Aspera Command-Line Interface Guide 3.7.5

Mac OS X

Revision: 3.7.5.147556 Generated: 08/31/2017 01:25

Contents

Introduction

IBM Aspera Command-Line Interface (the Aspera CLI) is a collection of Aspera tools for performing high-speed, secure data transfers from the command line. The Aspera CLI is for users and organizations who want to automate their transfer workflows.

The Aspera CLI is comprised of three command-line programs:

aspera	The **aspera** executable is a command-line client for performing transfers with Aspera Faspex, Aspera Files, and Aspera Shares transfer servers. For further information on the **aspera** program, see *aspera: The Command-Line Transfer Client*
ascp	The **ascp** executable is a command-line FASP transfer program. For information on the **ascp** program, see *ascp: Transferring from the Command Line*
ascp4	**ascp4**, or A4, is a FASP transfer program similar to **ascp** that has been optimized for sending large sets of individual files and can support UDP multicast through Aspera FASPStream. For information on A4, see *Transferring with ascp4*.

System Requirements

Mac OS X 10.7, 10.8, 10.9, 10.10, 10.11, or macOS 10.12.

Required Aspera Licenses

- The Aspera CLI requires a Connect-enabled license on the transfer server. For detailed information on your transfer server's license file, see the *Aspera Connect Server Admin Guide* .
- The Aspera CLI package includes a free client license. It or another valid **aspera-license** file must be present in the Aspera CLI installation directory.

Installation

Installing the Aspera CLI

1. Download the Aspera CLI package from the Aspera website.
2. Run the installation script:

   aspera-cli-x.x.x.xxx.xxxxxxx-mac-xx.x-64-release.sh

   The script places the Aspera CLI in the **$HOME/Library/Aspera** directory.

   Note: If you have a previous installation of the **aspera** command-line client, note that the default installation directory has changed.
3. [Optional] To install the Aspera CLI in your PATH, run the following command:

```
# export PATH=~/Applications/Aspera\ CLI/bin:$PATH
```

4. [Optional] To install the man pages, run the following command:

```
# export MANPATH=~/Applications/Aspera\ CLI/share/man:$MANPATH
```

5. [Optional] To set an environment variable with the value of your password, to be used with all **aspera** client commands, run the following command:

```
# export ASPERA_PASS=mypassword
```

Configuring for Faspex

If you plan to use the Aspera CLI to browse the contents of a remote directory through Faspex, you must configure the client after installing it. The settings that affect the Aspera CLI with **faspex browse** reside in the following file:

```
.aspera_cli_conf
```

In a text editor, edit the **.aspera_cli_conf** file to set the following:

- server name
- port
- username
- password
- base directory

Configuration File Syntax

The Aspera CLI package installs a **.aspera_cli_conf** file with sample configurations that you can use to see the correct syntax for this file.

Credentials in Your Configuration File

The username and password credentials that you set in the **.aspera_cli_conf** file should be the same as the credentials for the Node API user on your Faspex server (not host system credentials).

Defining Multiple Servers

You can define multiple servers in the **.aspera_cli_conf** file, and multiple sources for each server. Then at the command line, you can specify which one to use, with the **uid** or **name** value that you defined in the configuration file.

Configuring for Files

If you plan to use the Aspera CLI to transfer packages to a Files workspace, you must configure both.

Configuration Steps

The following steps assume that

- You have a working Files instance and at least one workspace.
- You have the relevant JWT private/public key pair.
- You have installed the Aspera CLI.

To configure Files and the Aspera CLI to work together, do the following:

1. In Files, set up JWT authentication for the Aspera CLI.

   Use the instructions in the Files Help Center at *https://testeng.qa.asperafiles.com/helpcenter/admin/organization/registering-an-api-client*. In this process, you are registering the Aspera CLI as a client that will be allowed to use the Files API without going through a web browser.

   Note: This registration must be performed by a Files admin. The admin must have the JWT public key. The output of the registration process is the client ID and secret, which are used in a later step.
2. In Files, add your JWT public key.

   Use the instructions in the Files Help Center at *https://aspera.asperafiles.com/helpcenter/using-files/basic/adding-your-public-key*.
3. In the Aspera CLI installation, add your private key.

   The file must be located in *installation_directory***cli****etc**.
4. In the Aspera CLI installation, edit the configuration file.

   a. Locate the **.aspera_cli_conf** configuration file:

   ```
   .aspera_cli_conf
   ```

   b. In a text editor, edit the Files section of the **.aspera_cli_conf** file to set the following:

      - The client ID (**client_ID**) and secret (**client_secret**) that you obtained in Step *1*.
      - The filename of the private key (**private_key_file_name**). The default value is **private.pem**.
      - The organization name (**name**) and hostname (**host**) of the Files server.

   c. Save the **.aspera_cli_conf** file.

Configuration File Syntax

The Aspera CLI package installs a **.aspera_cli_conf** file with sample configurations that you can use to see the correct syntax for this file.

Defining Multiple Servers

You can define multiple servers in the **.aspera_cli_conf** file, and multiple sources for each server. Then at the command line, you can specify which one to use, with the **name** value that you defined in the configuration file.

Uninstalling

You can uninstall the Aspera CLI by deleting the installation directory with the following command:

```
# rm -rf ~/Applications/Aspera \ CLI/
```

aspera: The Command-Line Transfer Client

About the Command-Line Client

The **aspera** program is a client application that allows you to interact with Aspera Faspex, Aspera Files, and Aspera Shares transfer servers from the command line. The client provides the same data-transfer functionality as Faspex and Shares, in convenient commands that allow you to automate operations.

For example, with the **aspera** client, you can automate the following:

- Listing the contents of your Faspex inbox, Files workspace, or Shares share.
- Uploading to and downloading from your Shares server.
- Sending Faspex packages using files from your local directories.
- Sending files from remote storage sources, such as clusters or S3.
- Downloading packages that are sent to you, to a local storage location.

Prerequisites

Certificates

All **aspera** client operations perform certificate validation.

The included **certs** directory (or your own certificate authority keys) must be located either in the parent directory of the **aspera** executable, or in a location that you specify through the **-b** command-line argument.

If a transfer server does not have a valid certificate to allow the operation, you must specify the **--insecure** option.

Required Aspera Software

The machine that runs the **aspera** client must have either an Aspera server or ascp (which is provided with this package) installed, in the same directory as the **aspera** executable -- in the **PATH** or in standard Aspera installation locations.

aspera Command Reference

Syntax

All command sequences begin with the program name, **aspera**. The **aspera** program uses the following command syntax:

```
# aspera command subcommand [arguments]
```

Commands

The **aspera** program offers the following commands:

faspex	use the Faspex application
files	use the Files application
shares	use the Shares application
help	view help information for a command

version	print the version number of this program

Faspex Subcommands

The **faspex** command offers the following subcommands:

browse	view the contents of a source directory
dropbox	show information about a dropbox
get	download package
list	show information about an inbox
send	send a package

For details on the Faspex subcommands, see *faspex Command Reference*

Files Subcommands

The **files** command offers the following subcommands:

send	send a package
help	view help information for a command
version	print the version number of this program

For details on the Faspex subcommands, see *faspex Command Reference*

Shares Subcommands

The **shares** command offers the following subcommands:

upload	upload files or directories to a Shares server
download	download files or directories from a Shares server
browse	browse a directory of a Shares server
delete	delete a file or directory
rename	rename a file or directory

For details on the Shares subcommands, see *shares Command Reference*

Getting Help

The **aspera** program also offers a **help** command that can explain any command in more detail, and provide you with sample use cases. To view the help, type the following:

```
# aspera help [faspex|files|shares]
```

To view the help for a particular subcommand, the syntax is as follows:

```
# aspera [faspex|files|shares] help subcommand
```

For example, to see the options for the Faspex **send** command, type the following:

```
# aspera faspex help send
```

Finding the Software Version Number

To see the version of the **aspera** client that is installed, type the following:

```
# aspera version
```

faspex Command Reference

Faspex Subcommands

The **faspex** command offers the following subcommands:

browse	view the contents of a source directory
dropbox	show information about a dropbox
get	download a package
list	show information about an inbox
send	send a package

For examples of Faspex subcommands in use, see *Faspex Examples*

The browse Subcommand

Use the **browse** subcommand to browse a remote source that is defined in the **.aspera_cli_conf** file. The syntax for **browse** is as follows:

```
# aspera faspex browse [args]
```

The arguments you give to the **browse** subcommand specify the remote source by source ID name, and the directory you want to browse. The output shows a list of the directories and files, in human-readable format. If you prefer to retrieve the output in JSON format for integration into your automation workflow, use the **-j** parameter.

-b *path* **--base-ca-path=***path*	The base path for your CA certificates. If your certificates are in the default location, this argument is not required. The default path is **~/.aspera/cli/certs**.
-c *num* **--count=***num*	List only up to *num* items.
-i **---insecure**	Accept the certificate, even if it's invalid.
-j **--json**	Output raw JSON.
-k *num* **--skip=***num*	Skip the first *num* items.
-o *order* **--sort=***order*	Sort by *order* (required). The options for *order* are as follows: • **type** = sort directories first, then files • **size_a** = sort by file size (ascending) • **size_d** = sort by file size (descending)

	• **mtime_a** = sort by file modification time (ascending) • **mtime_d** = sort by file modification time (descending)
-p *path* **--path=***path*	The path to the source you want to view. This path is relative to the path you specified in **.aspera_cli_conf**.
-s *id_or_name* **--source=***id_or_name*	The ID or name of the source server (a matching ID takes precedence over a matching name), as defined in the **.aspera_cli_conf** configuration file.
-v **--verbose**	Show more verbose output, for debugging.

The dropbox Subcommand

Use the **dropbox** subcommand to show information about dropboxes. The syntax for **dropbox** is as follows:

```
# aspera faspex dropbox [args]
```

Arguments for the **dropbox** subcommand:

-a **--list-all**	Show info for all dropboxes.
-b *path* **--base-ca-path=***path*	The base path for your CA certificates. If your certificates are in the default location, this argument is not required. The default path is **~/.aspera/cli/certs**.
-H *hostname* **--host=***hostname*	The hostname or IP address of the Faspex server.
-i **---insecure**	Accept the certificate, even if it's invalid.
-j **--json**	Output raw JSON.
-l *dropbox_id* **--list=***dropbox_id*	Show info for this dropbox only. The *dropbox_id* is defined in the Faspex application.
-p [*password*] **--password=**[*password*]	The Faspex user password. If you specify **-p** but omit the *password* value, the system assumes an empty string for this value. If you do not specify **-p**, the Aspera CLI prompts you for a non-echoing password. Alternatively, you can set the **ASPERA_PASS** environment variable. For instructions, see *Installing the Aspera CLI*.
-T *port_number* **--port=***port_number*	The listening port on the Faspex server.
-u *username* **--user=***username*	The Faspex username.

-U *url_prefix* **--url-prefix=***url_prefix*	A prefix to the Faspex URL. The default prefix string is **/aspera/faspex/**.
-v **--verbose**	Show more verbose output, for debugging.

The get Subcommand

Use the **get** subcommand to download a Faspex package. The syntax for **get** is as follows:

```
# aspera faspex get [args]
```

Arguments for the **get** subcommand:

-b *path* **--base-ca-path=***path*	[Optional] The base path for your CA certificates. If your certificates are in the default location, this argument is not required. The default path is **~/.aspera/cli/certs**.
--cipher=*cipher*	Attempt to set the encryption cipher (if server settings allow). *cipher* can have the following values: • **aes-128** • **aes-192** • **aes-256** • **none**
--content-protect-password=*password*	Specify the password that is used to encrypt/decrypt files on the server.
-E *pattern* **--exclude=***pattern*	Exclude files that match the given pattern. To specify multiple patterns, repeat the **-E** option.
-f *path* **---file=***path*	The file path to download to.
-H *hostname* **--host=***hostname*	The hostname or IP address of the Faspex server.
-i **---insecure**	Accept the certificate, even if it's invalid.
--min-rate=*new_rate*	Attempt to revise the minimum rate (if server settings allow) to a new throughput value, in kbps.
-o *overwrite_method* **--overwrite=***overwrite_method*	Overwrite existing files. *overwrite_method* can be any of the following values: • **never** • **always** • **older** • **diff** • **diff+older**
-p [*password*] **--password=**[*password*]	The Faspex user password. If you specify **-p** but omit the *password* value, the Aspera CLI assumes an empty string for this value.

	If you do not specify **-p**, the Aspera CLI prompts you for a non-echoing password. Alternatively, you can set the **ASPERA_PASS** environment variable. For instructions, see *Installing the Aspera CLI*.
--rate-policy=*policy*	Attempt to revise the rate policy (if server settings allow). The options for *policy* are • **fixed** • **high** • **fair** • **low**
-T *port_number* **--port=***port_number*	The listening port on the Faspex server.
--target-rate=*new_rate*	Attempt to revise the target rate (if server settings allow) to a new throughput value, in kbps.
-u *username* **--user=***username*	The Faspex username.
--url="*URL***"**	The FASP URL from which to download. To find the FASP URL for a package, use the **list** subcommand.
-U *url_prefix* **--url-prefix=***url_prefix*	A prefix to the Faspex URL. The default prefix string is **/aspera/faspex/**.
-v **--verbose**	Show more verbose output, for debugging.
-x *proxy_hostOrIp* **--proxy=***proxy_hostOrIp*	The hostname or IP address of the proxy computer (forward proxy).

The Faspex list Subcommand

Use the **list** subcommand to see the contents of a user's inbox. The syntax for **list** is as follows:

```
# aspera faspex list [args]
```

Arguments for the **list** subcommand:

-a **---archived**	List archived packages.
-b *path* **--base-ca-path=***path*	The base path for your CA certificates. If your certificates are in the default location, this argument is not required. The default path is **~/.aspera/cli/certs**.
-H *hostname* **--host=***hostname*	The hostname or IP address of the Faspex server.
-i **---insecure**	Accept the certificate, even if it's invalid.

-n **---inbox**	List the packages in the inbox.
-p [*password*] **--password=**[*password*]	The Faspex user password. If you specify **-p** but omit the *password* value, the Aspera CLI assumes an empty string for this value. If you do not specify **-p**, the Aspera CLI prompts you for a non-echoing password. Alternatively, you can set the **ASPERA_PASS** environment variable. For instructions, see *Installing the Aspera CLI*.
-s **---sent**	List sent packages.
-T *port_number* **--port=***port_number*	The listening port on the Faspex server.
-u *username* **--user=***username*	The Faspex username.
-U *url_prefix* **--url-prefix=***url_prefix*	A prefix to the Faspex URL. The default prefix string is **/aspera/faspex/**.
-v **--verbose**	Show more verbose output, for debugging.
-x **--xml**	Get raw XML RSS atom for this inbox. **Note:** As the format returned with this option is XML, if you want to download a package referenced in a link tag, make sure that you un-escape the returned XML value in the **href=** attribute.

The send Subcommand

Use the **send** subcommand to send a Faspex package. The syntax for **send** is as follows:

```
# aspera faspex send [args]
```

Arguments for the **send** subcommand:

-b *path* **--base-ca-path=***path*	The base path for your CA certificates. If your certificates are in the default location, this argument is not required. The default path is **~/.aspera/cli/certs**.
--cipher=*cipher*	Attempt to set the encryption cipher (if server settings allow). *cipher* can have the following values: • **aes-128** • **aes-192** • **aes-256** • **none**
--content-protect-password=*password*	Specify the password that is used to encrypt/decrypt files on the server.

-e **--remove-empty-directories**	When the transfer is complete, remove empty directories.
-E *pattern* **--exclude=***pattern*	Exclude files that match the given pattern. To specify multiple patterns, repeat the **-E** option.
-f *path* **---file=***path*	The file to send. You can specify this option multiple times, to indicate multiple files.
-H *hostname* **--host=***hostname*	The hostname or IP address of the Faspex server.
-i **---insecure**	Accept the certificate, even if it's invalid.
-m *metadata* **---metadata=***metadata*	Send metadata (JSON object text) with the package.
--min-rate=*newRate*	Attempt to revise the minimum rate (if server settings allow) to a new throughput value, in kbps.
-n "*body_text***"** **---note="***body_text***"**	A note for the body of the email message.
-p [*password*] **--password=**[*password*]	The Faspex user password. If you specify **-p** but omit the *password* value, the Aspera CLI assumes an empty string for this value. If you do not specify **-p**, the Aspera CLI prompts you for a non-echoing password. Alternatively, you can set the **ASPERA_PASS** environment variable. For instructions, see *Installing the Aspera CLI*.
-r *recipient* **--recipient=***recipient*	Recipient(s) of the package. You can specify this option multiple times, to indicate multiple recipients. The *recipient* can be a valid email address, a Faspex user account name, a Faspex dropbox name, or a workgroup.
-R **--remove-after-transfer**	When the transfer is complete, remove the transferred content from the source.
--rate-policy=*policy*	Attempt to revise the rate policy (if server settings allow). The options for *policy* are • **fixed** • **high** • **fair** • **low**
-s *ID* **--source-id=***ID*	Send a file from a source ID (as defined in the Faspex application).
-t *title_text*	A title (subject line) for the email message.

--title=*title_text*	
-T *port_number* **--port=***port_number*	The listening port on the Faspex server.
--target-rate=*new_rate*	Attempt to revise the target rate (if server settings allow) to a new throughput value, in kbps.
-u *username* **--user=***username*	The Faspex username.
-U *url_prefix* **--url-prefix=***url_prefix*	A prefix to the Faspex URL. The default prefix string is **/aspera/faspex/**.
-v **--verbose**	Show more verbose output, for debugging.
-x *proxy_hostOrIp* **--proxy=***proxy_hostOrIp*	The hostname or IP address of the proxy computer (forward proxy).

files Command Reference

Files Subcommands

The **files** command offers the following subcommands:

send	send a package to a Files workspace
help	view help information for a command
version	print the version number of this program

For examples of Files subcommands in use, see *Files Examples*

The send Subcommand

Use the **send** subcommand to send a package. The syntax for **send** is as follows:

```
# aspera files send [args]
```

Arguments for the **send** subcommand:

-f *path* **---file=***path*	The file or files to send in the package. You can specify this option multiple times, to indicate multiple files.
-lw **--lsworkspace**	Get a list of the Files workspaces.
-n *package_name* **---name=***package_name*	A name for the package.
-m "*body_text***"**	[Optional] Text for the body of the email message.

---message=*body_text*	
-o *organization_name* **--organization=***organization_name*	The name of the organization in Files that is the source of the package you are sending. Organizations are specified in the **.aspera_cli_conf** file.
-q *list_file* **--filelist** *list_file*	A file that contains a list of files to transfer. In the *list_file*, list the files that you want to transfer. *list_file* must be a plaintext file. Files must be sparated by newline characters. Paths to these files can be stated as relative to the current directory, or as absolute paths.
-r *recipient* **--recipient=***recipient*	Recipient(s) of the package. You can specify this option multiple times, to indicate multiple recipients. The *recipient* string should be a valid email address.
-u *username* **--user=***username*	The Files username (an email address).
-v **--verbose**	[Optional] Show more verbose output, for debugging.
-w *workspace_name* **--workspace=***workspace_name*	The Files workspace to send content to.

shares Command Reference

Shares Subcommands

The **shares** command offers the following subcommands:

browse	browse a directory of a Shares server
delete	delete a file, directory, or share
download	download files or directories from a Shares server
rename	rename a file or directory
upload	upload files or directories to a Shares server

For examples of Shares subcommands in use, see *Shares Examples*

The browse Subcommand

Use the **browse** subcommand to see what content is on a Shares server. The syntax for **browse** is as follows:

```
# aspera shares browse [args]
```

Arguments for the **browse** subcommand:

-b *path* **--base-ca-path=***path*	The base path for your CA certificates. If your certificates are in the default location, this argument is not required. The default path is **~/.aspera/cli/certs**.
-c *num* **--count=***num*	List only up to *num* items.

-H *host* **--host=***host*	The Shares host name.
-i **---insecure**	Accept the certificate, even if it's invalid.
-j **--json**	Output raw JSON.
-k *num* **--skip=***num*	Skip the first *num* items.
-o *order* **--sort=***order*	Sort by *order*. The options for *order* are as follows: • **type** = sort directories first, then files • **size_a** = sort by file size (ascending) • **size_d** = sort by file size (descending) • **mtime_a** = sort by file modification time (ascending) • **mtime_d** = sort by file modification time (descending)
-p [*password*] **--password=**[*password*]	The Shares user password. If you specify **-p** but omit the *password* value, the Aspera CLI assumes an empty string for this value. If you do not specify **-p**, the Aspera CLI prompts you for a non-echoing password. Alternatively, you can set the **ASPERA_PASS** environment variable. For instructions, see *Installing the Aspera CLI*.
-P *path* **--path=***path*	The Shares remote path (the default is /; or use the format */shareName/relativePathTo/fileOrFolder*).
-T *port_number* **--port=***port_number*	The listening port on the Shares server.
-u *username* **--user=***username*	The Shares username.
-v **--verbose**	Show more verbose output, for debugging.

The delete Subcommand

Use the **delete** subcommand to delete content from a Shares server. The syntax for **delete** is as follows:

```
# aspera shares delete [args]
```

Arguments for the **delete** subcommand:

-b *path* **--base-ca-path=***path*	The base path for your CA certificates. If your certificates are in the default location, this argument is not required. The default path is **~/.aspera/cli/certs**.
-H *host*	The Shares host name.

--host=*host*	
-i **---insecure**	Accept the certificate, even if it's invalid.
-j **--json**	Output raw JSON.
-p [*password*] **--password=**[*password*]	The Shares user password. If you specify **-p** but omit the *password* value, the Aspera CLI assumes an empty string for this value. If you do not specify **-p**, the Aspera CLI prompts you for a non-echoing password. Alternatively, you can set the **ASPERA_PASS** environment variable. For instructions, see *Installing the Aspera CLI*.
-P *path* **--path=***path*	The path to the remote file or directory to be deleted (with the format */shareName/relativePathTo/fileOrFolder*).
-T *port_number* **--port=***port_number*	The listening port on the Shares server.
-u *username* **--user=***username*	The Shares username.
-v **--verbose**	Show more verbose output, for debugging.

The download Subcommand

Use the **download** subcommand to download content from a Shares server. The syntax for **download** is as follows:

```
# aspera shares download [args]
```

Arguments for the **download** subcommand:

-b *path* **--base-ca-path=***path*	The base path for your CA certificates. If your certificates are in the default location, this argument is not required. The default path is **~/.aspera/cli/certs**.
-c *cookie_string* **--cookie=***cookie_string*	Cookie, if one is required.
--cipher=*cipher*	Attempt to set the encryption cipher (if server settings allow). *cipher* can be any of the following values: • **aes-128** • **aes-192** • **aes-256** • **none**
--content-protect-password=*password*	Specify the password that is used to encrypt/decrypt files on the server.

-d *path* **--destination=***path*	Destination directory path (the default is **./**).
-e **--remove-empty-directories**	When the transfer is complete, remove empty directories.
-E *pattern* **--exclude=***pattern*	Exclude files that match the given pattern. To specify multiple patterns, repeat the **-E** option.
-H *host* **--host=***host*	The Shares host name.
-i **---insecure**	Accept the certificate, even if it's invalid.
--min-rate=*new_rate*	Attempt to revise the minimum rate (if server settings allow) to a new throughput value, in kbps.
-o *overwrite_method* **--overwrite=***overwrite_method*	Overwrite existing files. *overwrite_method* can be any of the following values: • **never** • **always** • **older** • **diff** • **diff+older**
-p [*password*] **--password=**[*password*]	The Shares user password. If you specify **-p** but omit the *password* value, the Aspera CLI assumes an empty string for this value. If you do not specify **-p**, the Aspera CLI prompts you for a non-echoing password. Alternatively, you can set the **ASPERA_PASS** environment variable. For instructions, see *Installing the Aspera CLI*.
-R **--remove-after-transfer**	When the transfer is complete, remove the transferred content from the source.
--rate-policy=*policy*	Attempt to revise the rate policy (if server settings allow). The options for *policy* are • **fixed** • **high** • **fair** • **low**
-s *path* **--source=***path*	File path to the source of the content you are downloading (with the format */shareName/relativePathTo/fileOrFolder*).
-T *port_number* **--port=***port_number*	The listening port on the Shares server.

--target-rate=*newRate*	Attempt to revise the target rate (if server settings allow) to a new throughput value, in kbps.
-u *username* **--user=***username*	The Shares username.
-v **--verbose**	Show more verbose output, for debugging.
-x *proxy_hostOrIp* **--proxy=***proxy_hostOrIp*	The hostname or IP address of the proxy computer (forward proxy).

The rename Subcommand

Use the **rename** subcommand to rename content on a Shares server. The syntax for **rename** is as follows:

```
# aspera shares rename [args]
```

Arguments for the **rename** subcommand:

-b *path* **--base-ca-path=***path*	The base path for your CA certificates. If your certificates are in the default location, this argument is not required. The default path is **~/.aspera/cli/certs**.
-d *path* **--destination=***path*	The new name for the file or directory.
-H *host* **--host=***host*	The Shares host name.
-i **---insecure**	Accept the certificate, even if it's invalid.
-j **--json**	Output raw JSON.
-p [*password*] **--password=**[*password*]	The Shares user password. If you specify **-p** but omit the *password* value, the Aspera CLI assumes an empty string for this value. If you do not specify **-p**, the Aspera CLI prompts you for a non-echoing password. Alternatively, you can set the **ASPERA_PASS** environment variable. For instructions, see *Installing the Aspera CLI*.
-P *path* **--path=***path*	The remote path to the content you are renaming (with the format */share_name/relative_path_to/file_or_folder*).
-s *path* **--source=***path*	The remote file or directory you are renaming.
-T *port_number*	The listening port on the Shares server.

--port=*port_number*	
-u *username* **--user=***username*	The Shares username.
-v **--verbose**	Show more verbose output, for debugging.

The upload Subcommand

Use the **upload** subcommand to upload content to a Shares server. The syntax for **upload** is as follows:

```
# aspera shares upload [args]
```

Arguments for the **upload** subcommand:

-b *path* **--base-ca-path=***path*	The base path for your CA certificates. If your certificates are in the default location, this argument is not required. The default path is **~/.aspera/cli/certs**.
-c *cookie_string* **--cookie=***cookie_string*	Cookie, if one is required.
--cipher=*cipher*	Attempt to set the encryption cipher (if server settings allow). *cipher* can be any of the following values: • **aes-128** • **aes-192** • **aes-256** • **none**
--content-protect-password=*password*	Specify the password that is used to encrypt/decrypt files on the server.
-d *path* **--destination=***path*	Destination directory path (with the format /*share_name*/*relative_path_to*/*file_or_folder*).
-e **--remove-empty-directories**	When the transfer is complete, remove empty directories.
-E *pattern* **--exclude=***pattern*	Exclude files that match the given pattern. To specify multiple patterns, repeat the **-E** option.
-H *host* **--host=***host*	The Shares host name.
-i **---insecure**	Accept the certificate, even if it's invalid.
--min-rate=*new_rate*	Attempt to revise the minimum rate (if server settings allow) to a new throughput value, in kbps.
-o *overwrite_method*	Overwrite existing files. *overwrite_method* can be any of the following values:

--overwrite=*overwrite_method*	• **never** • **always** • **older** • **diff** • **diff+older**
-p [*password*] **--password=**[*password*]	The Shares user password. If you specify **-p** but omit the *password* value, the Aspera CLI assumes an empty string for this value. If you do not specify **-p**, the Aspera CLI prompts you for a non-echoing password. Alternatively, you can set the **ASPERA_PASS** environment variable. For instructions, see *Installing the Aspera CLI*.
-R **--remove-after-transfer**	When the transfer is complete, remove the transferred content from the source.
--rate-policy=*policy*	Attempt to revise the rate policy (if server settings allow). The options for *policy* are • **fixed** • **high** • **fair** • **low**
-s *path* **--source=***path*	File path to the source of the content you are uploading.
-T *port_number* **--port=***port_number*	The listening port on the Shares server.
--target-rate=*newRate*	Attempt to revise the target rate (if server settings allow) to a new throughput value, in kbps.
-u *username* **--user=***username*	The Shares username.
-v **--verbose**	Show more verbose output, for debugging.
-x *proxy_hostOrIp* **--proxy=***proxy_hostOrIp*	The hostname or IP address of the proxy computer (forward proxy).

Faspex Examples

List the contents of a remote source (in this example, 22). You can then use the results to send packages with contents from that remote source.

```
# aspera faspex browse --sort=type --source=22 -p/Datasheets
```

Send a Faspex package containing a file named **test_file** in the current directory to **recipient** at **host.com**. This command contacts the Faspex server at **https://host.com** and logs in as the user **myusername** with the password **mypassword**. When the recipient receives the email, the subject line "File 4 U" will identify this package.

```
# aspera faspex send -f test_file -n"This is a note for a Faspex package
 sent with the command-line client" -t"File 4 U" -r"recipient" -H"host.com"
 -umyusername -pmypassword
```

Send a Faspex package containing a file called **test_file** from a remote source. This command contacts the Faspex server at **https://host.com** and logs in as the user **myusername** with the password **mypassword**. When the recipient receives the email, the subject line "File 4 U" will identify this package.

```
# aspera faspex send -f test_file --source=22 -n"This is a note for a Faspex
 package sent with the command-line client" -t"File 4 U" -r"recipient" -
H"host.com" -umyusername -pmypassword
```

Download the specified package based on the **faspe:**// URL that the Faspex **list** commmand returns.

```
# aspera faspex get -umyusername -H"myhost.com" -pmypassword --
url="faspe://..."
```

List the packages in a user's inbox, in short format.

```
# aspera faspex list -umyusername -H"myhost.com" -pmypassword -n
```

List the packages in a user's inbox, in XML (RSS) format.

👉 **Note:** As the format returned with the **-x** option is XML, if you want to download a package referenced in a link tag, make sure that you un-escape the returned XML value in the **href=** attribute.

```
# aspera faspex list -umyusername -H"myhost.com" -pmypassword -n -x
```

Files Examples

List the Files workspaces.

```
# aspera files send -lw -u user_email@example.com -o organization_name
```

👉 **Note:** When you use the **-lw** option, the only other required arguments are **-u** and **-o**.

Send a package containing a file called **test_file**. When the recipient receives the notification email, **Package name** will appear in the subject line.

```
# aspera files send -f test_file --organization test_org -n "Package name" -
r recipient_email@example.com -u user_email@example.com -w workspace_name -m
 "This is the body of the email message."
```

Shares Examples

Uploading a File

Upload **local_file** to the destination directory using the user with a username of **username** and a password of **password** and host **123.45.67.89**.

```
# aspera shares upload -i --host=123.45.67.89 -uusername -ppassword --
source=./local_file --destination=/upload_share/incoming
```

Downloading a File

Download **Bytestream-Sender-Receiver.mov** to the local destination **local_dir** using the user with a username of **username** and a password of **password** and host **123.45.67.89**.

```
# aspera shares download -i --host=123.45.67.89 -uusername -ppassword
 --source=/download_share/outgoing/Bytestream-Sender-Receiver.mov --
destination=./local_dir
```

Browsing a Server

Browse the **test_share** on the server **123.45.67.89** using the user with a username of **username** and a password of **password** to authenticate.

```
# aspera shares browse -i --host=123.45.67.89 -uusername -ppassword --path=/
test_share
```

Renaming a File

Rename the file **oldName.mov** to **newName.mov** on the host **123.45.67.89** using the user with a username of **username** and a password of **password** to authenticate.

```
# aspera shares rename -i --host=123.45.67.89 -uusername -ppassword --path=/
test --source=/outgoing/oldName.mov --destination=/outgoing/newName.mov
```

Deleting a File

Delete the file **/test/file** on the host **123.45.67.89** using the user with a username of **username** and a password of **password** to authenticate.

```
# aspera shares delete -i --host=123.45.67.89 -uusername -ppassword --path=/
test/file
```

ascp: Transferring from the Command Line

Ascp Command Reference

The executable `ascp` is a command-line FASP transfer program that has the following syntax and command options, and that supports the following environment variables.

For examples of `ascp` commands, see the following topics:

- *Ascp General Examples*
- *Ascp File Manipulation Examples*
- *Ascp Transfers with Object Storage and HDFS*

Ascp Syntax

```
ascp options [[username@]src_host:]source1[ source2 ...]
 [[username@]dest_host:]dest_path
```

username

The username of the Aspera transfer user can be specified as part of the source or destination, whichever is the remote server. It can also be specified with the `--user` option. If you do not specify a username for the transfer, the local username is authenticated by default.

> **Note:** If you are authenticating on a Windows machine as a domain user, the transfer server strips the domain from the username. For example, `Administrator` is authenticated rather than `DOMAIN\Administrator`. For this reason, you must specify the domain explicitly.

src_host

The name or IP address of the machine where the files or directories to be transferred reside.

source

The source file or directory to be transferred. Multiple arguments are separated by space characters.

dest_host

The name or IP address of the machine where the source files or directories are to be transferred.

dest_path

The destination directory where the source files or directories are to be transferred. If the source is a single file, the destination can be a filename. However, if there are multiple source arguments, the destination must be a directory. To transfer to the transfer user's docroot, specify "`.`" as the destination.

Specifying Files, Directories, and Paths

- Avoid the following characters in file and directory names: `/ \ " : ' ? > < & * |`
- Specify paths with forward-slashes, regardless of the operating system.
- If directory or file arguments contain special characters, specify arguments with single-quotes (' ') to avoid interpretation by the shell.

URI paths: URI paths are supported, but with the following restrictions:

- If the source paths are URIs, they must all be in the same cloud storage account. No docroot (download), local docroot (upload), or source prefix can be specified.
- If a destination path is a URI, no docroot (upload) or local docroot (download) can be specified.

- The special schemes `stdio://` and `stdio-tar://` are supported on the client side only. They cannot be used for specifying an upload destination or download source.
- If required, specify the URI passphrase as part of the URI or set it as an environment variable (`ASPERA_SRC_PASS` or `ASPERA_DST_PASS`, depending on the transfer direction).

UNC paths: If the server is Windows and the path on the server is a UNC path (a path that points to a shared directory or file on Windows), it can be specified in an `ascp` command using one of the following conventions:

- As an UNC path that uses backslashes (\ ): If the client side is a Windows machine, the UNC path can be used with no alteration. For example, `\\192.168.0.10\temp`. If the client is not a Windows computer, every backslash in the UNC path must be replaced with two backslashes. For example, `\\\\192.168.0.10\\temp`.
- As an UNC path that uses forward slashes (/): Replace each backslash in the UNC path with a forward slash. For example, if the UNC path is `\\192.168.0.10\temp`, change it to `//192.168.0.10/temp`. This format can be used with any client-side operating system.

Testing paths: To test `ascp` transfers, you can use a `faux://` argument in place of the source or target path to send random data without writing it to disk at the destination. For more information, see *IBM Aspera Enterprise Server Admin Guide: Testing and Optimizing Transfer Performance*. For examples, see *Ascp General Examples*.

Environment Variables

The following environment variables can be used with the `ascp` command:

`ASPERA_DST_PASS=`*`password`*

Set the password to authenticate a URI destination.

`ASPERA_PROXY_PASS=`*`proxy_server_password`*

Set the password for an Aspera Proxy server.

`ASPERA_SCP_COOKIE=`*`cookie`*

Set a cookie that you want associated with transfers.

`ASPERA_SCP_DOCROOT=`*`docroot`*

Set the transfer user docroot. Equivalent to using `--apply-local-docroot` when a docroot is set in `aspera.conf`.

`ASPERA_SCP_FILEPASS=`*`password`*

Set the passphrase to be used to encrypt or decrypt files. For use with `--file-crypt`.

`ASPERA_SCP_KEY="-----BEGIN RSA PRIVATE KEY..."`

Set the transfer user private key. Use instead of the `-i` option.

`ASPERA_SCP_PASS=`*`password`*

Set the password for the transfer user.

`ASPERA_SCP_TOKEN=`*`token`*

Set the transfer user authorization token. Overridden by `-W`.

`ASPERA_SRC_PASS=`*`password`*

Set the password to authenticate to a URI source.

Ascp Options

`-6`

Enable IPv6 address support. When specifying an IPv6 numeric host for *src_host* or *dest_host*, write it in brackets. For example, `username@[2001:0:4137:9e50:201b:63d3:ba92:da]:/path` or `--host=[fe80::21b:21ff:fe1c:5072%eth1]`.

`-@` ***range_start*:*range_end***

Transfer only part of a file: *range_start* is the first byte to send, and *range_end* is the last. If either position is unspecified, the file's first and last bytes (respectively) are assumed. This option only works for downloads of a single file and does not support transfer resume.

`-A, --version`

Display version and license information.

`--apply-local-docroot`

Apply the local docroot set in `aspera.conf` for this transfer user. Use to avoid specifying object storage access credentials in the command line. This option is equivalent to setting the environment variable `ASPERA_SCP_DOCROOT`.

`-C` *nodeid*:*nodecount*

Enable multi-session transfers (also known as parallel transfers) on a multi-node/multi-core system. A node ID (*nodeid*) and count (*nodecount*) are required for each session. *nodeid* and *nodecount* can be 1-128, but *nodeid* must be less than or equal to *nodecount*, such as 1:2, 2:2. Each session must use a different UDP port specified with the `-O` option. Large files can be split across sessions, see `--multi-session-threshold`. For more information, see the *Enterprise Server Admin Guide: Configuring Multi-Session Transfers*.

`-c {aes128|aes192|aes256|none}`

Encrypt in-transit file data using the specified cipher. This option overrides the `<encryption_cipher>` setting in `aspera.conf`.

`--check-sshfp=`*fingerprint*

Compare *fingerprint* to the server SSH host key fingerprint that is set with `<ssh_host_key_fingerprint>` in `aspera.conf`. Aspera fingerprint convention is to use a hex string without the colons; for example, f74e5de9ed0d62feaf0616ed1e851133c42a0082. For more information on SSH host key fingerprints, see the *Enterprise Server Admin Guide: Securing your SSH Server*.

Note: If HTTP fallback is enabled and the transfer "falls back" to HTTP, this option enforces server SSL certificate validation (HTTPS). Validation fails if the server has a self-signed certificate; a properly signed certificate is required.

`-D | -DD | -DDD`

Log at the specified debug level. With each `D`, an additional level of debugging information is written to the log.

`-d`

Create the destination directory if it doesn't already exist. This option is applied automatically to uploads to object storage.

`--delete-before-transfer`

Before transfer, delete any files that exist at the destination but not also at the source. Do not use with multiple sources, keepalive, URI storage, or HTTP fallback. The `asdelete` tool provides the same capability.

`--dest64`

Indicate that the destination path or URI is base64 encoded.

`-E` *pattern*

Exclude files or directories from the transfer based on the specified pattern. Use the `-N` option (include) to specify exceptions to `-E` rules. Up to 16 `-E` and `-N` rules can be specified. Rules are applied in the order in which they are encountered, from left to right. The following symbols can be used in the pattern:

- **`*`** (asterisk) represents zero or more characters in a string, for example `*.tmp` matches `.tmp` and `abcde.tmp`.
- **`?`** (question mark) represents a single character, for example `t?p` matches `tmp` but not `temp`.

For details and examples, see *Applying Filters to Include and Exclude Files*.

Note: When filtering rules are found in `aspera.conf`, they are applied *before* rules given on the command line (`-E` and `-N`).

`-e` *prepost_script*

Run the specified pre-post script as an alternate to the default `aspera-prepost` script. Specify the full path to the pre-post script. The purpose of the pre-script is to run custom commands such as shellscripts, perl scripts, Windows batch files, and executable binaries. The custom commands can make use of transfer statistics and other information placed in environment variables. For details on the setup and usage of prepost scripts, see the Enterprise Server Admin guide.

`--exclude-newer-than=`*mtime*, `--exclude-older-than=`*mtime*

Exclude files (but not directories) from the transfer, based on when the file was last modified. Positive *mtime* values are used to express time, in seconds, since the original system time (usually 1970-01-01 00:00:00). Negative *mtime* values (prefixed with "**-**") are used to express the number of seconds prior to the current time.

`-f` *config_file*

Read Aspera configuration settings from *config_file* rather than `aspera.conf`(the default).

`--file-checksum=`*hash*

Enable checksum reporting for transferred files, where *hash* is the type of checksum to calculate: `sha1`, `md5`, `sha-512`, `sha-384`, `sha-256`, or `none` (the default). For more information about checksum reporting, see *Reporting Checksums* .

Note: If the default value is `none`, the checksum is the type configured on the server, if any.

`--file-crypt={encrypt|decrypt}`

Encrypt files (when sending) or decrypt files (when receiving) for client-side encryption-at-rest (EAR). Encrypted files have the file extension `.aspera-env`. This option requires the encryption/decryption passphrase to be set with the environment variable `ASPERA_SCP_FILEPASS`. If a client-side encrypted file is downloaded with an incorrect password, the download is successful, but the file remains encrypted and still has the file extension `.aspera-env`.

`--file-list=`*file*

Transfer all source files and directories listed in *file*. Each source item is specified on a separate line. UTF-8 file format is supported. Only the files and directories are transferred. Path information is not preserved at the destination. To read a file list from standard input, use "-" in place of *file*.

For example, if `list.txt` contains the following list of sources:

```
/tmp/code/compute.php
doc_dir
images/iris.png
images/rose.png
```

and the following command is run:

```
# ascp --file-list=list.txt --mode=send --user=username --
host=ip_addr .
```

then the destination, in this case the the transfer user's docroot, will contain the following:

```
compute.php
doc_dir (and its contents)
iris.png
rose.png
```

Restrictions:

- The command line cannot use the *`user@host:source`* syntax. Instead, specify this information with the options `--mode`, `--host`, and `--user`.
- Paths specified in the file list cannot use the *`user@host:source`* syntax.
- Because multiple sources are being transferred, the destination must be a directory.
- Only one `--file-list` or `--file-pair-list` option is allowed per `ascp` session. If multiple lists are specified, only the last one is used.
- Only files and directories specified in the file list are transferred; any sources specified on the command line are ignored.
- If the source paths are URIs, the size of the file list cannot exceed 24 KB.

To create a file list that also specifies destination paths, use `--file-pair-list`.

`--file-manifest={none|text}`

Generate a list of all transferred files when set to `text`. Requires `--file-manifest-path` to specify the location of the list. (Default: `none`)

`--file-manifest-path=`*directory*

Save the file manifest to the specified location when using `--file-manifest=text`. File manifests must be stored locally. For cloud or other non-local storage, specify a *local* manifest path.

`--file-manifest-inprogress-suffix=`*suffix*

Apply the specified suffix to the file manifest's temporary file. For use with `--file-manifest=text`. (Default suffix: `.aspera-inprogress`)

`--file-pair-list=`*file*

Transfer files and directories listed in *file* to their corresponding destinations. Each source is specified on a separate line, with its destination on the line following it.

Specify destinations relative to the transfer user's docroot. Even if a destination is specified as an absolute path, the resulting path at the destination will still be relative to the docroot. Destination paths specified in the list are created automatically if they do not already exist.

For example, if the file `pairlist.txt` contains the following list of sources and destinations:

```
Dir1
Dir2
my_images/iris.png
project_images/iris.png
/tmp/code/compute.php
/tmp/code/compute.php
/tmp/tests/testfile
testfile2
```

and the following command is run:

```
# ascp --file-pair-list=pairlist.txt --mode=send --user=username
 --host=ip_addr .
```

then the destination, in this case the transfer user's docroot, now contains the following:

```
Dir2  (and its contents)
project_images/iris.png
tmp/code/compute.php
testfile2
```

Restrictions:

- The command line cannot use the *`user@host:source`* syntax. Instead, specify this information with the options `--mode`, `--host`, and `--user`.
- The *`user@host:source`* syntax cannot be used with paths specified in the file list.

- Because multiple sources are being transferred, the destination specified on the command line must be a directory.
- Only one `--file-pair-list` or `--file-list` option is allowed per `ascp` session. If multiple lists are specified, only the last one is used.
- Only files from the file pair list are transferred; any additional source files specified on the command line are ignored.
- If the source paths are URIs, the file list cannot exceed 24 KB.

For additional examples, see *Ascp General Examples*.

`-G` *write_size*

If the transfer destination is a server, use the specified write-block size, which is the maximum number of bytes that the receiver can write to disk at a time. Default: 256 KB, Range: up to 500 MB. This option accepts suffixes "M" or "m" for *mega* and "K" or "k" for *kilo*, such that a *write_size* of `1M` is one MB.

This is a performance-tuning option that overrides the `write_block_size` set in the client's `aspera.conf`. However, the `-G` setting is overriden by the `write_block_size` set in the server's `aspera.conf`. The receiving server never uses the `write_block_size` set in the client's `aspera.conf`.

`-g` *read_size*

If the transfer source is a server, use the specified read-block size, which is the maximum number of bytes that the sender reads from the source disk at a time. Default: 256 KB, Range: up to 500 MB. This option accepts suffixes "M" or "m" for *mega* and "K" or "k" for *kilo*, such that a *read_size* of `1M` is one MB.

This is a performance-tuning option that overrides the `read_block_size` set in the client's `aspera.conf`. However, the `-g` setting is overriden by the `read_block_size` set in the server's `aspera.conf`. When set to the default value, the read size is the default internal buffer size of the server, which might vary by operating system. The sending server never uses the `read_block_size` set in the client's `aspera.conf`.

`-h, --help`

Display the help text.

`--host=`*hostname*

Transfer to the specified host name or address. Requires `--mode`. This option can be used instead of specifying the host with the *hostname*:*file* syntax.

`-i` *private_key_file*

Authenticate the transfer using public key authentication with the specified SSH private key file. The argument can be just the file name if the private key is located in *`user_home_dir/`*`.ssh/`, because `ascp` automatically searches for key files there. Multiple private key files can be specified by repeating the `-i` option. The keys are tried in order and the process ends when a key passes authentication or when all keys have been tried without success, at which point authentication fails.

`-K` *probe_rate*

Measure bottleneck bandwidth at the specified probing rate (Kbps). (Default: 100Kbps)

`-k {0|1|2|3}`

Enable the resuming of partially transferred files at the specified resume level. (Default: 0)

Specify this option for the first transfer or it will not work for subsequent transfers. Resume levels:

`-k 0` – Always retransfer the entire file.

`-k 1` – Compare file attributes and resume if they match, and retransfer if they do not.

`-k 2` – Compare file attributes and the sparse file checksums; resume if they match, and retransfer if they do not.

`-k 3` – Compare file attributes and the full file checksums; resume if they match, and retransfer if they do not.

If a complete file exists at the destination (no `.aspx`), the source and destination file sizes are compared. If a partial file and a valid `.aspx` file exist at the destination, the source file size and the file size recorded in the `.aspx` file are compared.

`-L` *local_log_dir*[:*size*]

Log to the specified directory on the client machine rather than the default directory. Optionally, set the size of the log file (Default: 10 MB). See also `-R` for setting the log directory on the server.

`-l` *max_rate*

Transfer at rates up to the specified target rate. (Default: 10000 Kbps) This option accepts suffixes "G" or "g" for *giga*, "M" or "m" for *mega*, "K" or "k" for *kilo*, and "P", "p", or "%" for percentage. Decimals are allowed. If this option is not set by the client, the setting in the server's `aspera.conf` is used. If a rate cap is set in the local or server `aspera.conf`, the rate does not exceed the cap.

`-m` *min_rate*

Attempt to transfer no slower than the specified minimum transfer rate. (Default: 0) If this option is not set by the client, then the server's `aspera.conf` setting is used. If a rate cap is set in the local or server `aspera.conf`, then the rate does not exceed the cap.

`--mode={send|recv}`

Transfer in the specified direction: `send` or `recv` (receive). Requires `--host`.

`--move-after-transfer=`*archivedir*

Move source files and copy source directories to *archivedir* after they are successfully transferred. Because directories are copied, the original source tree remains in place. The transfer user must have write permissions to the *archivedir*. The *archivedir* is created if it does not already exist. If the archive directory cannot be created, the transfer proceeds and the source files remain in their original location.

To preserve portions of the file path above the transferred file or directory, use this option with `--src-base`. For an example, see *Ascp File Manipulation Examples*.

To remove empty source directories (except those specified as the source to transfer), use this option with `--remove-empty-directories` .

Restrictions:

- *archivedir* must be on the same file system as the source. If the specified archive is on a separate file system, it is created (if it does not exist), but an error is generated and files are not moved to it. For cloud storage, *archivedir* must be in the same cloud storage account.
- If the source is on a remote system (`ascp` is run in receive mode), *archivedir* is subject to the same docroot restrictions as the remote user.
- `--remove-after-transfer` and `--move-after-transfer` are mutually exclusive. Using both in the same session generates an error.
- Empty directories are not saved to *archivedir*.
- When used with `--remove-empty-directories` and `--src-base`, scanning for empty directories starts at the specified source base and proceeds down any subdirectories. If no source base is specified and a file path (as opposed to a directory path) is specified, then only the immediate parent directory is removed (if empty) after the source files have been moved.

`--multi-session-threshold=`*threshold*

Split files across multiple `ascp` sessions if their size is greater than or equal to *threshold*. Use with `-C`, which enables multi-session transfers.

Files whose sizes are less than *threshold* are not split. If *threshold* is set to 0 (the default), no files are split.

If `--multi-session-threshold` is not used, the threshold value is taken from the setting for `<multi_session_threshold_default>` in the `aspera.conf` file on the client. If not found in `aspera.conf` on the client, the setting is taken from `aspera.conf` on the server. The command-line setting overrides any `aspera.conf` settings, including when the command-line setting is 0 (zero).

Multi-session uploads to cloud storage are supported for S3 only and require additional configuration. For more information, see the *Enterprise Server Admin Guide: Configuring Multi-Session Transfers*.

`-N` *pattern*

Protect ("include") files or directories from exclusion by any `-E` (exclude) options that follow it. Files and directories are specified using *pattern*. Each option-plus-pattern is a *rule*. Up to 16 rules can be specified. Rules are applied in the order (left to right) in which they're encountered. Thus, `-N` rules protect files only from `-E` rules that follow them. Create patterns using standard globbing wildcards and special characters such as the following:

- **`*`** (asterisk) represents zero or more characters in a string, for example `*.tmp` matches `.tmp` and `abcde.tmp`.
- **`?`** (question mark) represents any single character, for example `t?p` matches `tmp` but not `temp`.

For details on specifying patterns and rules, including examples, see *Applying Filters to Include and Exclude Files*.

> **Note:** Filtering rules can also be specified in `aspera.conf`. Rules found in `aspera.conf` are applied *before* any `-E` and `-N` rules specified on the command line.

`-O` *fasp_port*

Use the specified UDP port for FASP transfers. (Default: 33001)

`--overwrite={never|always|diff|diff+older|older}`

Overwrite destination files with source files of the same name. Default: `diff`. This option takes the following values:

`never`

Never overwrite the file. However, if the parent folder is not empty, its access, modify, and change times may still be updated.

`always`

Always overwrite the file.

`diff`

Overwrite the file if different from the source. If a complete file at the destination is the same as a file on the source, it is not overwritten. Partial files are overwritten or resumed depending on the resume policy.

`diff+older`

Overwrite the file if older and also different than the source. For example, if the destination file is the same as the source, but with a different timestamp, it will not be overwritten. Plus, if the destination file is different than the source, but newer, it will not be overwritten.

`older`

Overwrite the file if its timestamp is older than the source timestamp.

Interaction with resume policy (-k): If the overwrite method is `diff` or `diff+older`, difference is determined by the resume policy (`-k {0|1|2|3}`). If `-k 0` or no `-k` is specified, the source and destination files are always considered different and the destination file is always overwritten. If `-k 1`, the source and destination files are compared based on file attributes (currently file size). If `-k 2`, the source and destination files are compared based on sparse checksums. If `-k 3`, the source and destination files are compared based on full checksums.

-P ***ssh-port***

Use the specified TCP port to initiate the FASP session. (Default: 22)

-p

Preserve file timestamps for access and modification time. Equivalent to setting `--preserve-modification-time`, `--preserve-access-time`, and `--preserve-creation-time`. Timestamp support in object storage varies by provider; consult your object storage documentation to determine which settings are supported.

On Windows, modification time may be affected when the system automatically adjusts for Daylight Savings Time (DST). For details, see the Microsoft KB article, *http://support.microsoft.com/kb/129574*.

On Isilon IQ OneFS systems, access time (`atime`) is disabled by default. In this case, `atime` is the same as `mtime`. To enable the preservation of `atime`, run the following command:

```
# sysctl efs.bam.atime_enabled=1
```

--partial-file-suffix=*suffix*

Enable the use of partial files for files that are in transit, and set the suffix to add to names of partial files. (The suffix does not include a " `.` ", as for a file extension, unless explicitly specified as part of the suffix.) This option only takes effect when set on the receiver side. When the transfer is complete, the suffix is removed. (Default: suffix is null; use of partial files is disabled.)

--policy={fixed|high|fair|low}

Set the FASP transfer policy.

fixed

Attempt to transfer at the specified target rate, regardless of network capacity. Content is transferred at a constant rate and the transfer finishes in a guaranteed time. The `fixed` policy can consume most of the network's bandwidth and is not recommended for most types of file transfers. It requires setting a maximum (target) rate (`-l` option).

high

Adjust the transfer rate to fully utilize the available bandwidth up to the maximum rate. When congestion occurs, the transfer rate is twice as fast as a fair-policy transfer. The `high` policy requires the setting of maximum (target) and minimum transfer rates (`-l` and `-m`).

fair

Adjust the transfer rate to fully utilize the available bandwidth up to the maximum rate. When congestion occurs, bandwidth is shared fairly by transferring at an even rate. The `fair` policy requires the setting of maximum (target) and minimum transfer rates (`-l` and `-m`).

low

Adjust the transfer rate to use the available bandwidth up to the maximum rate. Similar to fair mode, but less aggressive when sharing bandwidth with other network traffic. When congestion occurs, the transfer rate is reduced to the minimum rate until other traffic decreases.

If `--policy` is not set, `ascp` uses the server-side policy setting (`fair` by default).

--precalculate-job-size

Calculate the total size before starting the transfer. The server-side `pre_calculate_job_size` setting in `aspera.conf` overrides this option.

--preserve-access-time

Preserve the source-file access timestamps at the destination. Because source access times are updated by the transfer operation, the timestamp preserved is the one just *prior* to the transfer. (To prevent access times at the source from being updated by the transfer operation, use the `--preserve-source-access-time` option.)

On Isilon IQ OneFS systems, access time (`atime`) is disabled by default. In this case, `atime` is the same as `mtime`. To enable the preservation of `atime`, run the following command:

```
# sysctl efs.bam.atime_enabled=1
```

`--preserve-acls=`*mode*, `--remote-preserve-acls=`*mode*
`--preserve-xattrs=`*mode*, `--remote-preserve-xattrs=`*mode*

Preserve a file's access control lists (ACLs) and/or extended attributes (xattrs) when transferring between different file system types. The storage *mode* can be one of the following:

`native`

Preserve attributes using the native capabilities of the file system. However, `native` mode is not supported on all file systems; `--preserve-acls=native` and `--remote-preserve-acls=native`work only on Windows computers, and `--preserve-xattrs=native` and `--remote-preserve-xattrs=native` work only on Linux computers.

`metafile`

Preserve attributes in a separate file, named *`filename`*`.aspera-meta`. For example, attributes for `readme.txt` are preserved in a second file named `readme.txt.aspera-meta`. The metafiles are platform independent and can be copied between hosts without loss of information. The `metafile` mode is supported on all file systems.

`none`

Do not preserve attributes (default).

If the client and server have different values for *mode*, `metafile` is used silently. Metafiles are overwritten by subsequent transfers if `--overwrite` is set to any value other than `never`.

The `remote-` options specify the storage mode to use on the remote file system. If this option is not specified, the mode will be whatever is specified for the local file system. A `remote-` option with mode set to `native` may be overridden by the remote `ascp` if `native` mode is unsupported on the remote file system.

The amount of attribute data per file that can be transferred successfully is subject to `ascp`'s internal PDPU size limitation.

Note that older versions of `ascp` do not support values other than `none`, and transfers using `native` or `metafile` fail with an error that reports incompatible FASP protocol versions.

`--preserve-creation-time`

(Windows only) Preserve source-file creation timestamps at the destination. Only Windows systems retain information about creation time. If the destination is not a Windows machine, this option is ignored.

`--preserve-file-owner-gid`, `--preserve-file-owner-uid`

(Linux, UNIX, and macOS only) Preserve the group information (`gid`) or owner information (`uid`) of the transferred files. These options require the transfer user to be authenticated as a superuser.

`--preserve-modification-time`

Set the modification time, the last time a file or directory was modified (written), of a transferred file to the modification of the source file or directory. Preserve source-file modification timestamps at the destination.

On Windows, modification time may be affected when the system automatically adjusts for Daylight Savings Time (DST). For details, see the Microsoft KB article, *http://support.microsoft.com/kb/129574*.

`--preserve-source-access-time`

Preserve the access times of the original sources to the last access times prior to transfer. This prevents access times at the source from being updated by the transfer operation. Typically used in conjunction with the `--preserve-access-time` option.

`--preserve-xattrs={native|metafile|none}`

Preserve a file's extended attributes (xattrs) when transferring between different file system types. *mode* can be `native`, `metafile`, or `none` (default). See `--preserve-acls` for a full description of *mode* and the behavior of this option.

`--proxy=`*`proxy_url`*

Use the proxy server at the specified address. *proxy_url* should be specified with the following syntax:

`dnat[s]://`*`proxy_username`*`:`*`proxy_password`*`@`*`server_ip_address`*`:`*`port`*

The default ports for DNAT and DNATS protocols are 9091 and 9092. For a usage example, see *Ascp General Examples*.

`-q`

Run `ascp` in quiet mode (disables the progress display).

`-R` ***remote_log_dir***

Log to the specified directory on the server rather than the default directory. **Note:** Client users restricted to aspshell are not allowed to use this option. To specify the location of the local log, use `-L`.

`--remote-preserve-acls={native|metafile|none}`

Preserve a file's access control lists (ACLs) when transferring between different file system types. *mode* can be `native`, `metafile`, or `none` (default). See `--preserve-acls` for a full description of *mode* and the behavior of this option.

`--remote-preserve-xattrs={native|metafile|none}`

Preserve a file's extended attributes (xattrs) when transferring between different file system types. *mode* can be `native`, `metafile`, or `none` (default). See `--preserve-acls` for a full description of *mode* and the behavior of this option.

`--remove-after-transfer`

Remove all source files, but not the source directories, once the transfer has completed sucessfully. Requires write permissions on the source.

`--remove-empty-directories`

Remove empty source directories once the transfer has completed sucessfully, but do not remove a directory specified as the source argument. To also remove the specified source directory, use `--remove-empty-source-directory`. Directories can be emptied using `--move-after-transfer` or `--remove-after-transfer`. Scanning for empty directories starts at the srcbase and proceeds down any subdirectories. If no source base is specified and a file path (as opposed to a directory path) is specified, then only the immediate parent directory is scanned and removed if it's empty following the move of the source file. **Note:** Do not use this option if multiple processes (`ascp` or other) might access the source directory at the same time.

`--remove-empty-source-directory`

Remove directories specified as the source arguments. For use with `--remove-empty-directories`.

`-S` ***remote_ascp***

Use the specified remote `ascp` binary, if different than `ascp`.

`--save-before-overwrite`

Save a copy of a file before it is overwritten by the transfer. A copy of `filename.ext` is saved as `filename.`*`yyyy.mm.dd.hh.mm.ss.index`*`.ext` in the same directory. *index* is set to 1 at the start of each second and incremented for each additional file saved during that second. The saved copies retain the attributes of the original.

`--skip-special-files`

Skip special files, such as devices and pipes, without reporting errors for them.

`--source-prefix=`*prefix*

Prepend *prefix* to each source path. The prefix can be a conventional path or a URI; however, URI paths can be used only if no docroot is defined.

`--source-prefix64=`*prefix*

Prepend the base64-encoded *prefix* to each source path. If `--source-prefix=`*prefix* is also used, the last option takes precedence.

`--src-base=`*prefix*

Strip the specified path prefix from the source path of each transferred file or directory. The remaining portion of the path remains intact at the destination.

Without `--src-base`, source files and directories are transferred without their source path. (However, directories do include their contents.)

Example: To transfer the folders and files in the `/clips/out` folder, but not the `out` folder itself, run the following command:

```
# ascp -d --src-base=/clips/out/ /clips/out/ root@10.0.0.1:/in
```

Result: At the destination, the source folders and files appear in the `in` directory:

Source	Destination (docroot)	Destination without --src-base
`/clips/out/file1`	`/in/file1`	`/in/out/file1`
`/clips/out/folderA/file2`	`/in/folderA/file2`	`/in/out/folderA/file2`
`/clips/out/folderB/file3`	`/in/folderB/file3`	`/in/out/folderB/file3`

Note: Sources located outside the source base are not transferred. No errors or warnings are issued, but the skipped files are logged. For example, if `/clips/file4` were included in the above example sources, it would not be transferred because it is located outside the specified source base, `/clips/out/`.

Use with URIs: The `--src-base` option performs a character-to-character match with the source path. For object storage source paths, the prefix must specify the URI in the same manner as the source paths. For example, if a source path includes an embedded passphrase, the prefix must also include the embedded passphrase otherwise it will not match.

For additional examples, see *Ascp File Manipulation Examples*.

`--symbolic-links={follow|copy|copy+force|skip}`

Handle symbolic links using the specified method. On Windows, the only method is `skip`. On other operating systems, any of the following methods can be used:

`follow`

Follow symbolic links and transfer the linked files. (Default)

`copy`

Copy only the alias file. If a file with the same name is found at the destination, the symbolic link is not copied.

`copy+force`

Copy only the alias file. If a file (not a directory) with the same name is found at the destination, the alias replaces the file. If the destination is a symbolic link to a directory, it's not replaced.

`skip`

Skip symbolic links. Do not copy the link or the file it points to.

`-T`

Disable in-transit encryption for maximum throughput.

`-u` ***user_string***

Define a user string, such as variables, for pre- and post-processing. This string is passed to the pre- and -post-processing scripts as the environment variable `$USERSTR`.

`--user=`*username*

Authenticate the transfer using the specified username. You can use this option instead of specifying the username as part of the destination path (as *user@host:file*).

> **Note:** If you are authenticating on a Windows machine as a domain user, the transfer server strips the domain from the username. For example, `Administrator` is authenticated rather than `DOMAIN\Administrator`. For this reason, you must specify the domain explicitly.

`-v`

Run `ascp` in verbose mode. This option prints connection and authentication debug messages in the log file. For information on log files, see *IBM Aspera Enterprise Server Admin Guide: Log Files*.

`-W` **{*token_string*|*@token_file*}**

Authenticate using the authorization token string for the transfer, either as the string itself or when preceded with an `@`, the full path to the token file. This option takes precedence over the setting for the `ASPERA_SCP_TOKEN` environment variable.

`-wr`, `-wf`

Measure and report bandwidth from server to client (`-wr`) or client to server (`-wf`) before the transfer.

`-X` ***rexmsg_size***

Limit the size of retransmission requests to no larger than the specified size, in bytes. (Max: 1440)

`-Z` ***dgram_size***

Use the specified datagram size (MTU) for FASP transfers. Range: 296-65535 bytes. (Default: the detected path MTU)

As of v3.3, datagram size can be specified on the server by setting `<datagram_size>` in `aspera.conf`. The server setting overrides the client setting, unless the client is using a version of `ascp` that is older than 3.3, in which case the client setting is used. If the pre-3.3 client does not set `-Z`, the datagram size is the discovered MTU and the server logs the message "LOG Peer client doesn't support alternative datagram size".

Ascp Options for HTTP Fallback

`-I` ***cert_file***

Certify fallback transfers with the specified HTTPS certificate file.

`-j {0|1}`

Encode all HTTP transfers as JPEG files when set to 1. (Default: 0)

`-t` ***port***

Transfer via the specified server port for HTTP fallback.

`-x` ***proxy_server***

Transfer to the specified proxy server address for HTTP fallback.

`-Y` ***key_file***

Cerfity HTTPS fallback transfers using the specified HTTPS transfer key.

`-y {0|1}`

If set to "`1`", use the HTTP fallback transfer server when a UDP connection fails. (Default: 0)

Ascp General Examples

The following are examples of initiating FASP file transfers using the `ascp` command.

To describe filepaths, use single-quote (' ') and forward-slashes (/) on all platforms. Avoid the following characters in filenames: `/ \ " : ' ? > < & * |`

- **Fair-policy transfer**

  Fair-policy transfer with maximum rate 100 Mbps and minimum at 1 Mbps, without encryption, transfer all files in `\local-dir\files` to 10.0.0.2:

  ```
  # ascp -T --policy=fair -l 100m -m 1m /local-dir/files root@10.0.0.2:/remote-dir
  ```

- **Fixed-policy transfer**

  Fixed-policy transfer with target rate 100 Mbps, without encryption, transfer all files in `\local-dir\files` to 10.0.0.2:

  ```
  # ascp -T -l 100m /local-dir/files root@10.0.0.2:/remote-dir
  ```

- **Specify UDP port for transfer**

  Perform a transfer with UDP port 42000:

  ```
  # ascp -l 100m -O 42000 /local-dir/files user@10.0.0.2:/remote-dir
  ```

- **Public key authentication**

  Transfer with public key authentication using key file `<home dir>/.ssh/aspera_user_1-key local-dir/files`:

  ```
  $ ascp -T -l 10m -i ~/.ssh/aspera_user_1-key local-dir/files root@10.0.0.2:/remote-dir
  ```

- **Username or filepath contains a space**

  Enclose the target in double-quotes when spaces are present in the username and remote path:

  ```
  # ascp -l 100m local-dir/files "User Name@10.0.0.2:/remote directory"
  ```

- **Content is specified in a file pair list**

  Specify source content to transfer to various destinations in a file pair list. Source content is specified using the full file or directory path. Destination directories are specified relative to the transfer user's docroot, which is specified as a "." at the end of the `ascp` command. For example, the following is a simple file pair list, `filepairlist.txt` that lists two source folders, `folder1` and `folder2`, with two destinations, `tmp1` and `tmp2`:

  ```
  /tmp/folder1
  tmp1
  /tmp/folder2
  tmp2
  ```

  ```
  # ascp --user=user_1 --host=10.0.0.2 --mode=send --file-pair-list=/tmp/
  filepairlist.txt .
  ```

  This command and file pair list create the following directories within the transfer user's docroot on the destination:

  ```
  /tmp1/folder1
  /tmp2/folder2
  ```

- **Network shared location transfer**

Send files to a network shares location `\\1.2.3.4\nw-share-dir`, through the computer `10.0.0.2`:

```
# ascp local-dir/files root@10.0.0.2:"//1.2.3.4/nw-share-dir/"
```

- **Parallel transfer on a multicore system**

  Use parallel transfer on a dual-core system, together transferring at the rate 200Mbps, using UDP ports 33001 and 33002. Two commands are executed in different Terminal windows:

```
# ascp -C 1:2 -O 33001 -l 100m /file root@10.0.0.2:/remote-dir &
# ascp -C 2:2 -O 33002 -l 100m /file root@10.0.0.2:/remote-dir
```

- **Upload with content protection**

  Upload the file `space\file` to the server 10.0.0.2 with password protection (password: secRet):

```
$ export ASPERA_SCP_FILEPASS=secRet; ascp -l 10m --file-crypt=encrypt local-dir/file
 root@10.0.0.2:/remote-dir/
```

- **Download with content protection and decryption**

  Download from the server 10.0.0.2 and decrypt while transferring:

```
$ export ASPERA_SCP_FILEPASS=secRet; ascp -l 10m --file-crypt=decrypt root@10.0.0.2:/remote-
dir  /local-dir
```

- **Decrypt a downloaded, encrypted file**

  If the password-protected file `file1` is downloaded on the local computer without decrypting, decrypt `file1.aspera-env` (the name of the downloaded/encrypted version of `file1`) to `file1`:

```
$ export ASPERA_SCP_FILEPASS=secRet; /Library/Aspera/bin/asunprotect -o file1 file1.aspera-
env
```

- **Download through Aspera forward proxy with proxy authentication**

  User `Pat` transfers the file `/data/file1` to `/Pat_data/` on 10.0.0.2, through the proxy server at 10.0.0.7 with the proxy username`aspera_proxy` and password `pa33w0rd`. After running the command, Pat is prompted for the ascp password.

```
# ascp --proxy dnat://aspera_proxy:pa33w0rd@10.0.0.7 /data/file1 Pat@10.0.0.2:/Pat_data/
```

Test transfers using `faux://`

For information on the syntax, see *IBM Aspera Enterprise Server Admin Guide: Testing and Optimizing Transfer Performance*.

- **Transfer random data (no source storage required)**

  Transfer 20 GB of random data as user `root` to file `newfile` in the directory `/remote-dir` on 10.0.0.2:

```
#ascp --mode=send --user=root --host=10.0.0.2 faux:///newfile?20g /remote-dir
```

- **Transfer a file but do not save results to disk (no destination storage required)**

  Transfer the file `/tmp/sample` as user `root` to 10.0.0.2, but do not save results to disk:

```
#ascp --mode=send --user=root --host=10.0.0.2 /temp/sample faux://
```

- **Transfer random data and do not save result to disk (no source or destination storage required)**

  Transfer 10 MB of random data from 10.0.0.2 as user `root` and do not save result to disk:

```
#ascp --mode=send --user=root --host=10.0.0.2 faux:///dummy?10m faux://
```

Ascp File Manipulation Examples

Below are examples of using the `ascp` command to manipulate files. In each example, the client is the local computer and the server is the remote computer.

- **Upload a directory**

  Upload the directory `/data/` to the server at 10.0.0.1, and place it in the `/storage/` directory on the server:

  ```
  # ascp /src/data/  root@10.0.0.1:/storage/
  ```

- **Upload only the contents of a directory (not the directory itself) by using the `--src-base` option:**

  Upload only the contents of `/data/` to the `/storage/` directory at the destination. Strip the `/src/data/` portion of the source path and preserve the remainder of the file structure at the destination:

  ```
  # ascp --src-base=/src/data/ /src/data/ root@10.0.0.1:/storage/
  ```

- **Upload a directory and its contents to a new directory by using the `-d` opton**.

  Upload the `/data/` directory to the server and if it doesn't already exist, create the new folder `/storage2/` to contain it, resulting in `/storage2/data/` at the destination.

  ```
  # ascp -d /src/data/ root@10.0.0.1:/storage2/
  ```

- **Upload the contents of a directory, but not the directory itself, by using the `--src-base` option**:

  Upload all folders and files in the `/clips/out/` folder, but not the `out/` folder itself, to the `/in/` folder at the destination.

  ```
  # ascp -d --src-base=/clips/out/ /clips/out/ root@10.0.0.1:/in/
  ```

  Result: The source folders and their content appear in the `in` directory at the destination:

Source	Destination (docroot)	Destination without --src-base
`/clips/out/file1`	`/in/file1`	`/in/out/file1`
`/clips/out/folderA/file2`	`/in/folderA/file2`	`/in/out/folderA/file2`
`/clips/out/folderB/file3`	`/in/folderB/file3`	`/in/out/folderB/file3`

  Without `--src-base`, the example command transfers not only the contents of the `out/` folder, but the folder itself.

- **Upload only the contents of a file and a directory to a new directory by using `--src-base`**

  Upload a file, `/monday/file1`, and a directory, `/tuesday/*`, to the `/storage/` directory on the server, while stripping the srcbase path and preserving the rest of the file structure. The content is saved as `/storage/monday/file1`and `/storage/tuesday/*` on the server.

  ```
  # ascp --src-base=/data/content /data/content/monday/file1 /data/content/
  tuesday/ root@10.0.0.1:/storage
  ```

- **Download only the contents of a file and a directory to a new directory by using `--src-base`**

  Download a file, `/monday/file1`, and a directory, `/tuesday/*`, from the server, while stripping the srcbase path and preserving the rest of the file structure. The content is saved as `/data/monday/file1` and `/data/tuesday/*` on the client.

  ```
  # ascp --src-base=/storage/content root@10.0.0.1:/storage/content/monday/
  file1 root@10.0.0.1:/storage/content/tuesday/ /data
  ```

- **Move the source file on the client after it is uploaded to the server by using `--move-after-transfer`**

  Upload`file0012` to Pat's docroot on the server at 10.0.0.1, and move (not copy) the file from `C:/Users/Pat/srcdir/` to `C:/Users/Pat/Archive` on the client.

  ```
  # ascp --move-after-transfer=C:/Users/Pat/Archive C:/Users/Pat/srcdir/
  file0012 Pat@10.0.0.1:/
  ```

- **Move the source file on the server after it is downloaded to the client by using `--move-after-transfer`**

  Download `srcdir` from the server to `C:/Users/Pat` on the client, and move (not copy) `srcdir` to the archive directory `/Archive` on the server.

  ```
  # ascp --move-after-transfer=Archive Pat@10.0.0.1:/srcdir C:/Users/Pat
  ```

- **Move the source file on the client after it is uploaded to the server and preserve the file structure one level above it by using `--src-base` and `--move-after-transfer`**

  Upload `file0012` to Pat's docroot on the server at 10.0.0.1, and save it as `/srcdir/file0012` (stripped of `C:/Users/Pat`). Also move `file0012` from `C:/Users/Pat/srcdir/` to `C:/Users/Pat/Archive` on the client, where it is saved as `C:/Users/Pat/Archive/srcdir/file0012`.

  ```
  # ascp --src-base=C:/Users/Pat --move-after-transfer=C:/Users/Pat/Archive
   C:/Users/Pat/srcdir/file0012 Pat@10.0.0.1:/
  ```

- **Delete a local directory once it is uploaded to the remote server by using `--remove-after-transfer` and `--remove-empty-directories`**

  Upload `/content/` to the server, then delete its contents (excluding partial files) and any empty directories on the client.

  ```
  # ascp -k2 -E "*.partial" --remove-after-transfer --remove-empty-
  directories /data/content root@10.0.0.1:/storage
  ```

- **Delete a local directory once its contents have been transferred to the remote server by using `--src-base`, `--remove-after-transfer`, and `--remove-empty-directories`**

  Upload `/content/` to the server, while stripping the srcbase path and preserving the rest of the file structure. The content is saved as `/storage/*` on the server. On the client, the contents of `/content/`, including empty directories but excluding partial files, are deleted.

  ```
  # ascp -k2 -E "*.partial" --src-base=/data/content --remove-after-transfer
   --remove-empty-directories /data/content root@10.0.0.1:/storage
  ```

Ascp Transfers with Object Storage and HDFS

With an Aspera On Demand-entitled Aspera server installed in your cloud or on-premises object storage, you can use `ascp` to transfer to and from it. The syntax of an `ascp` command transferring to cloud or on-premises object storage depends on how you authenticate the transfer. The following options for authenticating to the object storage are described below:

- Specify the storage password or secret key in the transfer user's docroot. (Preferred method)
- Set the storage password or secret key as an environment variable.
- Specify the storage password or secret key in the command line.

Authenticating the Aspera Transfer User

You must enter the transfer user's password each time you run an `ascp` transfer, unless you either set the transfer user's password as an environment variable or set up an SSH key (token) and specify it in the command.

- **Environment Variable:**To set the transfer user's password as the value of the `ASPERA_SCP_PASS` environment variable, run the following command:

```
# export ASPERA_SCP_PASS = password
```

- **SSH Key:**To authenticate with an SSH key, configure token authorization as described in *Aspera Enterprise Server Admin Guide: Setting Up Token Authorization*. When you run the `ascp` transfer, specify the SSH key as an option:

```
# ascp -i path_to_private_key ...
```

With Docroot Configured: Authenticate in the Docroot

If your transfer user account has a docroot set, `ascp` transfers to and from AWS S3, IBM COS - S3, Google Cloud Storage, Akamai, Softlayer, and Azure are the same as regular `ascp` transfers. For command syntax examples, see *Ascp General Examples*.

For instructions on configuring a docroot for these types of storage, see *Aspera Enterprise Server Admin Guide (Linux): Docroot Path Formatting for Cloud, Object, and HDFS Storage*. You are prompted for the transfer user's password upon running these commands unless you have set the `ASPERA_SCP_PASS` environment variable or are using an SSH key, as described previously.

With No Docroot Configured: Authenticate with Environment Variables

You can set an environment variable (`ASPERA_DEST_PASS`) with the storage password or access key using the command below:

```
# export ASPERA_DEST_PASS = secret_key
```

With this and `ASPERA_SCP_PASS` set, run `ascp` with the syntax listed in the table above, but you do not need to include the storage password or access key, and are not prompted for the Aspera password upon running the command.

Note: The `ASPERA_DEST_PASS` variable is not applicable to Google Cloud Storage or AWS S3 using IAM roles.

With No Docroot Configured: Authenticate in the Command Line

If you do not have a docroot configured and do not set an environment variable (described previously), you must authenticate in the command line. In the examples below, you include the storage password or secret key as part of the destination path. You are prompted for the transfer user's password upon running these commands unless you have set the `ASPERA_SCP_PASS` environment variable or are using an SSH key, as described above.

Storage Platform	ascp Syntax and Examples

AWS S3	• If you are using IAM roles, you do not need to specify the access ID or secret key for your S3 storage.
	Upload syntax:
	`# ascp options --mode=send --user=username --host=s3_server_addr source_files s3://access_id:secret_key@s3.amazonaws`
	Upload example:
	`# ascp --mode=send --user=bear --host=s3.asperasoft.com bigfile.txt s3://1K3C18FBWF9902:GEyU...AqXuxtTVHWtc@s3.amazonaws.com/demos2014`

Storage Platform	ascp Syntax and Examples

	Dowload syntax:
	`# ascp options --mode=recv --user=username --host=s3_server_addr s3://access_id:secret_key@s3.amazonaws.com/my_bucke my_source_path destination_path`
	Download example:
	`# ascp --mode=recv --user=bear --host=s3.asperasoft.com s3://1K3C18FBWF9902:GEyU...AqXuxtTVHWtc@s3.amazonaws.com/demos2014/bigfile.txt /tmp/`

Azure	Upload syntax:
	`# ascp options --mode=send --user=username --host=server_address source_files azu://storage_account:storage_access_k`
	Upload example:
	`# ascp --mode=send --user=AS037d8eda429737d6 --host=dev920350144d2.azure.asperaondemand.com bigfile.txt azu://astransfer:zNfMtU...nBTkhB@blob.core.windows.net/abc`
	Dowload syntax:
	`# ascp options --mode=recv --user=username --host=server azu://storage_account:storage_access_key@blob.core.windows.`
	Download example:
	`# ascp --mode=recv --user=AS037d8eda429737d6 --host=dev920350144d2.azure.asperaondemand.com azu://astransfer:zNfMtU...nBTkhB@blob.core.windows.net/abc /downloads`

Google Cloud Storage	**Note:** The examples below require that the VMI running the Aspera server is a Google Compute instance.
	`# ascp options --mode=send --user=username --host=server_address source_files gs:///my_bucket/my_path`
	Upload example:
	`# ascp --mode=send --user=bear --host=10.0.0.5 bigfile.txt gs:///2017_transfers/data`
	Dowload syntax:
	`# ascp options --mode=recv --user=username --host=server gs:///my_bucket/my_path/source_file destination_path`
	Download example:
	`# ascp --mode=recv --user=bear --host=10.0.0.5 gs:///2017_transfers/data/bigfile.txt /data`

Storage Platform	ascp Syntax and Examples
HDFS	Aspera recommends running `ascp` transfers with HDFS with a docroot configured.

IBM COS - S3	Upload syntax:
	`# ascp options --mode=send --user=username --host=server_address source_files s3://access_id:secret_key@accessor_end`
	Upload example:
	`# ascp --mode=send --user=bear --host=s3.asperasoft.com bigfile.txt s3://3ITI3OIUFEH233:KrcEW...AIuwQ@38.123.76.24/demo2017`
	Dowload syntax:
	`# ascp options --mode=send --user=username --host=server_address s3://access_id:secret_key@accessor_endpoint/vault_n source_files destination_path`
	Download example:
	`# ascp --mode=send --user=bear --host=s3.asperasoft.com s3://3ITI3OIUFEH233:KrcEW...AIuwQ@38.123.76.24/demo2017 /tmp/`

IBM Cloud Object Storage (COS) - Swift and IBM Bluemix	Aspera recommends running `ascp` transfers with IBM Cloud Object Storage (COS) - Swift and IBM Bluemix with a docroot configured.

OpenStack Swift	Upload syntax:
	`# ascp options --mode=send --user=username --host=ip_addr source_files swift://account_id:api_key@auth_url/my_bucket`
	Example Upload:
	`# ascp --mode=send --user=bear --host=192.155.218.130 bigfile.txt swift://XYZO...46-2:bob:437e...bc16@sjc01.objectstorage.service.networklayer.co test`
	Dowload syntax:
	`# ascp options --mode=recv --user=username --host=ip_addr swift://account_id:api_key@auth_url/my_bucket/my_source_path destination_path`
	Download example:
	`# ascp --mode=recv --user=bear --host=192.155.218.130 swift://XYZO...46-2:bob:437e29...f616@sjc01.objectstorage.service.networklayer. test/bigfile.txt /tmp/`
	Note: Swift requires additional Trapd configuration settings that can be included as queries attached to the docroot, with the format *docroot*?*setting*.

Storage Platform	ascp Syntax and Examples
	For example, for an upload to IBM COS - Swift, the path is written as follows: `swift://XYZO...46-2:bob:437e...bc16@sjc01.objectstorage.service.networklaye test?aspera.swift.endpoint.auth-path=/auth/v1.0`

Applying Filters to Include and Exclude Files

Filters allow you to refine the list of files (or directories) designated for transfer. With filters, you indicate which files in the transfer list to skip or include. At runtime, `ascp` looks for filters in two locations: on the `ascp` command line, and in `aspera.conf`. Filters can be set in the `aspera.conf` file either from the GUI, or by modifying it directly with an editor or `asconfigurator`. When filtering rules are found in `aspera.conf`, they are applied *before* rules on the command line. If no filtering rules are specified, `ascp` transfers all source files in the transfer list. This topic describes filtering using option flags on the `ascp` command line.

Note: Filter settings apply only when the server is acting as a client. Servers cannot exclude files or directories uploaded or downloaded by remote clients.

Specifying Rules on the Command Line

To specify filtering rules on the `ascp` command line, use the `-E` and `-N` options:

`-E` *pattern* Exclude files or directories matching *pattern*.
`-N` *pattern* Include files or directories matching *pattern*.

Each rule consists of a `-E` or `-N` option and its pattern. A pattern can be a file or directory name, or a set of names expressed with UNIX *glob* patterns.

To determine which files to transfer, each file in the set of source files to transfer (the transfer list) is evaluated by the filters as follows:

1. `ascp` compares the next file (or directory) in the transfer list to the first rule.
2. If the file matches the pattern, `ascp` includes it (`-N`) or excludes it (`-E`) and for this file, filtering stops.
3. If the file does not match, `ascp` compares it with the next rule and repeats the process for each rule until a match is found or until all rules have been tried.
4. If the file never matches any rules, it is included in the transfer.

Filtering operates only on the set of files and directories in the transfer list. That is, an include option (`-N`) cannot add files or directories that are not already part of the transfer list.

Filtering is a process of exclusion, and `-N` rules act as overrides to any `-E` rules that follow them. For example, consider the following example command:

```
$ ascp -N 'file2' -E 'file[0-9]' /tmp/L/file* user1@examplehost:/tmp
```

The transfer set is `file*` (all files that start with `file`). If `file1`, `file2`, and `fileA` are in `/tmp/L`, they are filtered as follows:

1. When `file1` is compared with the first rule (`-N`), no match is found, and filtering continues. When `file1` is compared with the second rule (`-E`), there is a match; `file1` is therefore excluded from transfer, and filtering stops for `file1`.
2. When `file2` is compared with the first rule, there is a match; `file2` is therefore included in the transfer, and filtering stops for `file2`.
3. When `fileA` is compared with the first rule, no match is found. When it is compared with the second rule, again no match is found. Because no further rules exclude it, `fileA` is therefore included in the transfer.

If directories or files reside in directories that have already been excluded, they will also be excluded and therefore not checked against any further rules. Thus, with the command-line options `-E '/above/' -N '/above/below'`, the file `/above/below` is never considered because its parent directory `/above/` has already been excluded.

Creating Rule Patterns

In order to filter directories and files to be transferred, their names are matched against patterns (globs) that include wildcards and special characters. The patterns use the standard globbing syntax found in UNIX systems as well as several Aspera extensions to the standard.

Character case: Case always matters, even if the scanned file system does not enforce such a distinction. For example, "debug" does not match "Debug". To match both, the pattern should be "[Dd]ebug".

Single quotes: Patterns must be interpreted only by `ascp`, not by the command shell. For this reason, patterns that contain wildcards should be surrounded by single quotes to protect them from expansion by the shell. (Even if patterns contain no wildcards, they can still be surrounded by single quotes.)

Partial matches: With globs, unlike standard regular expressions, the entire filename or directory name must match the pattern. That is, `abcdef` matches the pattern `abc*f` but `abcdefg` does not.

Pattern position: A pattern given with `-N` will match a path only if it falls directly under the transfer directory. However, a pattern given with `-E` will match a path regardless of where (which level) the path falls under the transfer directory. For example, given the pattern `'zzz*'` and a transfer directory `AAA`:

- The `-N` option matches only if the path to file (or directory) `zzz` falls *directly* under `AAA`. That is, `AAA/zzz`.
- The `-E` option matches regardless of the where the path to file (or directory) `zzz` falls under `AAA`. For example, `AAA/abc/def/zzz`.

Standard Globbing: Wildcards and Special Characters

`/`	The only recognized path separator.
`\`	Quotes any character literally, including itself. The `\` character is exclusively a quoting operator, not a path separator.
`*`	Matches zero or more characters, except a `/`, or the `.` when preceded immediately by a `/` character.
`?`	Matches any single character, except a `/`, or a `.` when preceded immediately by a `/` character.
`[ … ]`	Matches exactly one of a set of characters, except a `/` or a `.` preceded immediately by a `/` character.
`[^… ]`	When `^` is the first character, matches exactly one character *not* in the set.
`[!… ]`	When `!` is the first character, matches exactly one character *not* in the set.
`[`*x*`-`*x*`]`	Matches exactly one of a range of characters.
`[:`*xxxxx*`:]`	For details about this type of wildcard, see any POSIX-standard guide to globbing.

Globbing Extensions: Wildcards and Special Characters

`/**`	Like `*` but also matches the `/` character, or a `.` preceded immediately by a `/` (that is, the `.` in `/.`).
`*` or `/**` at end of pattern	Matches both directories and files.
`/` at end of pattern	Matches directories only. With `-N`, no files under matched directories or their subdirectories are included in the transfer. All subdirectories are still included,

	although their files will not be included. However, with `-E`, excluding a directory also excludes all files and subdirectories under it.
no `/` or `*` at end of pattern	Matches files only.
`/` at start of pattern	Must match the entire string from the root of the transfer set. (Note: The leading `/` does not refer to the system root or the docroot.)

Standard Globbing Examples

Wildcard	Example	Matches	Does Not Match
`/`	`abc/def/xyz`	`abc/def/xyz`	`abc/def`
`\`	`abc\?`	`abc?`	`abc\? abc/D abcD`
`*`	`abc*f`	`abcdef abc.f`	`abc/f abcefg`
`?`	`abc??`	`abcde abc.z`	`abcdef abc/d abc/.`
`[ … ]`	`[abc]def`	`adef cdef`	`abcdef ade`
`[^… ]`	`[^abc]def`	`zdef .def 2def`	`bdef /def /.def`
`[!… ]`	`[!abc]def`	`zdef .def 2def`	`cdef /def /.def`
`[:`*xxxxx*`:]`	`[[:lower:]]def`	`cdef ydef`	`Adef 2def .def`

Globbing Extension Examples

Wildcard	Example	Matches	Does Not Match
`/**`	`a/**/f`	`a/f a/.z/f a/d/e/f`	`a/d/f/ za/d/f`
`*` at end of rule	`abc*`	`abc/ abcfile`	
`/**` at end of rule	`abc/**`	`abc/.file abc/d/e/`	`abc/`
`/` at end of rule	`abc/*/`	`abc/dir`	`abc/file`
no `/` at end of rule	`abc`	`abc` (file)	`abc/`
`/` at start of rule	`/abc/def`	`/abc/def`	`xyz/abc/def`

Rule Composition

Example	Transfer Result
`-N` *rule*	Includes all files and directories whose names match *rule*. Because there is no `-E`, all the originally specified files and directories are included anyway; in other words, by itself, a `-N` rule does nothing.
`-N` *rule1* `-E` *rule2*	Includes all files and directories whose names match *rule1*. Excludes all that match *rule2*, *except* those that also matched *rule1*.
`-E` *rule*	Excludes all files and directories whose names match *rule*.
`-E` *rule1* `-N` *rule2*	Excludes all files and directories whose names match *rule1*. Because there is no `-E` following the `-N`, all files and directories not already excluded by the preceding `-E` are included anyway; in other words, a trailing `-N` rule does nothing to change the result.

Testing Your Filter Rules

If you plan to use filtering rules, it's best to test them first. An easy way to test filtering rules, or to learn how they work, is to set up source and destination directories and use `demo.asperasoft.com` as the Aspera server:

1. On your computer, create a small set of directories and files that generally matches a file set you typically transfer. Since filenames are all that matter, the files can be small.
2. Place the file set in an accessible location, for example `/tmp/src`.
3. Upload the file set to the Aspera demo server as user "aspera". Specify the demo-server target directory `Upload`. You will be prompted for the password, which is "`demoaspera`":

   ```
   $ ascp /tmp/src aspera@demo.asperasoft.com:Upload/
   ```

4. Create a destination directory on your computer, for example `/tmp/dest`.
5. You can now download your files from the demo server to `/tmp/dest`, running the `ascp` commands with `-N` and `-E` to test your filtering rules. For example:

   ```
   $ ascp -N 'wxy/**' -E 'def' aspera@demo.asperasoft.com:Upload/src/abc/ /
   tmp/dest
   ```

6. Compare the destination directory with the source to determine whether files were filtered as expected.

   ```
   $ diff -r dest/ src/
   ```

   The `diff` output will show the missing (untransferred) files and directories.

Example Filter Rules

The example rules below are based on running a command such as the following to download a directory `AAA` from `demo.asperasoft.com` to `/tmp/dest`:

```
$ ascp rules aspera@demo.asperasoft.com:Upload/AAA /tmp/dest
```

The examples below use the following file set:

```
AAA/abc/def
AAA/abc/.def
AAA/abc/.wxy/def
AAA/abc/wxy/def
AAA/abc/wxy/.def
AAA/abc/wxy/tuv/def
AAA/abc/xyz/def/wxy
AAA/wxyfile
AAA/wxy/xyx/
AAA/wxy/xyxfile
```

Key for interpreting example results below:

`< xxx/yyy` = Excluded
`xxx/yyy` = Included
`zzz/` = directory name
`zzz` = filename

(1) Transfer everything except files and directories starting with "`.`":

```
-N '*' -E 'AAA/**'
```

Results:

```
AAA/abc/def
```

```
AAA/abc/wxy/def
AAA/abc/wxy/tuv/def
AAA/abc/xyz/def/wxy
AAA/wxyfile
AAA/wxy/xyx/
AAA/wxy/xyxfile
< AAA/abc/.def
< AAA/abc/.wxy/def
< AAA/abc/wxy/.def
```

(2) Exclude directories and files whose names start with `wxy`:

```
-E 'wxy*'
```

Results:

```
AAA/abc/def
AAA/abc/.def
AAA/abc/.wxy/def
AAA/abc/xyz/def/
< AAA/abc/wxy/def
< AAA/abc/wxy/.def
< AAA/abc/wxy/tuv/def
< AAA/abc/xyz/def/wxy
< AAA/wxyfile
< AAA/wxy/xyx/
< AAA/wxy/xyxfile
```

(3) Include directories and files that start with "`wxy`" if they fall directly under `AAA`:

```
-N 'wxy*' -E 'AAA/**'
```

Results:

```
AAA/wxy/
AAA/wxyfile
< AAA/abc/def
< AAA/abc/.def
< AAA/abc/.wxy/def
< AAA/abc/wxy/def
< AAA/abc/wxy/.def
< AAA/abc/wxy/tuv/def
< AAA/abc/xyz/def/wxy
< AAA/wxy/xyx/
< AAA/wxy/xyxfile
```

(4) Include directories and files at any level that start with `wxy`, but do not include dot-files, dot-directories, or any files under the `wxy` directories (unless they start with `wxy`). However, subdirectories under `wxy` will be included:

```
-N '*/wxy*' -E 'AAA/**'
```

Results:

```
AAA/abc/wxy/tuv/
AAA/abc/xyz/def/wxy
AAA/wxyfile
AAA/wxy/xyx/
< AAA/abc/def
< AAA/abc/.def
< AAA/abc/.wxy/def
< AAA/abc/wxy/def     *
```

```
< AAA/abc/wxy/.def
< AAA/abc/wxy/tuv/def
< AAA/wxy/xyxfile
```

* Even though `wxy` is included, `def` is excluded because it's a file.

(5) Include `wxy` directories and files at any level, even those starting with "`.`":

```
-N '*/wxy*' -N '*/wxy/**' -E 'AAA/**'
```

Results:

```
AAA/abc/wxy/def
AAA/abc/wxy/.def
AAA/abc/wxy/tuv/def
AAA/abc/xyz/def/wxy
AAA/wxyfile
AAA/wxy/xyx/
AAA/wxy/xyxfile
< AAA/abc/def
< AAA/abc/.def
< AAA/abc/.wxy/def
```

(6) Exclude directories and files starting with `wxy`, but only those found at a specific location in the tree:

```
-E '/AAA/abc/wxy*'
```

Results:

```
AAA/abc/def
AAA/abc/.def
AAA/abc/.wxy/def
AAA/abc/xyz/def/wxy
AAA/wxyfile
AAA/wxy/xyx/
AAA/wxy/xyxfile
< AAA/abc/wxy/def
< AAA/abc/wxy/.def
< AAA/abc/wxy/tuv/def
```

(7) Include the `wxy` directory at a specific location, and include all its subdirectories and files, including those starting with "`.`":

```
-N 'AAA/abc/wxy/**' -E 'AAA/**'
```

Results:

```
AAA/abc/wxy/def
AAA/abc/wxy/.def
AAA/abc/wxy/tuv/def
< AAA/abc/def
< AAA/abc/.def
< AAA/abc/.wxy/def
< AAA/abc/xyz/def/wxy
< AAA/wxyfile
< AAA/wxy/xyx/
< AAA/wxy/xyxfile
```

Creating SSH Keys (Command Line)

Public key authentication (SSH Key) is a more secure alternative to password authentication that allows users to avoid entering or storing a password, or sending it over the network. Public key authentication uses the client computer to generate the key-pair (a public key and a private key). The public key is then provided to the remote computer's administrator to be installed on that machine.

1. Create a `.ssh` directory in your home directory if it does not already exist:

   ```
   $ mkdir /Users/username/.ssh
   ```

   Go to the `.ssh` folder:

   ```
   $ cd /Users/username/.ssh
   ```

2. Run `ssh-keygen` to generate an SSH key-pair.

   Run the following command in the `.ssh` folder to create a key pair. For *`key_type`*, specify either RSA (`rsa`) or ED25519 (`ed25519`). At the prompt for the key-pair's filename, press ENTER to use the default name `id_rsa` or `id_ed25519`, or enter a different name, such as your username. For a passphrase, you can either enter a password, or press return twice to leave it blank:

   ```
   # ssh-keygen -t key_type
   ```

   Note: When you run `ascp` in FIPS mode (`<fips_enabled>` is set to `true` in `aspera.conf`), and you use passphrase-protected SSH keys, you must either (1) use keys generated by running `ssh-keygen` in a FIPS-enabled system, or (2) convert existing keys to a FIPS-compatible format using a command such as the following:

   ```
   # openssl pkcs8 -topk8 -v2 aes128 -in id_rsa -out new-id_rsa
   ```

3. Retrieve the public key file.

   The key-pair is generated to your home directory's `.ssh` folder. For example, assuming you generated the key with the default name `id_rsa`:

   `/Users/username/.ssh/id_rsa.pub`

   Provide the public key file (for example, `id_rsa.pub`) to your server administrator so that it can be set up for your server connection.

4. Start a transfer using public key authentication with the `ascp` command.

   To transfer files using public key authentication on the command line, use the option `-i` *private_key_file*. For example:

   ```
   $ ascp -T -l 10M -m 1M  -i ~/.ssh/id_rsa  myfile.txt  jane@10.0.0.2:/space
   ```

   In this example, you are connecting to the server (`10.0.0.2`, directory `/space`) with the user account `jane` and the private key `~/.ssh/id_rsa`.

Reporting Checksums

File checksums are useful for trouble-shooting file corruption, allowing you to determine at what point in the transfer file corruption occurred. Aspera servers can report source file checksums that are calculated on-the-fly during transfer and then sent from the source to the destination. To do so, the transfer must meet both of the following requirements:

- Both the server and client computers must be running Enterprise Server, Connect Server, or Point-to-Point Client version 3.4.2 or higher.
- The transfer must be encrypted. Encryption is enabled by default.

The user on the destination can calculate a checksum for the received file and compare it (manually or programmatically) to the checksum reported by the sender. The checksum reported by the source can be retrieved in the destination logs, a manifest file, in Aspera Console, or as an environment variable. Instructions for comparing checksums follow the instructions for enabling checksum reporting.

Checksum reporting is disabled by default. You can enable and configure checksum reporting on the server by using the following methods:

- Edit `aspera.conf` with `asconfigurator`.
- Set options in the client GUI.
- Set `ascp` command-line options (per-transfer configuration).

Command-line options override the settings in `aspera.conf` and the GUI.

Overview of Checksum Configuration Options

`asconfigurator` Option **`ascp` Option**	**Description**
`file_checksum` `--file-checksum=`*`type`*	Enable checksum reporting and specify the type of checksum to calculate for transferred files. `any` - Allow the checksum format to be whichever format the client requests. (Default in `aspera.conf`) `md5` - Calculate and report an MD5 checksum. `sha1` - Calculate and report a SHA-1 checksum. `sha256` - Calculate and report a SHA-256 checksum. `sha384` - Calculate and report a SHA-384 checksum. `sha512` - Calculate and report a SHA-512 checksum. **Note:** The default value for the `ascp` option is `none`, in which case the reported checksum is the one configured on the server, if any.
`file_manifest` `--file_manifest=`*`output`*	The file manifest is a file that contains a list of content that was transferred in a transfer session. The file name of the file manifest is automatically generated from the transfer session ID. When set to `none`, no file manifest is created. (Default) When set to `text`, a text file is generated that lists all files in each transfer session.
`file_manifest_path` `--file_manifest_path=`*`path`*	The location where manifest files are written. The location can be an absolute path or a path relative to the transfer user's home directory. If no path is specified (default), the file is generated under the destination path at the receiver, and under the first source path at the sender. **Note:** File manifests can be stored only locally. Thus, if you are using S3 or other non-local storage, you must specify a local manifest path.

Enabling checksum reporting by editing `aspera.conf`

To enable checksum reporting, run the following `asconfigurator` command:

```
# asconfigurator -x "set_node_data;file_checksum,checksum"
```

To enable and configure the file manifest where checksum report data is stored, run the following commands:

```
# asconfigurator -x "set_node_data;file_manifest,text"
# asconfigurator -x "set_node_data;file_manifest_path,filepath"
```

These commands create lines in `aspera.conf` as shown in the following example, where checksum type is `md5`, file manifest is enabled, and the path is `/tmp`.

```
<file_system>
    ...
    <file_checksum>md5</file_checksum>
    <file_manifest>text</file_manifest>
    <file_manifest_path>/tmp</file_manifest_path>
    ...
</file_system>
```

Enabling checksum reporting in an ascp session

To enable checksum reporting on a per-transfer-session basis, run `ascp` with the `--file-checksum=`*`hash`* option, where *`hash`* is `sha1`, `md5`, `sha-512`, `sha-384`, `sha-256`, or `none` (the default).

Enable the manifest with the option `--file-manifest=`*output* where *output* is either `text` or `none`. You can set the path to the manifest file with the option `--file-manifest-path=`*path*.

For example:

```
# ascp --file-checksum=md5 --file-manifest=text --file-manifest-path=/
tmp file aspera_user_1@189.0.202.39:/destination_path
```

Setting up a Pre/Post-processing Script

An alternative to enabling and configuring the file manifest to collect checksum reporting is to set up a pre/post-processing script to report the values.

The checksum of a transferred file is stored in the pre/post environment variable `FILE_CSUM`, which can be used in pre/post scripts to output file checksums. For example, the following script outputs the checksum to the file `/tmp/cksum.log`:

```
#!/bin/bash
if [ $TYPE == File ]; then
    if [ $STARTSTOP == Stop ]; then
        echo "The file is: $FILE" >> /tmp/cksum.log
        echo "The file checksum is: $FILE_CSUM" >> /tmp/cksum.log
        chmod 777 $FILE
    fi
fi
```

For information on pre- and post-processing scripts and environment variables, see *IBM Aspera Enterprise Server Admin Guide: Testing and Optimizing Transfer Performance*.

Comparing Checksums

If you open a file that you downloaded with Aspera and find that it is corrupted, you can determine when the corruption occurred by comparing the checksum that is reported by Aspera to the checksums of the files on the destination and on the source.

1. Retrieve the checksum that was calculated by Aspera as the file was transferred.
   - If you specified a file manifest and file manifest path as part of an `ascp` transfer or pre/post processing script, the checksums are in that file in the specified location.

- If you specified a file manifest and file manifest path in the GUI or `aspera.conf`, the checksums are in a file that is named `aspera-transfer-transfer_id-manifest.txt` in the specified location.

2. Calculate the checksum of the corrupted file. This example uses the MD5 checksum method; replace MD5 with the appropriate checksum method if you use a different one.

```
$ md5 filepath
```

3. Compare the checksum reported by Aspera with the checksum that you calculated for the corrupted file.
   - If they do not match, then corruption occurred as the file was written to the destination. Download the file again and confirm that it is not corrupted. If it is corrupted, compare the checksums again. If they do not match, investigate the write process or attempt another download. If they match, continue to the next step.
   - If they match, then corruption might have occurred as the file was read from the source. Continue to the next step.
4. Calculate the checksums for the file on the source. These examples use the MD5 checksum method; replace MD5 with the appropriate checksum method if you use a different one.

   Windows:

```
> CertUtil -hashfile filepath MD5
```

   Mac OS X:

```
$ md5 filepath
```

   Linux and Linux on z Systems:

```
# md5sum filepath
```

   AIX:

```
# csum -h MD5 filepath
```

   Solaris:

```
# digest -a md5 -v filepath
```

5. Compare the checksum of the file on the source with the one reported by Aspera.
   - If they do not match, then corruption occurred when the file was read from the source. Download the file again and confirm that it is not corrupted on the destination. If it is corrupted, continue to the next step.
   - If they match, confirm that the source file is not corrupted. If the source file is corrupted, replace it with an uncorrupted one, if possible, and then download the file again.

Comparison of Ascp and Ascp4 Options

Many command-line options are the same for `ascp` and `ascp4`; however, some options are available for only one or the behavior of an option is different. The following table lists the options that are available only for `ascp` or `ascp4`, and the options that are available with both. If the option behavior is different, the `ascp` option has ** added to the end and the difference is described following the table.

ascp	ascp4
`-6`	
`-@[`*range_low*:*range_high*`]`	
`-A, --version`	`-A, --version`

ascp	**ascp4**
`--apply-local-docroot`	
`-C` *nodeid:nodecount*	
`-c` *cipher*	
`--check-sshfp=`*fingerprint*	
	`--chunk-size=`*bytes*
	`--compare=`*method*
	`--compression=`*method*
	`--compression-hint=`*num*
`-D | -DD | -DDD`	
`-d`	
	`--delete-after`
	`--delete-before`
`--delete-before-transfer`	`--delete-before-transfer`
`--dest64`	
`-E` *pattern*	`-E` *pattern*
`-e` *prepost_filepath*	
	`--exclude-newer-than=`*mtime*
	`--exclude-older-than=`*mtime*
`-f` *config_file*	
	`--faspmgr-io`
`--file-checksum=`*hash*	
`--file-crypt={encrypt|decrypt}`	
`--file-list=`*filepath*	`--file-list=`*filepath*
`--file-manifest={none|text}`	
`--file-manifest-path=`*directory*	
`--file-manifest-inprogress-suffix=`*suffix*	
`--file-pair-list=`*filepath*	
`-G` *write_size*	
`-g` *read_size*	
`-h, --help`	`-h, --help`
`-i` *private_key_file_path***	`-i` *private_key_file_path*
`-K` *probe_rate*	
`-k {0|1|2|3}`	`-k {0|1|2|3}`
`--keepalive`	

ascp	**ascp4**
`-l` *max_rate*	`-l` *max_rate*
`-L` *local_log_dir*[:*size*]	`-L` *local_log_dir*[:*size*]
`-m` *min_rate*	`-m` *min_rate*
	`--memory=`*bytes*
	`--meta-threads=`*num*
`--mode={send|recv}`	`--mode={send|recv}`
`--move-after-transfer=`*archivedir*	
`--multi-session-threshold=`*threshold*	
`-N` *pattern*	`-N` *pattern*
	`--no-open`
	`--no-read`
	`--no-write`
`-O` *fasp_port*	`-O` *fasp_port*
`--overwrite=`*method*	`--overwrite=`*method*
`-P` *ssh-port*	`-P` *ssh-port*
`-p`	`-p`
`--partial-file-suffix=`*suffix*	
`--policy={fixed|high|fair|low}`	`--policy={fixed|high|fair|low}`
`--precalculate-job-size`	
`--preserve-access-time`	
`--preserve-acls=`*mode*	
`--preserve-creation-time`	
`--preserve-file-owner-gid`	`--preserve-file-owner-gid`
`--preserve-file-owner-uid`	`--preserve-file-owner-uid`
`--preserve-modification-time`	
`--preserve-source-access-time`	
`--preserve-xattrs=`*mode*	
`--proxy=`*proxy_url*	
`-q`	`-q`
`-R` *remote_log_dir*	`-R` *remote_log_dir*
	`--read-threads=`*num*
	`--remote-memory=`*bytes*
`--remote-preserve-acls=`*mode*	
`--remote-preserve-xattrs=`*mode*	

ascp	ascp4
`--remove-after-transfer`	
`--remove-empty-directories`	
`--remove-empty-source-directory`	
	`--resume` (similar to `-k`)
`--retry-timeout=`*secs*	
`-S` *remote_ascp*	
`--save-before-overwrite`	
	`--scan-threads=`*num*
`--source-prefix=`*prefix*	
`--source-prefix64=`*prefix*	
	`--sparse-file`
`--src-base=`*prefix*	`--src-base=`*prefix*
`--symbolic-links=`*method*	`--symbolic-links=`*method*
`-T`	`-T`
`-u` *user_string*	`-u` *user_string*
`--user=`*username*	`--user=`*username*
`-v`	
`-W` *token_string* `|` `@`*token_filepath*	
`-w{r|f}`	
`-X` *rexmsg_size*	`-X` *rexmsg_size*
`-Z` *dgram_size*	`-Z` *dgram_size*

Differences in Option Behavior

`-i`, SSH key authentication

With `ascp`, the argument for `-i` can be just the file name of the private key file and `ascp` automatically looks in the `.ssh` directory of the user's home directory. With `ascp4`, the full or relative path to the private key file must be specified.

Ascp FAQs

1. **How do I control the transfer speed?**

You can specify a transfer policy that determines how a FASP transfer utilizes the network resource, and you can specify target and minimum transfer rates where applicable. In an `ascp` command, use the following flags to specify transfer policies that are fixed, fair, high, or low:

Policy	Command template
`Fixed`	`--policy=fixed -l target_rate`
`Fair`	`--policy=fair -l target_rate -m min_rate`
`High`	`--policy=high -l target_rate -m min_rate`
`Low`	`--policy=low -l target_rate -m min_rate`

The policies have the following characteristics:

`fixed`

Attempt to transfer at the specified target rate, regardless of network capacity. Content is transferred at a constant rate and the transfer finishes in a guaranteed time. The `fixed` policy can consume most of the network's bandwidth and is not recommended for most types of file transfers. It requires setting a maximum (target) rate (`-l` option).

`high`

Adjust the transfer rate to fully utilize the available bandwidth up to the maximum rate. When congestion occurs, the transfer rate is twice as fast as a fair-policy transfer. The `high` policy requires the setting of maximum (target) and minimum transfer rates (`-l` and `-m`).

`fair`

Adjust the transfer rate to fully utilize the available bandwidth up to the maximum rate. When congestion occurs, bandwidth is shared fairly by transferring at an even rate. The `fair` policy requires the setting of maximum (target) and minimum transfer rates (`-l` and `-m`).

`low`

Adjust the transfer rate to use the available bandwidth up to the maximum rate. Similar to fair mode, but less aggressive when sharing bandwidth with other network traffic. When congestion occurs, the transfer rate is reduced to the minimum rate until other traffic decreases.

2. **What transfer speed should I expect? How do I know if something is "wrong" with the speed?**

Aspera's FASP transport has no theoretical throughput limit. Other than the network capacity, the transfer speed may be limited by rate settings and resources of the computers. To verify that your system's FASP transfer can fulfill the maximum bandwidth capacity, prepare a client machine to connect to this computer, and test the maximum bandwidth.

> **Note:** This test typically occupies most of a network's bandwidth. Aspera recommends this test be performed on a dedicated file transfer line or during a time of low network activity.

On the client machine, start a transfer with fixed bandwidth policy. Start with a lower transfer rate and gradually increase the transfer rate toward the network bandwidth (for example, 1 MB, 5 MB, 10 MB, and so on). Monitor the transfer rate; at its maximum, it should be slighly below your available bandwidth:

```
$ ascp -l 1m source-file destination
```

To improve the transfer speed, also consider upgrading the following hardware components:

Component	Description
Hard disk	The I/O throughput, the disk bus architecture (such as RAID, IDE, SCSI, ATA, and Fiber Channel).
Network I/O	The interface card, the internal bus of the computer.
CPU	Overall CPU performance affects the transfer, especially when encryption is enabled.

3. **How do I ensure that if the transfer is interrupted or fails to finish, it will resume without retransferring the files?**

   Use the `-k` flag to enable resume, and specify a resume rule:

   `-k 0` – Always retransfer the entire file.
   `-k 1` – Compare file attributes and resume if they match, and retransfer if they do not.
   `-k 2` – Compare file attributes and the sparse file checksums; resume if they match, and retransfer if they do not.
   `-k 3` – Compare file attributes and the full file checksums; resume if they match, and retransfer if they do not.

   Corruption or deletion of the `.asp-meta` file associated with an incomplete transfer will often result in a permanently unusable destination file even if the file transfer resumed and successfully transferred.

4. **How does Aspera handle symbolic links?**

   The `ascp` command follows symbolic links by default. This can be changed using `--symbolic-links=`*`method`* with the following options:

   `follow`
   Follow symbolic links and transfer the linked files. (Default)

   `copy`
   Copy only the alias file. If a file with the same name is found at the destination, the symbolic link is not copied.

   `copy+force`
   Copy only the alias file. If a file (not a directory) with the same name is found at the destination, the alias replaces the file. If the destination is a symbolic link to a directory, it's not replaced.

   `skip`
   Skip symbolic links. Do not copy the link or the file it points to.

   Important: On Windows, the only option is `skip`.

5. **What are my choices for overwriting files on the destination computer?**

   In `ascp`, you can specify the `--overwrite=`*`method`* rule with the following method options:

   `never`
   Never overwrite the file. However, if the parent folder is not empty, its access, modify, and change times may still be updated.

   `always`
   Always overwrite the file.

   `diff`
   Overwrite the file if different from the source. If a complete file at the destination is the same as a file on the source, it is not overwritten. Partial files are overwritten or resumed depending on the resume policy.

   `diff+older`
   Overwrite the file if older and also different than the source. For example, if the destination file is the same as the source, but with a different timestamp, it will not be overwritten. Plus, if the destination file is different than the source, but newer, it will not be overwritten.

`older`

Overwrite the file if its timestamp is older than the source timestamp.

Interaction with resume policy (-k): If the overwrite method is `diff` or `diff+older`, difference is determined by the resume policy (`-k {0|1|2|3}`). If `-k 0` or no `-k` is specified, the source and destination files are always considered different and the destination file is always overwritten. If `-k 1`, the source and destination files are compared based on file attributes (currently file size). If `-k 2`, the source and destination files are compared based on sparse checksums. If `-k 3`, the source and destination files are compared based on full checksums.

ascp4: Transferring from the Command Line with A4

Introduction to A4

Aspera A4 is an optimized transfer engine based on FASP technology. A4 is designed for sending extremely large sets of individual files efficiently, and it supports UDP multicast. The executable, `ascp4`, is similar to `ascp` and shares many of the same options and capabilities. For more information on using `ascp4` for UDP multicast, see the *IBM Aspera Faspstream User Guide*.

A4 Command Reference

Supported environment variables, the general syntax, and command options for A4 are described in the following sections. `ascp4` exits with a `0` on success or a `1` on error. The error code is logged in the `ascp4` log file.

Note: Not all standard `ascp` options are available with `ascp4`. For more information, see *Comparison of Ascp and Ascp4 Options*. Additionally, `ascp4` transfers fail if the user's docroot is a symlink, whereas `ascp` supports symlink docroots.

ascp4 Syntax

```
ascp4 options [[user@]srcHost:]source_file1[,source_file2,...]
 [[user@]destHost:]target_path
```

User

The username of the Aspera transfer user can be specified as part of the filepath or with the `--user` option. If you do not specify a username for the transfer, the local username is authenticated by default.

Note: If you are authenticating on a Windows machine as a domain user, the transfer server strips the domain from the username. For example, `Administrator` is authenticated rather than `DOMAIN\Administrator`. Thus, you must specify the domain explicitly.

Source and target paths

- If there are multiple source arguments, then the target path must be a directory.
- To describe filepaths, use single-quote (' ') and forward-slashes (/) on all platforms.
- Avoid the following characters in filenames: `/ \ " : ' ? > < & * |`.

URI paths: URI paths are supported, but only with the following restrictions:

- If the source paths are URIs, they must all be in the same cloud storage account. No docroot (download), local docroot (upload), or source prefix can be specified.
- If a destination path is a URI, no docroot (upload) or local docroot (download) can be specified.
- The special schemes `stdio://` and `stdio-tar://` are supported only on the client. They cannot be used as an upload destination or download source.
- If required, specify the URI passphrase as part of the URI or set it as an environment variable (`ASPERA_SRC_PASS` or `ASPERA_DST_PASS`, depending on the direction of transfer).

UNC paths: If the server is Windows and the path on the server is a UNC path (a path that points to a shared directory or file on Windows operating systems) then it can be specified in an `ascp4` command using one of the following conventions:

1. UNC path that uses backslashes (\ )

If the client is a Windows computer, the UNC path can be used with no alteration. For example, `\\192.168.0.10\temp`. If the client is not a Windows computer, every backslash in the UNC path must be replaced with two backslashes. For example, `\\\\192.168.0.10\\temp`.

2. UNC path that uses forward slashes (/)

   Replace each backslash in the UNC path with a forward slash. For example, if the UNC path is `\\192.168.0.10\temp`, change it to `//192.168.0.10/temp`. This format can be used with any client operating system.

Environment Variables

If needed, you can set the following environment variables for use with an `ascp4` session.

ASPERA_SCP_PASS=*password*

Set the transfer user password.

ASPERA_SCP_COOKIE=*cookie*

Set the transfer user cookie.

ASPERA_SRC_PASS=*password*

Set the password to authenticate to a URI source.

ASPERA_DST_PASS=*password*

Set the password to authenticate to a URI destination.

Ascp4 Options

`-A, --version`

Display version and license information, then exit.

`--chunk-size=`*bytes*

Set the buffer size that is used for storage read/write operations and as an internal transmission and compression block. Valid range: 4 Kb - 128 Mb.

`--compare=`*method*

Set the *method* used to compare files when using `--overwrite` and `--resume`. *method* can be `size`, `size+mtime`, `md5`, `md5-sparse`, `sha1`, or `sha1-sparse`. If the `--overwrite` method is `diff` or `diff+older`, the default `--compare` method is `size`.

`--compression=`*method*

Compress file data inline. *method* can be: `none`, `zlib`, or `lz4`. Default: `lz4`. If set to `zlib`, `--compression-hint` can be used to set the compression level.

`--compression-hint=`*num*

Use when `--compression` is set to an that accepts compression level settings (currently only zlib). A lower value results in less, but faster, data compression (0 = no compression). A higher value results in greater, slower compression.Valid values are -1 to 9, where -1 is "balanced". Default: -1.

`--delete-after`, `--delete-after-transfer`

After all files are transferred, delete files that exist at the destination but not at the source. Objects on the destination that have the same name but different type or size as objects on the source are not deleted. Requires write permissions on the destination. Do not use with multiple sources, `--keepalive`, URI storage, or HTTP fallback.

Using `--delete-after` can be slower than `--delete-before` because the destination data set that is used to compare objects can be larger after the transfer.

`--delete-before`, `--delete-before-transfer`

Before transfer, delete files that exist at the destination but not at the source. Requires write permissions on the destination. Objects on the destination that have the same name but different type or size as objects on the source are not deleted. Do not use with multiple sources, `--keepalive`, URI storage, or HTTP fallback.

Using `--delete-before` can be faster than `--delete-after` because the destination data set that is used to compare objects can be smaller before the transfer occurs.

`-E` *pattern*

Exclude files or directories from the transfer based on the specified pattern. Use the `-N` option (include) to specify exceptions to `-E` rules. Up to 16 `-E` and `-N` rules can be specified. Rules are applied in the order in which they are encountered, from left to right. The following symbols can be used in the pattern:

- **`*`** (asterisk) represents zero or more characters in a string, for example `*.tmp` matches `.tmp` and `abcde.tmp`.
- **`?`** (question mark) represents a single character, for example `t?p` matches `tmp` but not `temp`.

For details and examples, see *Applying Filters to Include and Exclude Files*.

> **Note:** When filtering rules are found in `aspera.conf`, they are applied *before* rules given on the command line (`-E` and `-N`).

`--exclude-newer-than=`*`mtime`*
`--exclude-older-than=`*`mtime`*

Exclude files from the transfer based on when the file was last changed. This option does not apply to directories. Positive *mtime* values are compared to the source file system's "mtime" timestamp, which is usually seconds since 1970-01-01 00:00:00. Negative *mtime* values are applied as time before the present.

`--faspmgr-io`

Run `ascp4` in API mode using FASP manager I/O. `ascp4` reads FASPMGR4 commands from management and executes them. The FASPMGR4 commands are PUT/WRITE/STOP to open/write/close on a file on the server.

`--file-list=`*`filename`*

Transfer the content that is listed in *filepath*. The file list supports UTF-8 files and input from standard input through "-". If a directory does not exist at the destination, it is created (`-d` is automatically applied). Each source must be specified on a separate line, for example:

```
src
src2
...
srcN
```

Restrictions:

- Paths in file lists cannot use `user@host:filepath` syntax. You must use `--user` with `--file-list`.
- Only one `--file-list` option is allowed per `ascp` session. If multiple file lists are specified, all but the last are ignored.
- Only files from the file list are transferred, and any additional source files specified on the command line are ignored.

`-h, --help`

Display usage reference, then exit.

`--host=`*`host`*

Specify the host name or address of the server. Requires `--mode`. This option can be used instead of specifying the host as part of the filename (as *hostname*:*filepath*).

`-i` *private_key_file*

Use public key authentication and specify the private key file with a full or relative path. The private key file is typically in the directory `$HOME/.ssh/`. If multiple `-i` options are specified, only the last one is used.

`-k` *resume_level*

Enable the resumption of partially transferred files at the specified resume level. Default: 0. This option must be specified for your first transfer or it does not work for subsequent transfers. Resume levels:

- `-k 0`: Always retransfer the entire file (same as `--overwrite=always`).
- `-k 1`: Check file modification time and size and resume if they match (same as `--overwrite=diff --compare=size --resume`).
- `-k 2`: Check sparse checksum and resume if they match (same as `--overwrite=diff --compare=md5-sparse --resume`).
- `-k 3`: Check full checksum and resume if they match (same as `--overwrite=diff --compare=md5 --resume`).

`-L` *local_log_dir*[:*size*]

Log to the specified directory on the local host rather than the default directory. Optionally, set the size of the log file (default 10 MB).

`-l` *max_rate*

Set the target transfer rate. Default: 10 Mbps. This option accepts suffixes "G/g" for Giga, "M/m" for Mega, "K/k" for Kilo, and "P/p/%" for percentage, and decimals are allowed. If this option is not set by the client, the server target rate is used. If a rate cap is set in the local or server `aspera.conf`, then the rate does not exceed the cap.

`-m` *min_rate*

Set the minimum transfer rate in Kbps. Default: 0. If this option is not set by the client, then the server's *aspera.conf* setting is used. If a rate cap is set in the local or server *aspera.conf*, then the rate does not exceed the cap.

`--memory=`*bytes*

Set the maximum memory that the local `ascp4` process is allowed to use. Default: 256 MB. See also `--remote-memory`.

`--meta-threads=`*num*

Set the number of directory "creation" threads (receiver only). Default: 2.

`--mode=`*mode*

Set the transfer direction, where *mode* is `send` or `recv`. Requires `--host`.

`-N` *pattern*

Protect ("include") files or directories from exclusion by any `-E` (exclude) options that follow it. Files and directories are specified using *pattern*. Each option-plus-pattern is a *rule*. Up to 16 rules can be specified. Rules are applied in the order (left to right) in which they're encountered. Thus, `-N` rules protect files only from `-E` rules that follow them. Create patterns using standard globbing wildcards and special characters such as the following:

- **`*`** (asterisk) represents zero or more characters in a string, for example `*.tmp` matches `.tmp` and `abcde.tmp`.
- **`?`** (question mark) represents any single character, for example `t?p` matches `tmp` but not `temp`.

For details on specifying patterns and rules, including examples, see *Applying Filters to Include and Exclude Files*.

> **Note:** Filtering rules can also be specified in `aspera.conf`. Rules found in `aspera.conf` are applied *before* any `-E` and `-N` rules specified on the command line.

`--no-open`

In test mode, do not actually open or write the contents of destination files.

`--no-read`

In test mode, do not read the contents of source files.

`--no-write`

In test mode, do not write the contents of destination files.

`-O` *fasp_port*

Set the UDP port that is used for FASP transfers. Default: 33001.

`--overwrite=`*method*

Overwrite files at the destination with source files of the same name based on the *method*. Default: `always`. Use with `--compare` and `--resume`. *method* can be the following:

- `always` – Always overwrite the file.
- `never` – Never overwrite the file. If the destination contains partial files that are older or the same as the source files and `--resume` is enabled, the partial files resume transfer. Partial files with checksums or sizes that differ from the source files are not overwritten.
- `diff` – Overwrite if the file is different from the source, depending on the `compare` method (default is `size`). If the destination is object storage, `diff` has the same effect as `always`.

  If `resume` is not enabled, partial files are overwritten if they are different from the source, otherwise they are skipped. If `resume` is enabled, only partial files with different sizes or checksums from the source are overwritten; otherwise, files resume.
- `diff+older` – Overwrite if the destination is older and different from the source, depending on the `compare` method (default is `size`). If `resume` is not enabled, partial files are overwritten if they are older and different from the source, otherwise they are skipped. If `resume` is enabled, only partial files that are different and older than the source are overwritten, otherwise they are resumed.
- `older` – Overwrite if the destination timestamp is older than the source timestamp.

`-P` *ssh-port*

Set the TCP port that is used to initiate the FASP session. Default: 22.

`-p`

Preserve file timestamps for source modification time (`mtime`) and last access time (`atime`).

Important: On Windows, `mtime` and `atime` can be affected when the system automatically adjusts for Daylight Savings Time (DST). For details, see the Microsoft KB article, *http://support.microsoft.com/kb/129574*.

`--policy=`*xfer_policy*

Set the FASP transfer policy:

- `fixed` – Attempt to transfer at the specified target rate, regardless of network capacity. Content is transferred at a constant rate and the transfer finishes in a guaranteed time. It can occupy most of the network's bandwidth and is not recommended in most file transfer scenarios. This option requires a maximum (target) rate value (`-l`).
- `high` – Monitor the network and adjust the transfer rate to fully utilize the available bandwidth up to the maximum rate. When congestion occurs, the transfer rate is twice as fast as transfer with a fair policy. This option requires maximum (target) and minimum transfer rates (`-l` and `-m`).

- `fair` – Monitor the network and adjust the transfer rate to fully utilize the available bandwidth up to the maximum rate. When congestion occurs, bandwidth is shared fairly by transferring at an even rate. This option requires maximum (target) and minimum transfer rates (`-l` and `-m`).
- `low` – Similar to fair mode, the low policy uses the available bandwidth up to the maximum rate, but is less aggressive when FASP transfers share bandwidth with other network traffic. When congestion occurs, the transfer rate is reduced to the minimum rate until other traffic retreats.

If `--policy` is not set, `ascp4` uses the server-side policy setting (`fair` by default).

`--preserve-access-time`

Preserve the file timestamps (currently the same as `-p`).

`--preserve-creation-time`

Preserve the file timestamps (currently the same as `-p`).

`--preserve-file-owner-gid`
`--preserve-file-owner-uid`

(OS X and Linux/UNIX systems only.) Preserve the group information (`gid`) or owner information (`uid`) of the transferred files. The transfer user must be authenticated as a superuser to use these options.

`--preserve-modification-time`

Preserve the file timestamps (currently the same as `-p`).

`--preserve-source-access-time`

Preserve the file timestamps (currently the same as `-p`).

`-q`

Run `ascp4` in quiet mode. This option disables the progress display.

`-R` *remote_log_dir*

Log to the specified directory on the remote host rather than the default directory. **Note:** Client users that are restricted to aspshell are not allowed to use this option.

`--read-threads=`*num*

Set the number of storage "read" threads (sender only). Default: 2. To set "write" threads on the receiver, use `--write-threads`.

`--remote-memory=`*bytes*

Set the maximum memory that the remote `ascp4` process is allowed to use. Default: 256 MB.

`--resume`

Resume a transfer rather than retransferring the content if partial files are present at the destination and they do not differ from the source file based on the `--compare` method. If the source and destination files do not match, then the source file is retransferred. See `-k` for another way to enable resume.

`--scan-threads=`*num*

Set the number of directory "scan" threads (sender only). Default: 2.

`--sparse-file`

Enable `ascp4` to write sparse files to disk. This option prevents `ascp4` from writing zero content to disk for sparse files; `ascp4` writes a block to disk if even one bit is set in that block. If no bits are set in the block, `ascp4` does not write the block (`ascp4` blocks are 64 KB by default).

`--src-base=`*prefix*

Specify the prefix to be stripped from each source path. The remaining portion of the source path is kept intact at the destination. For usage examples, see *Ascp File Manipulation Examples*.

Use with URIs: The `--src-base` option performs a character-to-character match with the source path. For object storage source paths, the source base prefix must specify the URI in the same manner as the source paths. For example, if the source paths include an embedded passphrase, the source base prefix must also include the embedded passphrase or else it does not match.

`--symbolic-links=`*method*

Specify how to handle symbolic links. On Windows, the only option is `skip`. On other operating systems, this option takes following values. Default: `follow`.

- `follow` – Follow symbolic links and transfer the linked files.
- `copy` – Copy only the alias file. If a file with the same name exists on the destination, the symbolic link is not copied.
- `skip` – Skip symbolic links.

`-T`

Disable encryption for maximum throughput.

`-u` *user_string*

Define a user string for pre- and post-processing. This string is passed to the pre- and -post-processing scripts as the environment variable `$USERSTR`.

`--user=`*username*

Use the specified username to authenticate to the transfer server. This option can be used instead of specifying the username as part of the filepath (as *user@host:filepath*). If you do not specify a username for the transfer, the local username is authenticated by default.

> 👉 **Note:** If you are authenticating on a Windows machine as a domain user, the transfer server strips the domain from the username. For example, `Administrator` is authenticated rather than `DOMAIN\Administrator`. Thus, you must specify the domain explicitly.

`--worker-threads=`*num*

Set the number of worker threads for deleting files. On the receiver, each thread deletes one file or directory at a time. On the sender, each thread checks for the presences of one file or directory at a time. Default: 1.

`--write-threads=`*num*

Set the number of storage "write" threads (receiver only). Default: 2. To set "read" threads on the sender, use `--read-threads`.

For transfers to object or HDFS storage, write threads cannot exceed the maximum number of jobs that are configured for Trapd. Default: 15. To use more threads, open `/opt/aspera/etc/trapd/trap.properties` on the server and set `aspera.session.upload.max-jobs` to a number larger than the number of write threads. For example,

```
# Number of jobs allowed to run in parallel for uploads.
# Default is 15
aspera.session.upload.max-jobs=50
```

`-X` *rexmsg_size*

Set the maximum size, in bytes, of a retransmission request. Max: 1440.

`-Z` *dgram_size*

Set the datagram size (MTU). Range: 296 - 10000 bytes. The detected path MTU is used by default.

As of version 3.3, datagram size can be set on the server by using the `<datagram_size>` option in *aspera.conf*. The server setting overrides the client setting, unless the client is using a version of `ascp` that is older than 3.3, in which case the client setting is used. If the pre-3.3 client does not set `-Z`, then the datagram size is the discovered MTU and the server logs the message "LOG Peer client doesn't support alternative datagram size".

Built-in I/O Providers

Input/Output providers are library modules that abstract I/O scheme in `ascp4` architecture. `ascp4` has the following three built-in I/O providers:

- file (as a simple path or `file://path`)
- UDP (as `udp://233.3.3.3`)
- TCP (as `tcp://192.168.120.11`)

The default I/O scheme is file if there is no docroot and no scheme in the path. For examples of `ascp4` sessions that use UDP and TCP providers, see *Ascp4 Examples*.

File provider

The local disk can be specified for `ascp4` I/O by using a simple path or URL that starts with `file`. The following paths identify the same file (`/test/ascp4.log`) on the disk:

```
file:////test/ascp4.log
/test/ascp4.log
file://localhost:/test/ascp4.log
```

Similarly, the following URLs identify the same file (`test/ascp4.log`) on the disk:

```
file:///test/ascp4.log
test/ascp4.log
```

UDP provider

A UDP stream can be specified for `ascp4` I/O by using a URL that starts with `udp`. If the UDP stream is a multicast IP address, then `ascp4` connects to the multicast address. `ascp4` reads the UDP datagrams on the source and writes UDP datagrams on the destination. A UDP-provider filepath has the following format:

```
udp://ip_address:port[?option=value[&option=value]]
```

The UDP provider URL accepts the following options:

`pktbatch={0|1}` — Enable packet batching in read/write. Default: 1.
`maxsize=N` — Set the maximum stream length. Default: unlimited.
`maxtime=N` — Set the maximum stream duration, in seconds. Default: unlimited.
`maxidle=N` — Set the maximum idle duration, in seconds. Default: unlimited.
`rcvbufsz=N` — Set the receive buffer size. Default: 10 MB.
`sndbufsz=N` — Set the send buffer size. Default: 10 MB.
`ifaddr=ip_address` — Set the multicast interface. Default: 0.0.0.0.
`srcaddr=ip_address` — Set the multicast source for IGMPb3 source-specific multicast.
`ttl=N` — Set the multicast time-to-live. Default: 1.
`loopback=N` — Set the multicast loopback. Default: 1.
`dontfrag=N` — Prevent fragmentation of outgoing packets. Default: 0.

TCP provider

A TCP stream can be used for `ascp4` I/O by specifying a URL that starts with `tcp`. `ascp4` reads TCP data from the source and writes TCP data on the destination. Use the following format to specify a TCP provider on the source or destination:

```
tcp://ip_address:port[?option=value[&option=value]]
```

The TCP provider of the sender can also be specified with the following format:

```
tcp://:port[?option=value[&option=value]]
```

With this format, `ascp4` listens on the specified port up to a specified time (`maxidle`, see the following description of options for TCP provider URLs).

The TCP provider URL accepts the following options:

`port=`*N* — Set the network port number, default: 0.
`iosize=`*N* — Specify the read/write size, default: 32 KB.
`maxsize=`*N* — Set the maximum stream length, in bytes, no default.
`maxtime=`*N* — Set the maximum stream duration, in seconds, no default.
`maxidle=`*N* — Set the maximum idle duration, in seconds, default: 10 sec.
`rcvbufsz=`*N* — Set the receive buffer size, default: 4 MB.
`sndbufsz=`*N* — Set the send buffer size, default: 4 MB.
`ifaddr=`*ip_address* — Specify the TCP connection interface address.
`srcaddr=`*ip_address* — Specify the TCP connection source-specific address.

Ascp4 Examples

The commands for `ascp4` are generally similar to those for `ascp`, see *Ascp Command Reference* for examples and *Comparison of Ascp and Ascp4 Options* for option availability.

The following command examples demonstrate options that are unique to A4. These options enable reading management commands, enable read/write concurrency,and transfer TCP and UDP data streams.

- **Read FASP4 management commands**

  Read management commands V4 from management port 5000 and execute the management commands. The management commands version 4 are PUT, WRITE and CLOSE.

  ```
  # ascp4 -L /tmp/client-logs -R /tmp/server-logs --faspmgr-io -M 5000
   localhost:/tmp
  ```

- **Increase concurrency**

  The following command runs `ascp4` with two scan threads and eight read threads on the client, and eight meta threads and 16 write threads on the server.

  ```
  # ascp4 -L /tmp/logs -R /tmp/logs -l1g --scan-threads=2 --read-threads=8
   --write-threads=16 --meta-threads=8  /data/100K aspera@10.0.113.53:/data
  ```

- **Send a TCP stream**

  Read a TCP stream from 192.168.10.10 port 2000 and send it to 10.10.0.51. On 10.10.0.51, write the stream to localhost port 3000.

  ```
  # ascp4 -l 6000 -m 5000 --host=10.10.0.51 --mode=send --read-threads=1 --
  write-threads=1 tcp://192.168.10.10:2000 tcp://localhost:3000
  ```

- **Send a UDP data stream**

  Send a UDP stream multicasted on 233.3.3.3 port 3000 to host 192.168.0.11, then multicast the stream on 233.3.3.3 port 3001.

  ```
  # ascp4 -l 6000 -m 5000 --host=192.168.0.11 --mode=send --read-threads=1
   --write-threads=1
     udp://233.3.3.3:3000/?pktbatch=0 udp://233.3.3.3:3001/?loopback=1
  ```

Technical Support

Support Websites

For an overview of IBM Aspera Support services, go to *http://asperasoft.com/company/support/*.

To view product announcements, webinars, and knowledgebase articles, as well as access the Aspera Support Community Forum, sign into the IBM Aspera Support site at *support.asperasoft.com* using your email address (not your company Aspera credentials), or set up a new account. You can click on a heading then click **Follow** to receive notifications when new knowledgebase articles are available; if you follow **RELEASE NOTES** under a specific product, you will be automatically notified of new releases.

Personalized Support

You may contact an Aspera support technician 24 hours a day, 7 days a week, through the following methods, with a guaranteed 4-hour response time.

If you have an emergency, create a ticket using the **Support Request Form** with as many details as you have available and then **call**. If you are asked to leave a voice message, include the ticket number.

Email	support@asperasoft.com
Phone (North America)	+1 (510) 849-2386, option 2
Phone (Europe)	+44 (0) 207-993-6653 option 2
Phone (Singapore)	+81 (0) 3-4578-9357 option 2
Support Request Form	*https://support.asperasoft.com/anonymous_requests/new/*

Legal Notice