creates a checksum hash for existing HPSS files.
**HSI Version**
   4.0.1.1 and later.
**Aliases**
   None

## Options

**-A**          Enable auto-scheduling of retrievals for files that live on tape.

**-C**          Cache purge option. If specified, purge files from HPSS disk cache after a successful hash create. Purging files from disk cache can help to optimize the use of this global HPSS resource when the files are not going to be fetched again for a long time.

**-F**          Explicitly use (**on**) or disable (**off**) Local File Mover I/O, overriding the settings based upon file size and the global enable/disable flag from the hsirc file.

**-H** *type*   Set case-insensitive checksum hash type to use (none, sha1, sha224, sha256, sha384, sha512, md5, crc32, adler32). Overrides hsirc and compile time options.

**-h**          Symlink option. If specified, ignores HPSS symlink. If not set, reads through HPSS symlinks when reading files.

**-N**          Disable auto-scheduling of retrievals (inverse of option **-A**). The normal default is to enable auto-scheduling in order to optimize tape mounts and tape positioning.

**-R**          [standard option] Recursively create hash entries for files in the specified HPSS path(s).

**-S**          Disable staging of the source file (attempts to read directly from tape).

**-T**          [future] Explicitly use (**on**) or disable (**off**) Transfer Agent I/O.

**MARKER**      Provides sh-style HERE document syntax for specifying filenames.

- **MARKER** may optionally be preceded by one or more spaces following the here document sentinel **<<**.
- Lines following **MARKER**, which must be the last item on the line, contain pathname(s) (which may include wildcards).
- The list is terminated by a line containing **MARKER** as the first token on the line.

## Keywords
None
## Usage Notes

1. The **-A** option is usually the default, but may be disabled in the hsirc file.
2. This command requires read access to the file in order to generate the hash checksum.
3. The HPSS Transfer Agent does not currently support checksum hashes.

**Examples**

1. Create a checksum hash for all `.c` files in the current directory: `hashcr *.c`
2. Create a checksum for the files F1 through F5 using HERE-document syntax: `hashcreate << EOF F1 F2 F3 F4 F5 EOF`

**Related Commands**

    **Section 19.37, "HASHDELETE command"**

    **Section 19.38, "HASHLIST command"**

    **Section 19.39, "HASHVERIFY command"**

    The **-c** option for **put** and **get** commands

# 19.37. HASHDELETE command

**Synopsis**
    **hashdel**[**ete**] [**-A**] [**-q**] [**-R**] *path* …
**Description**
    This command removes the checksum hash for HPSS files.
**HSI Version**
    4.0.1.1 and later.
**Aliases**
    None
**Options**

    **-A**  Display absolute pathnames.

    **-q**  Quiet mode - do not list info for deleted hash entries.

    **-R**  [standard option] Recursively delete hash entries for files in the specified path(s).

**Keywords**
    None
**Usage Notes**

1. The **hashdelete** command is only available when communicating with an HPSS server.
2. Setting, updating or deleting an ACL can be done by the object's owner without requiring any special permission (other than read if the **hashcreate** command is used). For all others, it requires either "group" or "other" write permission on the object.

**Examples**
    None
**Related Commands**

    **Section 19.36, "HASHCREATE command"**

    **Section 19.38, "HASHLIST command"**

**Section 19.39, "HASHVERIFY command"**

The **-c** option for **put** and **get** commands

# 19.38. HASHLIST command

**Synopsis**
 **hashli**[**st**] [**-A**] [**-a**] [**-R**] *path* …
**Description**
 This command displays the checksum hash for HPSS files.
**HSI Version**
 4.0.1.1 and later.
**Aliases**
 **lshash**
**Options**

  **-A** Display absolute pathnames.

  **-a** List all fields associated with the hash, one per line.

  **-R** [standard option] Recursively list hash entries for files in the specified path(s).

**Keywords**
 None
**Usage Notes**
 None
**Examples**
 None
**Related Commands**

 **Section 19.36, "HASHCREATE command"**

 **Section 19.37, "HASHDELETE command"**

 **Section 19.39, "HASHVERIFY command"**

The **-c** option for **put** and **get** commands

# 19.39. HASHVERIFY command

**Synopsis**
 **hashver**[**ify**] [**-A**] [**-C**] [**-F on|off**] [**-H** *type*] [**-h**] [**-N**] [**-S**] [**-T on|off**] [**-Y** *style*] *path* … [**<<MARKER**]
 …
**Description**
 This command verifies the checksum hash for existing HPSS file(s). Verification is performed by reading
 the files from HPSS, recreating the checksum hash, and comparing it with the checksum hash stored in
 HPSS metadata.
**HSI Version**
 4.0.1.1 and later.
**Aliases**

None

**Options**

| | |
|---|---|
| **-A** | Enable auto-scheduling of retrievals. |
| **-C** | Cache purge option. If specified, purge files from HPSS disk cache after a successful hash create. Normally used only when it is expected that files will only be fetched once (or infrequently), to help optimize disk cache use. |
| **-F** | Explicitly use (**on**) or disable (**off**) Local File Mover I/O, overriding the settings based upon file size and the global enable/disable flag from the hsirc file. |
| **-H** *type* | Set case-insensitive checksum hash type to use (none, sha1, sha224, sha256, sha384, sha512, md5, crc32, adler32). Overrides hsirc and compile time options. |
| **-h** | Symlink option. If specified, ignores HPSS symlink. If not set, reads through HPSS symlinks when reading files. |
| **-N** | Disable auto-scheduling of retrievals (inverse of option **-A**). The normal default is to enable auto-scheduling in order to optimize tape mounts and tape positioning. |
| **-R** | [standard option] Recursively create hash entries for files in the specified HPSS path(s). |
| **-S** | Disable staging of the source file (attempts to read directly from tape). |
| **-T** | [future] Explicitly use (**on**) or disable (**off**) Transfer Agent I/O. |
| **MARKER** | Provides sh-style HERE document syntax for specifying filenames. |

- **MARKER** may optionally be preceded by one or more spaces following the here document sentinel **<<**.
- Lines following **MARKER**, which must be the last item on the line, contain pathname(s) (which may include wildcards).
- The list is terminated by a line containing **MARKER** as the first token on the line.

**Keywords**

None

**Usage Notes**

1. The **-A** option is usually the default, but may be disabled in the hsirc file.
2. The HPSS Transfer Agent does not currently support checksum hashes.

**Examples**

1. Verify the checksum hashes for all `.ksh` files in the current directory: `hashverify *.ksh`
2. Verify checksums for the files F1 through F5 using HERE-document syntax. Free up disk cache entries after the files have been verified. `hashver -C << EOF F1 F2 F3 F4 F5 EOF`

**Related Commands**

**Section 19.37, "HASHDELETE command"**

**Section 19.38, "HASHLIST command"**

**Section 19.36, "HASHCREATE command"**

The **-c** option for **put** and **get** commands

# 19.40. HELP command

**Synopsis**
> **help**

**Description**
> This command displays brief help information.

**HSI Version**
> All

**Aliases**
> None

**Options**
> None

**Keywords**
> None

**Usage Notes**
> None

**Examples**
> None

**Related Commands**
> None

# 19.41. HISTORY command

**Synopsis**

> **hist**[**ory**] [**-e** *ename*] [**-nlr**] [*first* [*last*] ]

> or

> **hist**[**ory**] **-s** [*old=new*] [*cmd_string*]

**Description**
> This command provides command line history and re-execution.

**HSI Version**
> 3.4.1

**Aliases**
> r (alias for **hist -s**)

**Options**

**-**
**e**   Specifies editor name to use with first form of the command. If not specified, then the environment variables "HISTEDIT" or "FCEDIT" are checked, and if neither is set, then `/bin/edit` is used.

**-l**   Specifies <listing> action to display the commands on the listing file.

**-**
**n**   Suppresses command numbers when editing.

**Keywords**
       None
**Usage Notes**

1. The first form provides for listing or editing a range of command as described below, and the second form provides for re-executing the most recent command that starts with the string *cmd_string* (or the most recent command if *cmd_string* is not given), and optionally substituting the first occurrence of *old* with *new*.

**Examples**

1. Lists history: `hist`
2. Re-executes command number 60: `r 60`
3. Re-executes the last **ls** command, substituting the first occurrence of the string "HSI" with the string "HTAR": `r HSI=HTAR ls`

**Related Commands**
       None

# 19.42. ID command

**Synopsis**
       **id** [**-u** | **-G** | **-g** [**-n**]] [*user*]
**Description**
       This command displays the current login identity (initial login, "su" login, or "sudo" login).
**HSI Version**
       3.4
**Aliases**
       None
**Options**

   **-G**    Causes group list for current ID to be displayed.

   **-g**    Causes group ID to be displayed.

   **-u**    Causes user ID to be displayed.

**-n**     Causes **-g** or **-u** option to display names instead of numbers.

*user*  If *user* is specified, causes info to be displayed for the specified user. If not specified, info for the current login or SU identity is displayed.

**Keywords**
       None
**Usage Notes**

1. Only one of the **-G**, **-g**, and **-u** options may be specified.
2. The **-G** option applies only to the current login or SU/SUDO identity.
3. If no options are specified, the current login name and ID, group name and ID, and group list are displayed in the form: `uid=123(name) gid=123(name) groups=123(name)[, 234(name) …]`

**Examples**

1. List current identity info: `id`
2. List current login user ID: `id –u`
3. List current login group ID: `id –g`
4. List current login group name: `id –ng`

**Related Commands**

**Section 19.87, "SU command"**

**Section 19.88, "SUDO command"**

**Section 19.95, "WHOAMI command"**

# 19.43. IDLETIME command

**Synopsis**
       **idle**[**time**] [*value*]
**Description**
       This command sets the idle timeout value.
**HSI Version**
       All
**Aliases**
       None
**Options**

*value*  Optional idle timeout value, in seconds. If *value* is negative; that is, `timeout –1`, then an infinite timeout is used.

If *value* is not specified, the current idle timeout setting is displayed. The default value for this setting is 1800 seconds (1/2 hour).

**Keywords**
       None
**Usage Notes**

1. If command line editing is enabled at HSI build time, this command currently has no effect.
2. In general, use of an infinite timeout is discouraged, as it ties up system resources on both the client and server machines.

**Examples**

1. Display current idle timeout setting: `idletime`
2. Set idle timeout to one hour (3600 seconds): `idle 3600`
3. Set unlimited timeout: `idle -1`

**Related Commands**
**Section 19.81, "SET command"**

# 19.44. IN command

**Synopsis**
    **in** *local_filepath*
**Description**
    This command reads subsequent command input, up to the EOF or until END is encountered, from a local command file.
**HSI Version**
    All
**Aliases**
    None
**Options**
    None
**Keywords**
    None
**Usage Notes**

1. The only restriction on commands read from IN files is that no **in** command is allowed; for example, IN files may not be nested. All other command options, including wildcards for both local and HPSS files, are handled exactly the same as for interactive input.

**Examples**
    None
**Related Commands**
    None

# 19.45. KEEP command

**Synopsis**
    **k**[**eep**] *keyset*
**Description**
    This command stores the current set of keyword values from the active keyset into a local file in the `~/.hsikeysets` directory.
**HSI Version**
    All
**Aliases**
    None
**Options**

**-l**   Save keyset in local file space (not implemented).


**-h**   Save keyset in HPSS space (default).

**Keywords**
　　None
**Usage Notes**

　　　1. This command is deprecated and will be removed in a future release.

**Examples**
　　None
**Related Commands**

　　　**Section 19.1, "ADOPT command"**

　　　**Section 19.85, "SHOW command"**

　　　**Section 19.31, "FREE command"**

# 19.46. LCD command

**Synopsis**
　　**lcd** *localpath*
**Description**
　　This command issues a local "change directory" (CD) command on the local host.
**HSI Version**
　　All
**Aliases**
　　None
**Options**
　　None
**Keywords**
　　None
**Usage Notes**
　　None
**Examples**
　　None
**Related Commands**

　　　**Section 19.48, "LLS command"**

　　　**Section 19.47, "LCDLS command"**

　　　**Section 19.52, "LPWD command"**

# 19.47. LCDLS command

**Synopsis**
　　**lcdls** [*options*] *localpath*
**Description**

This command issues a local "change directory" (CD) command on the local host, followed by a "local list directory" (LLS) command.

**HSI Version**
All
**Aliases**
None
**Options**

*options*  Local list options. These may differ, depending upon the local host's version of UNIX.

**Keywords**
None
**Usage Notes**

1. This command combines the common operations of changing to a local directory, then listing its contents, into a single command.
2. Note: wildcard characters in pathname are currently not expanded.

**Examples**
```
lcdls /usr/local/bin
```
**Related Commands**

**Section 19.48, "LLS command"**

**Section 19.46, "LCD command"**

**Section 19.52, "LPWD command"**

# 19.48. LLS command

**Synopsis**
**lls** [*list options*] [*localpath …*]
**Description**
This command lists local files by issuing the **ls** command.
**HSI Version**
All
**Aliases**
None
**Options**

*options*  Local list options. These may differ, depending upon the local UNIX version.

**Keywords**
None
**Usage Notes**
None
**Examples**
None
**Related Commands**

**Section 19.47, "LCDLS command"**

**Section 19.46, "LCD command"**

**Section 19.52, "LPWD command"**

# 19.49. LMKDIR command

**Synopsis**
> {**lmkdir** | **lmd**} [**-p**] *localpath* …

**Description**
> This command makes a local directory.

**HSI Version**
> All

**Aliases**
> **lmd**

**Options**

> **-p**  Create any missing intermediate subdirectories in *local_path*.

**Keywords**
> None

**Usage Notes**
> None

**Examples**
> lmkdir –p new_subdirectory/another_subdir

**Related Commands**

> **Section 19.48, "LLS command"**

> **Section 19.46, "LCD command"**

> **Section 19.52, "LPWD command"**

# 19.50. LN command

**Synopsis**

> **ln** [**-f**] [**-h**] [**-n**] [**-s**] *srcname target*

> **ln** [**-f**] [**-h**] [**-n**] [**-s**] *srcname* [*srcname* … ] *targetdir*

**Description**
> This command creates a link (hard or symbolic) from one HPSS node to another.

**HSI Version**
> All

**Aliases**
> None

**Options**

> **-f**  Unlink any already existing file, permitting the link to occur.

> **-**   If the *target* or *targetdir* is a symbolic link, do not follow it. This is most useful with the **-f** option, to
> **h**   replace a symlink which may point to a directory.

**-**    Same as **-h**, for compatibility with other **ln** implementations.
**n**

**-s**  Create a symbolic link.

**Keywords**
>DIRn

**Usage Notes**

1. *srcname* must be an existing node; *target* must be a non-existent node unless it is a directory, or **-f** is specified.
2. If the second form of the command is used, where *targetdir* is specified, then the last component of *srcname* is used when creating the target symlink. For example, assuming directory toms_files exists: ln -s /home/users/tom/testfile someOtherFile toms_files would create the links: toms_files/testfile → /home/users/tom/testfile toms_files/someOtherFile → someOtherFile
3. To create a link in the current directory that points to the same object in *targetdir*, the correct form is: ln -s targetdir/srcname srcname resulting in: srcname → targetdir/srcname

**Examples**
>None

**Related Commands**
>None

# 19.51. LOG command

**Synopsis**

>**log** -

>**log [> | >!]** *localpath*

>**log** >> *localpath*

**Description**
>This command creates a time stamped history file of all HSI commands and responses.

**HSI Version**
>All

**Aliases**
>None

**Options**

| | |
|---|---|
| **-** | ("minus" character) Disable logfile mode. The *localpath* is ignored. |
| **>**, **>!** or no option preceding *localpath* | Write loggable output to *localpath*. |

**>>**                                    Append loggable output to *localpath*.

**Keywords**
 None
**Usage Notes**
 None
**Examples**
 None
**Related Commands**

  **Section 19.44, "IN command"**

  **Section 19.69, "OUT command"**

# 19.52. LPWD command

**Synopsis**
 **lpwd**
**Description**
 This command prints the current local working directory for the client host.
**HSI Version**
 All
**Aliases**
 None
**Options**
 None
**Keywords**
 None
**Usage Notes**
 None
**Examples**
 None
**Related Commands**

  **Section 19.48, "LLS command"**

  **Section 19.46, "LCD command"**

# 19.53. LS command

**Synopsis**
 {**ls** | **l**[**ist**] | **dir**} [*options*] [*path …*]
**Description**
 This command lists HPSS nodes.
**HSI Version**
 All
**Aliases**
 **list**, **dir**
**Options**

**-a**    List all entries, including "hidden" files whose names begin with ".".

**-c**    Use time of last modification for sorting (deferred).

**-d**    If file is a directory, list its name instead of its contents.

**-j**    Displays output in JSON format.

**-l**    (letter "ell") Long list format.

**-p**    Put a slash after each name if the file is a directory (deferred).

**-r**    Reverse alpha or age sort order, as appropriate (deferred).

**-s**    Display size as well as name if **-1** (numeral 1) option used.

**-u**    Use time of last access for sorting instead of last modification.

**-x**    Multicolumn output format, with entries sorted across page.

**-A**    Print annotation info.

**-C**    Multicolumn output format, with entries sorted down the columns.

**-D**    Displays the full date/time stamp.

**-F**    Puts a backslash ("/") after directory filenames, or asterisk ("*") if executable (deferred).

**-**    Print headings on long listings.
**H**

**-**    Print unordered "-l" or "-1" format listings.
**O**

**-P**   Prints one line per node with volume/position/VSN list/COS/file family/subsystem info. The fields on the line are tab-separated, as follows:

- Object type: "FILE", "HARDLINK", "DIRECTORY", "SYMLINK", "JUNCTION", "FILESET", "NS_FSET_ROOT".
- Object name.

For FILE and HARDLINK objects:

- File size (in bytes).
- Amount of data at first tape storage class (in bytes), or 0 if no tape level with data found.
- Position and relative offset of file on first tape level, first VV (1 = first file on tape). 0 if no tape levels with data were found in the hierarchy. The value is displayed as position+relative offset.
- Comma-separated list of VSNs for first (or only) tape level, or empty string if none found.
- Class of Service (COS) ID.
- File Family ID.
- Subsystem ID.
- File Creation Date in the form mm/dd/yyyy.
- File Creation Time in the form hh:mm:ss.
- File Last Modification Date in the form mm/dd/yyyy.
- File Last Modification Time in the form hh:mm:ss.

**-R**   Recursively list directories.

**-S**   Display sizes in the form XX.Y *sfx*, where *sfx* is "B", "KB", "MB", "GB", "TB", "PB", or "XB". For these suffix values, 1K = 1000. If specified, this option also enables the **-l** option.

**-T**   "type" where type is one of w, r, c, or m. This allows the user to specify which HPSS time value is displayed when one of the "long list" options (for example, `ls -l`) is used. The meaning of the letters are: w - last write time; r - last read time; c - file creation time; and m - time of last modification (default).

**-U**   Print HPSS-specific information.

**-V**   Print volume info for first tape level. Enables **-l** option.

**-**    Print volume info for first disk level. Enables **-l** option.
**W**

**-X**  Print extended volume info (for all levels). Enables **-l** option.


**-1**  (numeral 1) Forces one name-per-line list.

## Keywords
DIRn
## Usage Notes

1. List options are available to cause the output to be formatted in a variety of ways. They may be specified individually (for example, **-C**) or as a string (for example, **-1R**). See the examples below for descriptions of the fields for the **-P**, **-V**, and **-X** options.

## Examples

1. Simple long listing **ls -l** command. This produces normal UNIX-style output: `ls –l test.tar.bz2 –rw–r–r–– 1 user1 14 437276618 May 28 2003 test.tar.bz2`
2. List the information for the first tape level in the hierarchy in which the file resides. `ls –V somefile` produces output similar to this: `ls –V test.tar.bz2 –rw–r–r–– 1 user1 14 6002 legacy TAPE 437276618 May 28 2003 test.tar.bz2 Storage VV Stripe Level Count Width Bytes at Level ––––––––––––––––––––––––––––––––––––––––––––––––––––––––––––––––––––––– ––––––––– 1 (tape) 1 1 437276618 VV[ 0]: Object ID: b730e850–a89f–11e0–98d6– 10005afa75bf ServerDep: 6a61f9a2–fd36–11d0–93cb–10005afa75bf Pos: 7323 PV List: X3086300`

3. List file tape/position information on a single line: `ls –P test.tar.bz2 FILE /home/user1/test.tar.bz2 437276618 437276618 7323+0 X3086300 6002 0 1 05/28/2003 21:03:13 07/12/2011 09:53:21` This is sometimes used for sorting files by tape/position. Fields in the listing are separated by tabs. The fields are as follows:

    - type (FILE, DIRECTORY, SYMLINK, JUNCTION)
    - object name
    - size
    - bytes at 1st tape level
    - tape position+offset (offset=0: not in aggregate or first file in aggregate)
    - first or only tape VV cartridge(s) for file
    - class of service
    - file family ID
    - subsystem ID
    - create time
    - modification time

## Related Commands
None

# 19.54. LSACL command

## Synopsis
**lsacl** [**-A**] [**-e**] [**-ic**] [**-io**] [**-R**] *filelist*
## Description
This command displays HPSS Access Control List entries for files and directories.
## HSI Version
4.0.1.3 and later.

**Aliases**
  None
**Options**

  **-A**      Display absolute pathnames in messages.

  **-e**      Show the effective permissions after applying any mask object.

  **-ic**     Show the Initial Container ACL (only applies to directories).

  **-io**     Show the initial object ACL (only applies to directories).

  **-R**      Recursively lists ACLs for subdirectories.

  *filelist* A space-delimited list of one or more paths to be listed. Pathnames can contain wildcard characters.

**Keywords**
  None
**Usage Notes**

  1. HPSS ACLs were originally based on the Distributed Computing Environment (DCE), and are of the general form: `type:key:perms`, where:

     - "type" is one of the standard DCE ACL types, such as user, group, user_obj, group_obj, and so on.
     - "key" is of the general form *principal@realm*, but may be different depending on the ACL type.
     - "perms" are made up of the letters "rwxcid" for read, write, execute, control, insert and delete.
  2. See the Access Control List (ACL) section of this reference manual for more information on HPSS ACLs.
  3. Both **-ic** and **-io** can be specified for this command unlike the **chacl** command, which only operates on one ACL type per command.
  4. The **cp** command provides an option for copying ACLs.
  5. This command is only available when communicating with an HPSS server. It is not available in the non-HPSS gateway.

**Examples**

  1. List all of the ACL entries for files whose names end with `.tar`: `lsacl *.tar`
  2. List the Initial Container ACL for subdirectories `Project1`, `Project2`, and `Project3`: `lsacl -ic Project[123]`
  3. Recursively list all of the object ACL entries for the files in directory `Laser3`: `lsacl -R Laser3`

**Related Commands**
  **chacl** command, **cp** command ACL options **-c**, **-ic**, **-io**

# 19.55. LSCONNECTIONS command

**Synopsis**
> {**lscon[nections]** | **showc[on]**}

**Description**
> This command displays a list of active HPSS server connections.

**HSI Version**
> All

**Aliases**
> **showcon**

**Options**
> None

**Keywords**
> None

**Usage Notes**

> 1. The output from this command is a list of the currently active connections. The current active connection is marked by the arrow symbol ("->") preceding the logical drive letter.

**Examples**

> lscon

> An example output is shown below:

```
C:[hpss]/home/user1->lscon
List of Currently Active Connections
Current default connection handle: 3
Han- Remote Remote HSM Fire- I/O Ctl Data Port Cmd
Drive dle IP Address Hostname Level wall Mode Port Min Max Count
 ------------------------------------------------------------------------------
S: 1 <IPv4 addr8> hpss09i.site8 4.1.4 off normal 1219 0 65535 6
HomeDir: /home/user1
pwd0: /home/user1
->C: 3 <IPv4 addr9> hpss.site9 4.1.2 on extended 1219 0 65535 3
HomeDir: /home/user1
pwd0: /home/user1
```

> The columns in the display are:

> - Drive - logical drive letter assigned to the connection.
> - Handle - internally assigned handle for the connection. This field is used for debugging, and is normally not useful in any other context.
> - Remote IP Address - Internet address of the server host for the connection.
> - Remote Hostname - Hostname of the server host for the connection.
> - HSM Level - The release level of the Hierarchical Storage Manager system that HSI is communicating with. (For release 2.6 of HSI, the HSM system is HPSS).
> - Firewall - Displays the current firewall setting, which controls the I/O mode that HSI uses for data transfers.
> - I/O mode - Will be "normal" if firewall mode is "off", and "extended" if firewall mode is "on".
> - Ctl Port - the port number used for the TCP/IP connection to the server.
> - Data Port (min,max) - displays the port range used if HSI is operating in an environment that makes use of restricted ports for data transfers.
> - Cmd Count - a count of HSI commands that have been issued for this connection.

Related Commands

Section 19.30, "FIREWALL command"

Section 19.59, "LSSITES command"

Section 19.83, "SETDRIVE command"

19.56. LSCOS command

Synopsis
> **lscos** [**-n**] [**-N** *coslist_name*] [*drive …*]

Description
> This command, with no arguments, lists general COS info.

HSI Version
> All

Aliases
> None

Options

> **-n**   Shows Named COS Lists available for the connection.

> **-N**   List COS IDs associated with the specified Named COS List.

> *drive* One or more logical drive IDs; for example, `lscos A: C:`. Default is the current connection.

Keywords
> DIRn

Usage Notes
> None

Examples
> `lscos`

Related Commands
> **Section 19.15, "COSID keyword"** keyword

19.57. LSFILESETS command

Synopsis

> **lsfiles**[**ets**] **-?** (obtain interactive usage information)

> **lsfiles**[**ets**] [**-l**] [*drive …*] [*filesetname*]

Description
> This command lists filesets that have been created by the HPSS administrator.

HSI Version
> All

Aliases
> **lsfset**

Options

-l   Long listing.

drive Logical drive letter for the connection. Default is the current active connection.

Keywords
      None
Usage Notes

1. The output from this command includes the number of HPSS objects (directories, files, symbolic links and hard links) for the filesets.
2. If *filesetname* is not specified, then all filesets for the specified connection are listed.

Examples

1. `lsfset`

   `1 FilesetRoot.3109`

2. `lsfset –l`

   `Entry Fileset Name Number Type COS Family Subsys State Files Dirs Symlinks HardLinks Junctions 1 FilesetRoot.3109 HPSS_ONLY 0 0 1 RW 50964134 1590202 393675 718 0`

3. `lsfset R:`

   `HPSS Junctions for connection R, subsystem 0: 1 FilesetRoot.3109`

Related Commands
      Section 19.53, "LS command"

19.58. LSJUNCTIONS command

Synopsis

lsjun[**ctions**] **-?** (obtain interactive usage information)

lsjun[**ctions**] [**-s** *subsysID*] [*drive …*]

Description
      This command lists junctions and their related filesets that have been created by the HPSS administrators for HPSS systems.
HSI Version
      All
Aliases
      None
Options

-s   Subsystem ID (default = 1).

drive Logical drive letter for the connection. Default is the current active connection.

Keywords
    None
Usage Notes

1. The output from this command includes the number of HPSS objects (directories, files, symbolic links and hard links) for the fileset to which the junction is linked.

Examples

1. `lsjunc`

   Junction Name: ./home/bbd/site4 Fileset Name: FilesetRoot.3281 Fileset Type: HPSS_ONLY Class of Service: 0 Directories: 468 Files: 35595 Symlinks: 2 Hard Links: 0 Junctions: 1

Related Commands
    Section 19.53, "LS command"

19.59. LSSITES command

Synopsis
    lssi[tes] [**-a**] [**-c**] [**-h**] [**-m**] [**-p**] [**-s** *site*[,*site* …] [*drive* …]
Description
    This command displays a list of sites parsed from the hsirc files. The site names may be used with the **-s** sitename option in the HSI startup command line, and with the **-s** sitename option for the **open** command.
HSI Version
    All
Aliases
    None
Options

-a   List all info for sites.

-c   List server host name and connection info.

-h   List just server host info.

-m   List authentication method(s) defined for this site.

-p   List principal info for the site.

-s   List info for the specified sites. The parameter may contain either a single site name, or a comma-separated list of sitenames. Sitenames are case sensitive.

drive   Optional space-separated list of drive letters for which the site info is to be displayed; for example, `a: z:`.

Keywords
    None
Usage Notes

1. If no site or drive options are specified, info for all sites will be displayed. If both **-c** and **-h** are specified, only **-c** info will be displayed.

Examples
    `lssi`
Related Commands

Section 19.46, "LCD command"

Section 19.48, "LLS command"

19.60. MDELETE command

Synopsis
    mdel[ete] [**-i**] [**-R**] *path* …
Description
    This command removes files (same as the DELETE command), with an interactive prompt to confirm deletion of each file.
HSI Version
    All
Aliases
    None
Options

  -i   Interactively confirm each deletion (regardless of current "prompt" setting).

  -R   [standard option] Recursively remove directories in the specified path(s).

Keywords
    None
Usage Notes

1. Interactive prompting can be toggled with the PROMPT command.

Examples
    `mdel *`
Related Commands
    None

19.61. MGET command

Synopsis

mget [**-B** | **-b**] [**-C**] [**-h**] [**-L** *ofs path*] [**-N**] [**-o**] [**-O** *srcofs*:*sinkofs*:*length*] [**-P** | **-p**] [**-R** | **-r**] [**-S**] [**-t**] [**-U** | **-u**] [*path* …] [**<<MARKER**]

Description

This command retrieves a copy of a file from HPSS to your local file space on the host system (same as **get** command), with an interactive prompt to confirm the "get" of each file.

HSI Version

All

Aliases

None

Options

-B | **-b**	Backup option. Renames any existing local file by appending "~". (This option may not be used with **-L**, **-t**, or **-O**.)
-C	Cache purge option. Purge files from HPSS disk cache after a successful **get**. Normally used only when it's expected that files will only be fetched once (or infrequently), to help optimize disk cache use.
-h	Symlink option. Create local symlink if HPSS symlink. If not set, reads through HPSS symlinks when copying files to the local filesystem.
-L	Pathname of a file containing lists of partial file offsets, in the form specified by **-O** (see below). Multiple entries per line may be specified. All **-O** options are processed before any **-L** entries.
-N	Disable auto-scheduling of retrievals. The default is to schedule retrievals in an optimal way so as to minimize HPSS tape mounts.
-O	Partial file transfer. *srcofs* is the source file offset, *sinkofs* is the sink file offset, *length* is the transfer length. Missing options use defaults (src: 0, sink: 0, length: entire file). "CP" or "0CP" (case-insensitive) can be used to specify "current position". Multiple **-O** options may be specified, and are processed in order. Offsets may specified with case-insensitive suffixes such as "k", "kb", and so on.
-P | **-p**	Preserve timestamp. Attempts to copy HPSS file's timestamp to local file.
-R | **-r**	Recursively fetch directories.
-S	Disable staging of the source file (attempts to read directly from tape). Users having trouble retrieving files over 25 GB in size may wish to use the **-S** option for the **get** or **mget** commands. The **-S** option was added to allow users to disable staging files from tape, specifically for this type of situation.

-t   The "re-get" option. Restart a previously failed operation. This is equivalent to: −O
              sizeof(localfile):sizeof(localfile):0. Note: **-t** may not be used with **-O** or **-L**.

-U | -u   Update option. Only copy file from HPSS if local file timestamp is older.

MARKER   Provides sh-style HERE-document syntax for specifying filenames. Lines following the
              MARKER, which must be the last token on the line, contain pathname (which may include
              wildcards). The list is terminated by a line containing MARKER as the first token on the
              line. If **-O** or **-L** is specified, offsets apply only to the first file transferred.

Keywords
     None
Usage Notes

     1. Interactive prompting can be toggled with the PROMPT command.

Examples
     mget *
Related Commands
     None

19.62. MIGRATE command

Synopsis
     mig[**rate**] [**-h**] [**-L** *level* | **-l** *level*] [**-F** | **-f**] [**-P** | **-p**] [**-R**] [**-S**] *path* …
Description
     This command migrates one or more HPSS files from disk cache to lower levels of hierarchy.
HSI Version
     All
Aliases
     None
Options

-h   Treats symlinks as normal files. Default is to skip symlinks.

-L|-l Specifies hierarchy level from which to migrate data. The valid range is 0 (default) to 4.

-F|-f Force migration even if not needed.

-R   [standard option] Recursively traverse directories in the specified path(s).

-P   Purge file from disk cache after migration.

-S   Allow staging of file if no disk data.

Keywords
None
Usage Notes

1. If **-F** (force) is specified, the affected file(s) will be migrated regardless of normal disk residency criteria.
2. The **-S** option was added to provide a way to recover data from a second tape copy.
3. The underlying HPSS call applies only to open files, so note that each file to be migrated by this command will need to be opened. Use recursion and wildcards sparingly.
4. The use of this command may be restricted at some HPSS sites.

Examples
migrate –F my_file
Related Commands

Section 19.74, "PURGE command"

Section 19.86, "STAGE command"

19.63. MKDIR command

Synopsis
mkd[**ir**] [**-A** *annotation*] [**-m** *mode*] [**-p**] *path …*
Description
This command creates HPSS subdirectories.
HSI Version
All
Aliases
add, md
Options

-   Annotation string to set on newly created or preexisting directories. If **-p** is specified, and
A   intermediate subdirectories are created, this option only applies to the final subdirectory in the path(s).

-   Mode to use for creation of the one or more directories. This must be an octal value in the range 0-
m   7777. The mode value is silently truncated if it exceeds this range.

-p   Creates missing intermediate path name directories. If the **-p** flag is not specified, the parent directory of each newly-created directory must already exist.

Keywords
DIRn
Usage Notes

1. The **mkdir** command ignores any directory parameter that names an existing directory. No error is issued.

Examples

    mkdir Source Bin Include Doc

Related Commands

   [Section 19.79, "RMDIR command"](#)

19.64. MPUT command

Synopsis

   mput [**-A** *string*] [**-B** | **-b**] [**-d**] [**-F on|off**] [**-h**] [**-M** *mode*] [**-n** *days*] [**-P** | **-p**] [**-Q**] [**-R** | **-r**] [**-T on|off**] [**-t**]
   [**-U** | **-u**] *path* …

Description

   This command saves or replaces a file (same as **put** command), with interactive prompting for each
   specified path. If the HPSS file does not exist, a new file is created; if the HPSS file exists, it is
   overwritten.

HSI Version

   All

Aliases

   None

Options

   -A   Annotation string for HPSS file (type annotate ? for more details).

   -B|-   Backup option. Renames existing HPSS file by appending "~".
   b

   -d   Remove local files after a successful transfer to HPSS.

   -F   Enable (**on**) or disable (**off**) use of the HPSS Local File Mover, overriding any configuration
             settings in the HSIRC file(s).

   -h   Symlink option. Create symlink in HPSS if local symlink. If not set, reads through local symlinks
             when copying files to HPSS.

   -M   Specifies with *mode* the octal mode to use for file creations.

   -n   Only put files modified within last *days* number of days.

   -P   Create intermediate HPSS subdirectories for the file(s) if they do not exist.

   -p   Preserve timestamp. Attempts to copy local file's timestamp to HPSS file.

-Q   If running as the HPSS "root" user, preserve the local owner/group for HPSS files and directories that are created.

-R|-r   Recursively store directories.

-T   Enable (**on**) or disable (**off**) use of the HPSS Transfer Agent, overriding any configuration settings in the HSIRC files.

-t   The "re-put" operation. Restart a previously failed operation, using the size of the existing HPSS file as the starting offset.

-U|-u   Update option. Only copy file to HPSS if local file timestamp is newer.

Keywords
    BACKUP DIRn
Usage Notes

    1. Interactive prompting can be toggled with the **prompt** command.

Examples
    None
Related Commands

    Section 19.17, "CPUT command"

    Section 19.32, "GET command"

19.65. MV command

Synopsis

    {**mv** | **move**} [**-f**] [**-i**] [**-v**] *path1 path2*

    or

    {**mv** | **move**} [**-f**] [**-i**] [**-v**] *path1 path2 … dirpath*

Description
    This command moves files within the HPSS directory structure.
HSI Version
    All
Aliases
    move
Options

-f Force option. Removes existing target if it already exists. Default is to not remove an existing target.

-i In interactive mode, pause for confirmation before removing existing target.

Keywords
    DIRn
Usage Notes

1. For the first form:

    - If *path2* exists and is not a directory, then the command fails (that is, it can not overwrite an existing node).
    - If *path2* is a directory, then the second form is assumed; otherwise, the command acts like `mv path1 path2`.
    - If *path1* is a directory, then *path2* must also be a directory.

2. For the second form:

    - *dirpath* must be an existing directory node or the command fails.
    - All commands are issued as "mv pathname dirpath/pathname".
3. Note that cross-HPSS moves are not yet implemented. All paths must reference the same HPSS system.

Examples
    `mv *.c Source`
Related Commands

Section 19.16, "CP and CCP commands"

Section 19.77, "RENAME command"

19.66. NEWACCT command

Synopsis
    newa[**cct**] [*acctID* | *accountName*]
Description
    This command changes the active account ID.
HSI Version
    All
Aliases
    None
Options

acctID   Optional new account ID to start using.

accountName Optional new account name to start using.

If the account ID (or account name) is not specified, the current account ID is displayed.

Keywords
    None
Usage Notes

> 1. If the HPSS site is not using site accounting, this command is disabled.

Examples
    `newa 200`
Related Commands
    Section 19.7, "CHACCT command"

19.67. NEWGRP command

Synopsis
    newgrp *GID | groupName*
Description
    This command changes the active group ID.
HSI Version
    All
Aliases
    None
Options

> *GID*   Numeric group ID to start using.

> *groupName*   Optional new group name to start using.

Keywords
    None
Usage Notes

> 1. The GID or group name must be a member of the user's group set, as shown by the **groups** command.
> 2. New files and directories created after a successful **newgrp** command will have group ownership set to the new GID, unless otherwise overridden by the directory's SET_GID flag, which forces GIDs to be set to that of the parent directory.

Examples
    `newgrp staff` or `newgrp 10438`
Related Commands
    Section 19.35, "GROUPS command"

19.68. OPEN command

Synopsis
    {**open** | **conn**[**ect**]} [**-A** *auth_method*] [**-c** *filename*] [**-d** *driveletter:*] [**-h** *remote*] [**-k** *file*] [**-l** *principal*] [**-p** *portnumber*] [**-q**] [**-s** *sitename*]
Description
    This command opens a new HPSS server connection.
HSI Version
    All

Aliases

connect

Options

-   Use the specified authentication type; for example, "DCE".
A

-c   *filename* is the credentials cache filename (Kerberos).

-d   Virtual drive letter to which this connection will be mapped. Note that the colon (":") is optional.

-h   *remote* is a remote host name or IP address.

-k   *file* is a password file or keytab file.

-l   Log in as *principal* on the remote server; for example, `-l starbuck`.

-p   Remote port number on which to attempt a connection. Default is 1219.

-q   "Quiet" mode (no Message of The Day).

-s   Name of remote site, as specified in a stanza within the hsirc file. A list of sitenames defined in the hsirc or ~/.hsirc file can be obtained by using the **lssite** command.

Keywords

None

Usage Notes

1. The virtual drive letter provides a convenient shorthand method of accessing files on the remote system; for example: `A:/remote_hpss/pathname`
2. If one or both of the **-s** or **-d** options are used, then HSI will attempt to locate a stanza within the hsirc file that contains information to help automate the login process to the remote site. If possible, settings for the drive letter, port, server host, login principal, authentication method, and related files (for example, the Kerberos credentials cache pathname) will be used if specified in the hsirc stanza.
3. Any or all parameters contained in the hsirc stanza can be overridden by command line options.
4. If **-d** is not specified and cannot be found in an hsirc stanza, then HSI will choose the lowest (alphabetically sorted without regard to case) available drive letter.

Examples

```
open -l queequeg -A kerberos -d A hpss.remote.host
```

Related Commands

Section 19.90, "SWITCH command"

Section 19.55, "LSCONNECTIONS command"

Section 19.13, "CLOSE command"

19.69. OUT command

Synopsis
> **out - out** *local-file* **out >** *file* **out >|** *file* **out >>** *file*

Description
> This command directs listable output from HSI commands to a local file.

HSI Version
> All

Aliases
> None

Options
> None

Keywords
> None

Usage Notes

1. **out -** closes any current output file and reverts to terminal output.
2. **out** *file*, **out >** *file*, and **out >|** *file* all attempt to open and write to a local file called *file*, overwriting any previous contents.
3. **out >>** *file* attempts to append to an existing file.

Examples
> ? out > /tmp/list.out ? ls –l ? out –

Related Commands

Section 19.44, "IN command"

Section 19.51, "LOG command"

19.70. PLOCK command

Synopsis
> **plo**[**ck**] [**-h**] [**-L**] [**-P**] [**-R**] [**-s**] *path* …

Description
> This command marks files as non-purgeable from the disk cache in which they reside.

HSI Version
> All

Aliases
> None

Options

  -h Read through symlinks as if they were regular files and directories. Default: ignore symlinks.

  -L Operate on normal PURGELOCK flag (the default if neither **-L** nor **-P** is specified).

-P Operate on the SUPER PURGELOCK flag (this option is restricted to authorized users).

-R Recursively traverse directories in the specified path(s).

-s Prestage files if necessary prior to locking.

Keywords
> None

Usage Notes

> 1. **-L** and **-P** are mutually exclusive options.
> 2. This command should be used judiciously, as it may cause the disk cache to fill up with files that are not purgeable. Some sites may choose to disable this command; check with your site administrator to determine the local site policy for the use of this command.

Examples
```
plo –R aproject_dir
```
Related Commands

> **Section 19.73, "PUNLOCK command"**

> **Section 19.53, "LS command"** (**-U** option)

19.71. PROGRESS command

Synopsis
> **progress** [**-q**]

Description
> This command toggles the display of file transfer progress.

HSI Version
> All

Aliases
> None

Options

> **-q** Use the **-q** (quiet) command line option to disable progress messages.

Keywords
> None

Usage Notes
> None

Examples
> None

Related Commands
> None

19.72. PROMPT command

Synopsis

prompt
Description
This command toggles the interactive response request for **mput**, **mget**, and **mdel** commands.
HSI Version
All
Aliases
None
Options
None
Keywords
None
Usage Notes

1. If the interactive terminal is unavailable (for example, within a batch job), a "YES" response is assumed for all commands.

Examples
None
Related Commands
Section 19.81, "SET command" (to set PS1/PS2 prompt strings)

19.73. PUNLOCK command

Synopsis
punl[**ock**] [**-h**] [**-L**] [**-P**] [**-R**] *path* …
Description
This command marks files as purgeable from the disk cache in which they reside. This undoes the effect of a previous **plock** command.
HSI Version
All
Aliases
None
Options

-h Read through symlinks as if they were regular files and directories. Default: ignore symlinks.

-L Operate on normal PURGELOCK flag (the default if neither **-L** nor **-P** is specified).

-P Operate on the SUPER PURGELOCK flag (this option is restricted to authorized users).

-R Recursively traverse directories in the specified path(s).

Keywords
None
Usage Notes

1. **-L** and **-P** are mutually exclusive options.
2. **-P** is restricted to authorized users.

Examples
```
punlock −R ∗.c project_dir
```
Related Commands

Section 19.70, "PLOCK command"

Section 19.53, "LS command" (**-U** option)

19.74. PURGE command

Synopsis
> **purge** [**-F**] [**-h**] [**-l** *hlevel*] [**-R**] *path* …

Description
> This command purges files from disk cache.

HSI Version
> All

Aliases
> None

Options

> **-F**   Force purge to occur. This option is only available to HPSS root administrative users.

> **-h**   Treats symlinks as normal files. Default is to skip symlinks.

> **-l**   Purge data from hierarchy level hlevel (0-4). Default is level 0.

> **-R**   [standard option] Recursively traverse directories in the specified path(s).

Keywords
> None

Usage Notes

> 1. The underlying HPSS call applies only to open files, so note that each file to be purged by this command must first be opened. Use recursion and wildcards sparingly.

Examples
```
purge −R dir1 dir2 ∗.h
```
Related Commands
> None

19.75. PUT command

Synopsis
> **put** [**-A** *string*] [**-B** | **-b**] [**-c on**|**off**] [**-d**] [**-F on**|**off**] [**-H** *algorithm*] [**-h**] [**-M** *mode*] [**-n**] [**-P** | **-p**] [**-Q**] [**-R** | **-r**] [**-T on**|**off**] [**-t**] [**-U** | **-u**] [**-X** *max*] [**-Y** *style*] *path* …

Description
> This command saves or replaces a file. If the HPSS file does not exist, a new file is created; if the HPSS file exists, it is overwritten.

HSI Version

All

Aliases

replace, send, store

Options

-A   Annotation string for HPSS file (type `annotate ?` for more details).

-B|-b   Backup option. Renames existing HPSS file by appending "~".

-c   Enable (**on**) or disable (**off**) creation of HPSS checksums, which are stored separately from the file data in HPSS. This option overrides both the default compile-time option, and the hsirc setting for enabling/disabling checksums. Checksums can also be created for existing HPSS files by using the **hashcreate** command, and can be verified either by using the **hashverify** command or by using the **get** command with **-c on**.

Important

Checksum generation is very CPU-intensive and can significantly impact file transfer performance. As much as 80% degradation in transfer rates has been observed during testing of this feature.

-d   Remove local files after a successful transfer to HPSS.

-F   Enable (**on**) or disable (**off**) use of the HPSS Local File Mover, overriding any configuration settings in the HSIRC files.

-H   Specifies the checksum hashing algorithm to use when generating a checksum (**-c on**). The default algorithm is chosen by the HPSS administrator when HSI is built, and can be overridden in the global or private hsirc files. HSI is released with the default algorithm set to md5. Valid case-insensitive possible values for this option are: "none" (no checksum), "sha1", "sha224", "sha256", "sha384", "sha512", "md5" (default), "crc32", and "adler32".

-h   Symlink option. Create symlink in HPSS if local symlink. If not set, reads through local symlinks when copying files to HPSS.

-M   Specifies octal mode to use for file creations.

-n   Only put files modified within last *days* number of days.

-P   Create intermediate HPSS subdirectories for the file(s) if they do not exist.

-p   Preserve timestamp. Attempts to copy local file's timestamp to HPSS file.

-Q   If running as the HPSS "root" user, preserve the local owner/group for HPSS files and directories that are created.

-R|-r   Recursively store directories.

-T   Enable (**on**) or disable (**off**) use of the HPSS Transfer Agent, overriding any configuration settings in the HSIRC file(s).

-t   The "re-put" operation. Restart a previously failed operation, using the size of the existing HPSS file as the starting offset.

-U|-u   Update option. Only copy file to HPSS if local file timestamp is newer.

-X   [HSI 4.0.1.2+] Set maximum concurrent transfers for this command. This may only be used to reduce, not increase, the value set by the HPSS administrator.

-Y   [HSI 4.0.1.2+] Specifies the style of checksum to search for when verifying an existing checksum (**-c** option). Valid colon-separated, case-insensitive options are: "hsi" (default), "vfs" or "hpsssum". If more than one style of checksum is specified, the precedence order is: (1) HSI-style, (2) VFS-style, (3) HPSSSUM-style.

Keywords
    FAMILY BACKUP DIRn
Usage Notes

1. This command *always overwrites existing files*, in contrast to the **cput** command.
2. See *Chapter 5, FTP compatibility and differences* for information on valid characters and renaming files on **get** and **put** commands.
3. See *Chapter 9, Using PIPEs for input and output* for information on piping files into HPSS from an operating system utility or interactively from the TTY.
4. The **-R** option may be specified to recursively STORE a directory tree. In this case, HPSS subdirectories are automatically created as needed while traversing the local directory tree.
5. See *Chapter 3, Restricted TCP ports and restricted Mover hosts* for information on dealing with firewalls.

Examples
```
put local_file_name : hpss_file_name
```
Related Commands

Section 19.17, "CPUT command"

Section 19.32, "GET command"

19.76. PWD command

Synopsis
>   **pwd**[**0-9**] [*drive: …*]

Description
>   This command prints the current working directory within the HPSS directory structure. There are 10
>   working directories, numbered 0 through 9. Directory 0 is the default directory that is referenced. The
>   commands **pwd** and **pwd0** are equivalent.

HSI Version
>   All

Aliases
>   None

Options

>   **0** - **9**   Working directory number. Default is 0.

>   *drive*   Logical drive ID (connection). Default is the current active connection.

Keywords
>   None

Usage Notes
>   None

Examples

>   1. To display the contents of working directory 4: `pwd4`

Related Commands
>   None

19.77. RENAME command

Synopsis
>   **ren**[**ame**] *srcpath sinkpath*

Description
>   This command renames an HPSS node.

HSI Version
>   All

Aliases
>   None

Options

>   *srcpath*   Source path (must already exist).

sinkpath   Sink path (must not exist).

Keywords
　　DIRn
Usage Notes

　　1. Both paths must reference the same HPSS system.

　　2. The sink path is overwritten without warning if:

　　　　a. Both paths are file objects;
　　　　b. Both paths are directories and the sink directory is empty.

Examples
　　rename file1 subdir1/file2
Related Commands
　　Section 19.65, "MV command"

19.78. RM command

Synopsis
　　{**rm** | **erase** | **del**[**ete**] } [**-i**] [**-R**] *path* …
Description
　　This command removes files.
HSI Version
　　All
Aliases
　　erase, delete
Options

　　-i   Interactively confirm each deletion (regardless of current "prompt" setting).

　　-R   [standard option] Recursively remove files and subdirectories in the specified path(s).

Keywords
　　None
Usage Notes
　　None
Examples
　　rm –i file1
Related Commands
　　Section 19.79, "RMDIR command"

19.79. RMDIR command

Synopsis
　　{**rmdir** | **rem**[**ove**] | **rd**} *path* …
Description
　　This command removes HPSS directory nodes.
HSI Version

All
Aliases
rd, remove
Options
None
Keywords
DIRn
Usage Notes

1. Directory nodes must be empty (all files and subdirectories removed before they can be deleted).

Examples
None
Related Commands
Section 19.78, "RM command"

19.80. RUNIQUE command

Synopsis
runique
Description
This command toggles creating local files with unique names.
HSI Version
All
Aliases
None
Options
None
Keywords
None
Usage Notes

1. This command is a placeholder for possible future implementation. It currently does nothing.

Examples
None
Related Commands
None

19.81. SET command

Synopsis
set [*keyword = value …*]
Description
This command sets or lists values for the keywords in the active keyset.
HSI Version
All
Aliases
None
Options
None
Keywords

BA[CKUP], COL[UMNS], COPIES, COS[ID], COSLIST, DCREATE[MODE], DIRn, FAMILY[ID], FCREATE[MODE], IOB[UFSIZE], LINES, PROMPT[EN], PS1, PS2, PWID[TH]

Usage Notes

1. If the "=" value is omitted, the current setting for the keyword is displayed.
2. The "set" part of the command may be omitted; if a command starts with "keyword = value", then HSI assumes a **set** command (see example below).

Examples

    dir2 = ~/Test/myfiles

Related Commands

See *Chapter 16, HSI keywords* in this document for details.

19.82. SETCONNECTION command

Synopsis

setc[**on**] *conid*

Description

This command sets the current active connection.

HSI Version

All

Aliases

None

Options

connection   Connection handle (1-n).

Keywords

None

Usage Notes

1. *conid* must reference an existing open connection, as shown in the output from **lscon**.

Examples

1. To switch from the current connection to connection 3: setcon 3

Related Commands

Section 19.68, "OPEN command"

Section 19.59, "LSSITES command"

Section 19.55, "LSCONNECTIONS command"

19.83. SETDRIVE command

Synopsis

setdr[**ive**] *connection\drive newdrive*

Description

This command changes the logical drive letter assigned to a connection.

HSI Version

All

Aliases
    None
Options

connection   Connection handle ID (1-n), as shown by the **lsconnections** command.

drive   Current drive letter (a-z, A-Z).

newdrive   Drive letter to assign (a-z, A-Z).

If value = -1, then an indefinite timeout is used.

Trailing colons may optionally be specified in the connection or drive letters.

Keywords
    None
Usage Notes

1. *drive:* must reference an existing open connection, and *new_drive:* must not currently be in-use by an open connection.
2. Drive letters are not case-sensitive.

Examples

1. To change drive letter "T" to "M": `setdrive T: m:`

Related Commands

Section 19.68, "OPEN command"

Section 19.59, "LSSITES command"

Section 19.82, "SETCONNECTION command"

19.84. SHARED_MEM command

Synopsis
    shared_mem
Description
    This command is the on/off toggle for enabling or disabling the use of shared memory for I/O buffers when HSI is performing file transfers.
HSI Version
    All
Aliases
    None
Options
    None
Keywords
    None
Usage Notes

1. This command is only effective on systems which support shared memory. In addition, the use of shared memory may be disabled by the site administrator or at build time when HSI is compiled.

Examples
> None
Related Commands
> None

19.85. SHOW command

Synopsis
> **show** [**-l**] | [**-h**] [*keyset …*]

Description
> This command displays your saved keysets or the names of all saved keysets.

HSI Version
> All

Aliases
> None

Options

> **-l**   (Letter "ell"): show local keysets.

> **-h**   Show HPSS-resident keysets (default).

> *keyset* Name of an existing saved keyset.

Keywords
> None

Usage Notes

> 1. This command is deprecated and will be removed in a future release.

Examples
> None

Related Commands

> **Section 19.1, "ADOPT command"**

> **Section 19.31, "FREE command"**

19.86. STAGE command

Synopsis
> **sta**[**ge**] [**-A**] [**-c**] [**-i**] [**-m** *max*] [**-N**] [**-R**] [**-s**] [**-w**] [*path …*] [**<<MARKER**]

Description
> This command stages files from HPSS tape to disk cache.

HSI Version
> All

Aliases
> None

Options

-A	Enable tape mount optimization (auto-scheduling) (default).
-c	Do not delay until all stages have completed (*see usage note 1*).
-i	Interactive notification when file(s) have been staged (*see usage note 1*).
-m	Max number of stages to issue in a single batch. Default=no limit.
-N	Disable optimization of stage requests by VV and position.
-R	Recursively traverse directories in the specified path(s).
-s	Display background staging statistics (*see usage note 1*).
-w	Wait for each stage to complete (foreground stage). Default is to stage in the background. Setting **-w** internally disables **-A** and enables **-N**.
MARKER	provides sh-style "here"-document syntax for specifying filenames.

- Lines following the MARKER, which must be the last token on the line, contain one or more pathnames (which may include wildcards).
- The list is terminated by a line containing MARKER as the first token on the line.

Keywords
None
Usage Notes

1. The **-c**, **-i**, and **-s** options are available in HSI for HPSS Version 7.5.1 and later, in HSI Version 6.0.0.p10 and later (HPSS Version 7.5.1) and HSI Version 6.2.0.p0 and later (HPSS Version 7.5.2).
2. If **-w** is specified, it overrides **-A**, and disables optimization based on tape and position. Instead, HSI issues a foreground stage for each file as it is encountered in the list, and waits for it to complete before continuing with the next file in the list.

3. The shell-style HERE-Document syntax can be used to specify an interactive list of paths, or can be used in an IN file: **Interactive mode:** `stage file1 file2 << EOF FILENAMES: file3 file4 FILENAMES: file5 FILENAMES: EOF`

   Using an IN File: `hsi in stageInput`

   Contents of file "stageInput": `stage file1 file2 << EOF file3 file4 file5 EOF`

Examples

```
stage –w my_file
```
Related Commands

> **Section 19.62, "MIGRATE command"**

> **Section 19.74, "PURGE command"**

19.87. SU command

Synopsis
> **su** [**-** | **-l** | **-m**] [*user* | *#uid*]

Description
> This command assumes another user's identity.

HSI Version
> All

Aliases
> None

Options

> **-**   Same as **-l**.

> **-l**   Full login with target user's environment (deferred implementation).

> **-m**   Login with current user's environment (default).

> *user* Login name of the user whose identity is to be assumed.

> *#uid* Numeric user id (the "#" prefix is required, with no intervening whitespace) of the user whose identity is to be assumed. Default is root (uid 0).

Keywords
> None

Usage Notes

> 1. The current login identity must be authorized by the HPSS administrator in order to use this command.

Examples
> ```
> su – jane
> ```

Related Commands
> **Section 19.88, "SUDO command"**

19.88. SUDO command

Synopsis

> **sudo** [**-K** | **-k** | **-l**] [**-v**]

or

sudo [**-s**] [**-u** *user* | *#uid*] [--] *command*

Description

This command issues an HSI command using another user's identity.

HSI Version

HSI Version 3.4.1 and later

Aliases

None

Options

    -K   A "sure kill" option: invalidate the current sudo session and remove the starting timestamp.

    -k   Invalidate the current sudo session, but do not remove the starting timestamp.

    -l   List allowed and forbidden commands for the current active sudo session (deferred implementation).

    -s   SU to the target user and then run the command. Stays in "SU" mode after the command completes.

    -u   Target user login name or numeric uid (# prefix is required, with no intervening whitespace).

    -v   Validate and extend the current SUDO session's timestamp.

    command HSI command to be executed.

Keywords

None

Usage Notes

1. The current login identity must be authorized by the HPSS administrator in order to use this command. For SUDO, the administrator may further restrict the list of commands that users are allowed to execute.
2. The command string is currently limited to a single HSI command; semicolon characters in the string are treated as part of the command rather than as command separators.

Examples

1. Issue a single command as user "joe": `sudo -u joe ls -lR`
2. Kill the current sudo session, but do not invalidate the timestamp for this session: `sudo -k`
3. SU to user 53781 and then issue a command: `sudo -s -u #53781 get localFile : projectFile`

Related Commands

Section 19.87, "SU command"

19.89. SUNIQUE command

Synopsis
> **sunique**

Description
> This command toggles creating HPSS files with unique names.

HSI Version
> All

Aliases
> None

Options
> None

Keywords
> None

Usage Notes

> 1. This command is a placeholder for possible future implementation. It currently does nothing.

Examples
> None

Related Commands
> None

19.90. SWITCH command

Synopsis
> {**swi**[**tch**] | **setc**[**on**]} *hpss_connection_handle*

Description
> This command switches an active HPSS server connection.

HSI Version
> All

Aliases
> setcon

Options

> *hpss_connection_handle*
>> The connection handle number can be obtained by issuing the **lscon** command from an HSI prompt. The connection handles would be "1" and "2" in the following example: ? lscon List of Currently Active Connections Current default connection handle: 1 Han- Remote Remote Drive dle IP Address Hostname -- ->A: 1 <IPv4 addr8> hpss07g.site8 HomeDir: /home/kericson pwd0: /home/kericson B: 2 <IPv4 addr9> hpss.site9 HomeDir: /home/kericson pwd0: /home/kericson

Keywords
> None

Usage Notes

> 1. This can also be accomplished by typing drive:, for example B:.

Examples
>   switch 2

Related Commands

> **[Section 19.68, "OPEN command"](#)**

19.91. TOUCH command

Synopsis
   tou[ch] [-a] [-c *cosID*] [-f] [-l *localfile*] [-M *mode*] [-m] [-R] [-r *HPSS_file*] [-t *timestamp*] [-x] *path* …
Description
   This command updates the last access time for existing files, creates a new (0-length) file if a file does not exist.
HSI Version
   All
Aliases
   None
Options

   -a   Change access time of file. Modification time is not changed unless the **-m** flag is also specified.

   -c   Specifies Class of Service ID to be used. The string "auto" can be specified to cause HSI's automatic Class of Service selection algorithm to be used. This is the normal default, but **-c auto** can be used to temporarily override any current COS ID that was previously set by a **set cosid** command.

   -f   Reserved - currently unused.

   -l   Localfile: use local file's timestamp instead of current time.

   -M   Specifies octal mode flags to use for new file creation.

   -m   Change modification time of file. Access time is not changed unless the **-a** flag is also specified.

   -R   [standard option] Recursively traverse directories in the specified path(s).

   -r   Use *HPSS_file*'s timestamp instead of the current time.

   -t   Use *timestamp* as [[CC]YY]MMDDhhmm[.ss] instead of the current time. CC : Optional century YY : Optional year MM : Month number (1-12) DD : Day number (1-n) hh : Hour (00-23) mm : Minute (00-59) ss : Optional second (0-59)

-x   Do not create the file if it does not exist.

Keywords
> DIRn

Usage Notes

> 1. The options **-a**, **-f**, **-l**, **-m**, **-r**, **-t**, and **-x** are available in HSI Version 3.4.4 and later (for HPSS 6.2) and in HSI Version 3.5.1 and later (HPSS 7).
> 2. The Class of Service IDs that are available at a site can be viewed by using the **lscos** command.

Examples

> 1. Update timestamps for all files that end in `.c` or `.h`: `touch dir=sources *.c *.h`
> 2. Create a new empty file called "newFile" using Class of Service ID 401, with permissions set to read-write for owner and group: `touch -M 0662 -c 401 newFile`
> 3. Synchronize an HPSS file's timestamp with that of its local file counterpart: `touch -l myfile.c myfile.c`
> 4. Set the modify time of a file to April 1, 1949: `touch -t 194904010000 mikes_bd_cake`

Related Commands
> None

19.92. UMASK command

Synopsis
> **umask** [*perm_mask*]

Description
> This command sets the file creation mask, which is used when new files are created.

HSI Version
> All

Aliases
> None

Options

> *perm_mask* An octal number between 0 and 777 used to specify permissions which should not be set on files and directories created after this command is complete.

Keywords
> None

Usage Notes

> 1. Setting of this mask to allow "group" and "other" permissions for all nodes created by HPSS is discouraged. Individual nodes can have their permissions altered, if required for file sharing, by using the **chmod** command.
> 2. Currently, permission-mask must be an octal value (000-777). UNIX-style "chmod-style" syntax for owner, group, and other access (for example, o+x) will be added in the future.

Examples

> 1. Allow rwxr-xr-x access to be set for files and directories that are created: `umask 022`

Related Commands

Section 19.11, "CHMOD command"

Section 19.12, "CHOWN command"

19.93. VERBOSE command

Synopsis
> **verbose**

Description
> This command is the toggle command for enabling and disabling verbose mode. Currently, this option is used to control the issuing of "marker" messages when recursively traversing directories, to aid in finding a good restart point when failures occur.

HSI Version
> All

Aliases
> None

Options
> None

Keywords
> None

Usage Notes
> None

Examples
> None

Related Commands
> None

19.94. VERSION command

Synopsis
> **version**

Description
> This command shows HSI version information.

HSI Version
> All

Aliases
> None

Options
> None

Keywords
> None

Usage Notes
> None

Examples
> None

Related Commands
> None

19.95. WHOAMI command

Synopsis
> **whoami**

Description

This command displays the current login name.

HSI Version

HSI Version 3.4.1 and later.

Aliases

None

Options

None

Keywords

None

Usage Notes

1. This command is equivalent to id -un.

Examples

? whoami

Related Commands

[**Section 19.87, "SU command"**](#)

[**Section 19.88, "SUDO command"**](#)

[**Section 19.42, "ID command"**](#)