IBM Aspera Streaming User Guide 3.9.6

Windows

Revision:120 Generated:02/04/2020 19:54

Contents

Introduction

IBM Aspera Streaming (Streaming) uses Aspera FASPStream technology to send a unicast or multicast video stream over the internet using `ascp4`. Streams are reliably and bit-perfect replicated at the receiving endpoint with minimal and predictable latency, with rates from 10s of Mbps to multiple Gbps, and follow the standard Aspera FASP security framework. The most common example of an input stream is media, which is encoded as a 'transport stream' (often referred to as MPEG-TS).

IBM Aspera Streaming generally transfers a video stream from a *stream provider*, a system external to Streaming that produces a video stream, and a *stream consumer*, a system external to Streaming that receives a video stream.

License Requirements

Streaming can run on either a standalone computer or as an embedded service on third-party devices (for example, a video encoder). To run a Streaming session between two systems, both systems must have a Streaming license. Three license types are currently offered: *sender*, *receiver*, and *server*. In a single point-to-point transfer, the *sender* is the system that initiates the transfer and sends a request to a *receiver* for a streaming session. In general, a *sender* may only produce a stream and a *receiver* may only consume a stream. Additionally, a *server* may act as both a *sender* and a *receiver* and is not limited to a number of concurrent streams up to the bandwidth limit purchased. Additional information about Streaming licensing is in the product License Information document.

Installation

IBM Aspera Streaming is installed as a specially-licensed IBM Aspera High-Speed Transfer Endpoint (formerly Aspera Point-to-Point Client) with limited GUI capabilities.

Important: If this is a product upgrade, review all prerequisites described in "Before Upgrading or Downgrading" in the IBM Aspera High-Speed Transfer Endpoint Admin Guide.

1. Download the installer from the Aspera website.

   Use the credentials provided to your organization by Aspera to access:

   http://downloads.asperasoft.com/en/downloads/60

   If you need help determining your firm's access credentials, contact your Aspera account manager.
2. Install the product.
3. Install the license.

   Create the following file:

   ```
   //aspera/etc/aspera-license
   ```

   Copy and paste your license key string into it, then save and close the file. To verify the license information, run the following command:

   ```
   $ ascp4 -A
   ```
4. Prepare your computer as a Streaming sender (client) or receiver (server).

   Download or open the IBM Aspera High-Speed Transfer Endpoint Admin Guide for your OS and use it as a reference for the following preparation steps.

   If your computer is a Streaming receiver (server):

   a) Configure the firewall (see "Configuring Your Firewall").
   b) Change and secure the TCP port (see "Securing Your SSH Server").

c) Add a user and configure their access.

The Streaming sender authenticates to the receiver by using operating system accounts on the receiver. For example, if the sender user, "marketing_mgr" wants to send a stream to the receiver server, add marketing_mgr as a system user on the receiver. To secure your receiver, restrict marketing_mgr's access to only certain directories on the server (set a docroot), set transfer permissions, and set the default shell as aspshell. For instructions, see "Setting Up Users" to set up a user in the GUI, or "Setting Up Transfer Users (Terminal)" to set up a user from the command line.

Note: Multicast-to-multicast transfers must be authenticated by a transfer user who does not have a docroot configured in `aspera.conf`. For instructions, see Streaming for Video Command Syntax on page 5.

d) Set up the user's public key on the receiver.

Aspera recommends using SSH key authentication. Have the user on the sender create an SSH private-public key pair and send the public key to the admin of the receiver. Then follow the instructions in "Setting Up a User's Public Key on the Server."

e) If you are sending or receiving a multicast video stream with a macOS server, and do not want to use the default interface (en0), configure the new interface.

For instructions, see Configuring macOS Server for Multicast Streams on page 4.

f) For a complete overview of Aspera-recommended security settings, see "Aspera Ecosystem Security Best Practices".

If your computer is a Streaming sender (client):

a) Configure the firewall (see "Configuring Your Firewall").

b) If you need to authenticate to the receiver with an SSH key, create an SSH key and send the public key to the receiver admin.

For instructions on creating an SSH key, see "Creating SSH Keys in the GUI" or "Creating SSH Keys (Command Line)."

To uninstall Streaming, follow the instructions in "Uninstalling" in the IBM Aspera High-Speed Transfer Endpoint Admin Guide.

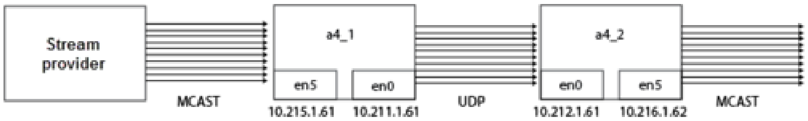
Configuring macOS Server for Multicast Streams

If you are sending or receiving multicast streams from a macOS server, multicast fails if the receiving or sending interface is not the macOS default interface. If no default gateway is defined on a macOS server, the default interface is **en0**. To use a different interface than the default, change the default interface for your server

Run the following commands:

```
> route delete default new_interface_ip
> route add default new_interface_ip
```

Example



Server Name	en0 IP Address	en5 IP Address
a4_1	10.211.1.61	10.215.1.61
a4_2	10.212.1.62	10.216.1.62

In this example, server **a4_1** acts as the multicast receiver and sends the stream over UDP to server **a4_2**. Server **a4_2** then broadcasts the multicast streams to waiting receivers. Since both servers are using a non-default interface (not

en0) to receive and send the multicast streams, we must set the interface receiving and sending the multicast streams as the server default interface.

Run the following command on **a4_1**:

```
> route delete default 10.215.0.1
> route add default 10.215.0.1
```

Run the following command on **a4_2**:

```
> route delete default 10.216.0.1
> route add default 10.216.0.1
```

Note: The interface IP addresses `10.215.0.1` and `10.216.0.1` routes are the primary gateways and route all `10.215.0.*` and `10.216.0.*` traffic, respectively.

Streaming for Video Command Syntax

Streaming for Video is used to transport MPEG transport streams (TS) across long distances. On the source host, the UDP provider captures packets delivered to a local multicast or unicast UDP port. The stream is transported reliably and in order to the remote host, where it is delivered to the specified local multicast or unicast UDP port. Video streaming that uses TCP and file I/O is also supported.

Streaming for Video uses the `ascp4` command line tool and many `ascp4` options are supported by Streaming for Video. For more information, see Ascp 4 Command Reference on page 7.

Required Configuration for Multicast-to-Multicast Transfers

The transfer user who authenticates the multicast-to-multicast video stream transfer must have no docroot configured in `aspera.conf`. A file restriction can be set instead to restrict access.

Run the following command to unset a docroot and set a file restriction:

```
> asconfigurator -x
 "set_user_data;user_name,username;absolute,AS_NULL;file_restriction,|restriction"
```

The restriction can be set to allow all access (*) or limited by protocol, hostname or path:

Restriction	Format Example
By protocol	udp://* tcp://*
By protocol and hostname	udp://hostname*
By protocol, hostname, and port	tcp://hostname:5000*

General Command Line Usage

```
> ascp4 -m minimum_rate -l target_rate --mode=mode --host=remote_hostname
 --compression=none --user=username --read-threads=1 --write-
threads=1 input_uri output_uri
```

- `ascp4` streaming supports two transfer directions: `send` and `recv`.
- The `ascp4` command defaults to multiple threads, but for reliable and in-order transport of streams you must use only one read and write thread by specifying `--read-threads=1 --write-threads=1`.

- The video stream source and destination can be `udp://`, `tcp://`, or `file://`. For more information, see Built-in I/O Providers on page 14
- For command line examples, see Ascp 4 Video Streaming Examples on page 15.

Recommended Rate Settings for Video Streams

ascp4 Option	Description	Recommendation
`-m`	Minimum rate	Take the encoding rate of the transport stream and add 1 Mbps.
`-l`	Target rate	Take the minimum rate and add 10% of the minimum rate.

For example, if the encoding rate is 10 Mbps, use the following settings:

```
> ascp4 -m 11M -l 13M ...
```

Multicast URI Syntax

The input multicast URI and the output multicast URI uses the same syntax.

```
multicast_protocol_scheme://stream_ip_address:port?option=value&option=value...
```

The multicast protocol scheme can be either `udp` or `mcast`. If the IP address of your video stream is a multicast address, `ascp4` uses multicast regardless of the protocol scheme (in other words, both `udp` and `mcast` use multicast). In order to use unicast addresses, you must use the `udp` scheme.

You can configure properties of the stream by adding options to the URI after the question mark (?), each separated by an ampersand (&). The following table describes the supported options.

Option	Description	Default
`pktbatch={1|0}`	How to handle packet read and write. If 1, batch read and write UDP datagrams. If 0, read and write one packet at a time.	1
`maxsize=`*`maximum_size`*	Maximum stream length	No default
`maxtime=`*`maximum_time`*	Maximum stream duration, in seconds	No default
`maxidle=`*`maximum_time`*	Maximum idle duration, in seconds	No default
`rcvbufsz=`*`buffer_size`*	Receive buffer size	10MB
`sndbufsz=`*`buffer_size`*	Send buffer size	10MB
`ifaddr=`*`ip_address`*	Multicast interface IP address	0.0.0.0
`srcaddr=`*`ip_address`*	Multicast source IP address	0.0.0.0
`ttl=`*`hops`*	Multicast time-to-live	1
`loopback=`*`boolean`*	Multicast loopback	1

Ascp 4 Command Reference

Supported environment variables, the general syntax, and command options for `ascp4` 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.

ascp4 Syntax

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

User

The username of the Aspera transfer user can be specified as part of the as part of the source or destination, whichever is the remote server 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 destination paths

- If there are multiple source arguments, then the target path must be a directory.
- To describe filepaths, use single quotes (' ') and forward slashes (/) on all platforms.
- To transfer to the transfer user's docroot, specify `"."` as the destination.
- Avoid the following characters in filenames: `/ \ " : ' ? > < & * |`.
- If the destination is a symbolic link, then the file is written to the target of the symbolic link. However, if the symbolic link path is a relative path to a file (not a directory) and a partial file name suffix is configured on the receiver, then the destination path is relative to the user's home directory. Files within directories that are sent to symbolic links that use relative paths are not affected.

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 side. 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 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 machine, 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-side operating system.

Environment Variables

If needed, you can set the following environment variables for use with an `ascp4` session. The total size for environment variables depends on your operating system and transfer session. Aspera recommends that each environment variable value should not exceed 4096 characters.

`ASPERA_SCP_PASS=`*password*

The password that is used for SSH authentication of the transfer user.

`ASPERA_SCP_TOKEN=`*token*

Set the transfer user authorization token. Ascp 4 currently supports transfer tokens, which must be created by using `astokengen` with the `--full-paths` option. For more information, see .

`ASPERA_SCP_COOKIE=`*cookie*

A cookie string that is passed to monitoring services.

`ASPERA_SRC_PASS=`*password*

The password that is used to authenticate to a URI source.

`ASPERA_DST_PASS=`*password*

Set the password that is used to authenticate to a URI destination.

Ascp 4 Options

`-A, --version`

Display version and license information.

`-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 .

`--chunk-size=`*bytes*

Perform storage read/write operations with the specified buffer size. Also use the buffer size as an internal transmission and compression block. Valid range: 4 KB - 128 MB. For transfers with object storage, use `--chunk-size=1048576` if chunk size is not configured on the server to ensure that the chunk size of `ascp4` and Trapd match.

`--compare={size|size+mtime|md5|md5-sparse|sha1|sha1-sparse}`*method*

When using `--overwrite` and `--resume`, compare files with the specified method. If the `--overwrite` method is `diff` or `diff+older`, the default `--compare` method is `size`.

`--compression={none|zlib|lz4}`

Compress file data inline. Default: `lz4`. If set to `zlib`, `--compression-hint` can be used to set the compression level.

`--compression-hint=`*num*

Compress file data to the specified level when `--compression` is set to an option 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.

`-D | -DD | -DDD`

Log at the specified debug level. With each `D`, an additional level of debugging information is written to the log. This option is not supported if the transfer user is restricted to aspshell.

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

Before transfer, delete files that exist at the destination but not at the source. The source and destination arguments must be directories that have matching names. 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 or `--keepalive`.

-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. 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`.

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 (but not directories) from the transfer based on when the file was last changed. 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.

--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=*filepath*

Transfer the files and directories that are listed in *filepath*. Only the files and directories are transferred; path information is not preserved at the destination. Each source must be specified on a separate line, for example:

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

To read a file list from standard input, use "-" in place of *filepath* (as `ascp4 --file-list=- ...`) . UTF-8 file format is supported. Use with `-d` if the destination folder does not exist.

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 `ascp4` session. If multiple file lists are specified, all but the last are ignored.
- Only files and directories from the file list are transferred, and any additional source files or directories specified on the command line are ignored.
- If more than one read thread is specified (default is 2) for a transfer that uses `--file-list`, the files in the file list must be unique. Duplicates can produce unexpected results on the destination.
- Because multiple sources are being transferred, the destination must be a directory.
- If the source paths are URIs, the size of the file list cannot exceed 24 KB.

For very large file lists (~100 MB+), use with `--memory` to increase available buffer space.

`-h, --help`

Display the usage summary.

`--host=`*host*

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

`-i` *private_key_file*

Authenticate the transfer using public key authentication with the specified SSH private key file (specified 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 {0|1|2|3}`

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

`--keepalive`

Enable `ascp4` to run in persistent mode. This option enables a persistent session that does not require that source content and its destination are specified at execution. Instead, the persistent session reads source and destination paths through mgmt commands. Requires `--mode` and `--host`.

`-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).

`-l` *max_rate*

Transfer at rates up to the specified target 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. 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*

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.

`--memory=`*bytes*

Allow the local `ascp4` process to use no more than the specified memory. Default: 256 MB. See also `--remote-memory`.

`--meta-threads=`*num*

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

`--mode={send|recv}`

Transfer in the specified direction: `send` or `recv` (receive). 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*. 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`.

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***

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

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

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

`-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
```

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

Transfer according to the specified 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. This option requires a maximum (target) rate value (`-l`).
- `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 transfer with a fair policy. This option requires 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. This option requires 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, `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`

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

`--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*

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

Note: If more than one read thread is specified for a transfer that uses `--file-list`, the files in the file list must be unique. Duplicates can produce unexpected results on the destination.

`--remote-memory=`*bytes*

Allow the remote `ascp4` process to use no more than the specified memory. Default: 256 MB.

`--resume`

Resume a transfer rather than re-transferring 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 re-transferred. See `-k` for another way to enable resume.

`--scan-threads=`*num*

Use the specified 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*

Strip the specified prefix from each source path. The remaining portion of the source path is kept intact at the destination. Available only in send mode.

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.

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

Handle symbolic links using the specified method. On Windows, the only option is `skip`. On other operating systems, this option takes following values:

- `follow` – Follow symbolic links and transfer the linked files. (Default)
- `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. 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 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. 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`. Thus, you must specify the domain explicitly.

`--worker-threads=`*num*

Use the specified 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*

Use the specified 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***

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 version 3.3, 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".

Built-in I/O Providers

Input/Output providers are library modules that abstract I/O scheme in Ascp 4 architecture. Ascp 4 has the following built-in I/O providers:

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

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
```

TCP provider

A TCP video 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.

UDP provider

A UDP video 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.

Ascp 4 Video Streaming Examples

Use the following examples as a guide for creating your own streaming transfers with Ascp 4.

- Send a multicast stream:

```
> ascp4 --mode=send --host=desthost --compression=none --read-
threads=1 --write-threads=1 udp://233.3.3.3:3000?loopback=1&ttl=2
 udp://233.4.4.4:3000?loopback=1&ttl=2
```

- Capture a local multicast stream and send it to the receiver as a UDP unicast stream:

```
> ascp4 --mode=send --host=desthost --compression=none --read-threads=1
 --write-threads=1 udp://233.3.3.3:3000?loopback=1&ttl=2 udp://
localhost:3000/
```

- 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 multicast UDP stream 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
```

- Multicast using the same multicast IP address and varying the multicast port.

```
> ascp4 -L/opt/test-local-01 -R/opt/test-remote-01 -DD -m 12m -l
 15m --mode send --host 10.132.117.2 --user root --read-threads
 1 --write-threads 1 --compression none "udp://233.33.3.1:3001?
sndbufsz=100MB&ifaddr=10.131.117.1" "udp://233.44.4.1:4001?
rcvbufsz=100M&loopback=0"
> ascp4 -L/opt/test-local-02 -R/opt/test-remote-02 -DD -m 12m -l
 15m --mode send --host 10.132.117.2 --user root --read-threads
 1 --write-threads 1 --compression none "udp://233.33.3.1:3002?
sndbufsz=100MB&ifaddr=10.131.117.1" "udp://233.44.4.1:4002?
rcvbufsz=100M&loopback=0"
> ascp4 -L/opt/test-local-03 -R/opt/test-remote-03 -DD -m 12m -l
 15m --mode send --host 10.132.117.2 --user root --read-threads
 1 --write-threads 1 --compression none "udp://233.33.3.1:3003?
sndbufsz=100MB&ifaddr=10.131.117.1" "udp://233.44.4.1:4003?
rcvbufsz=100M&loopback=0"
> ascp4 -L/opt/test-local-04 -R/opt/test-remote-04 -DD -m 12m -l
 15m --mode send --host 10.132.117.2 --user root --read-threads
 1 --write-threads 1 --compression none "udp://233.33.3.1:3004?
sndbufsz=100MB&ifaddr=10.131.117.1" "udp://233.44.4.1:4004?
rcvbufsz=100M&loopback=0"
```

- Multicast using the same multicast port and varying the multicast IP address:

```
> ascp4 -L/opt/test-local-01 -R/opt/test-remote-01 -DD -m 12m -l
 15m --mode send --host 10.132.117.2 --user root --read-threads
 1 --write-threads 1 --compression none "udp://233.33.3.1:3001?
sndbufsz=100MB&ifaddr=10.131.117.1" "udp://233.44.4.1:4001?
rcvbufsz=100M&loopback=0"
> ascp4 -L/opt/test-local-02 -R/opt/test-remote-02 -DD -m 12m -l
 15m --mode send --host 10.132.117.2 --user root --read-threads
 1 --write-threads 1 --compression none "udp://233.33.3.2:3001?
sndbufsz=100MB&ifaddr=10.131.117.1" "udp://233.44.4.2:4001?
rcvbufsz=100M&loopback=0"
> ascp4 -L/opt/test-local-03 -R/opt/test-remote-03 -DD -m 12m -l
 15m --mode send --host 10.132.117.2 --user root --read-threads
 1 --write-threads 1 --compression none "udp://233.33.3.3:3001?
sndbufsz=100MB&ifaddr=10.131.117.1" "udp://233.44.4.3:4001?
rcvbufsz=100M&loopback=0"
> ascp4 -L/opt/test-local-04 -R/opt/test-remote-04 -DD -m 12m -l
 15m --mode send --host 10.132.117.2 --user root --read-threads
 1 --write-threads 1 --compression none "udp://233.33.3.4:3001?
sndbufsz=100MB&ifaddr=10.131.117.1" "udp://233.44.4.4:4001?
rcvbufsz=100M&loopback=0"
```

Testing a Video Stream

This article describes how to set up an IBM Aspera Streaming connection between two computers, start a multicast with an encoded transport stream, test that the receiving computer is receiving packets, and play the streamed media content.

The following instructions requires two computers installed with Streaming:

- computer A: Linux computer with the built-in sender license.
- computer B: Linux computer with a receiver license.

The following steps must be performed on computer A:

1. Start `ascp4` to transport streams when an input is available.
   The following example assumes you have SSH key access to computer B from computer A.

   ```
   > ascp4 --mode=send --user=computerB_user -i ~/.ssh/id_rsa --
   host=computerB  --compression=none --read-threads=1 --write-threads=1
    udp://233.3.3.3:3000?loopback=1&ttl=2 udp://233.4.4.4:4000?
   loopback=1&ttl=2
   ```

   Note: For more information about the command syntax for starting a FASPStream, see Streaming for Video Command Syntax on page 5.
2. Download a test file to stream.

   In your browser, download the `ed24p_00.zip` test file from the www.w6rz.net website (a community transport stream testing website):

   http://www.w6rz.net/ed24p_00.zip

   Extract the contents into an easily accessible folder. You may provide your own media file, but the examples in this documentation assume that you are using the `ed24p_00.zip` transport stream file located at `/temp/ed24p_00.ts`.
3. Provide a multicast stream of a test file to `ascp4`.
   Run the `ffmpeg` command with the location of the media file and set the URI of the resulting stream:

   ```
   > ffmpeg -re  -i /temp/ed24p_00.ts -vcodec copy -acodec copy -f mpegts
    "udp://233.3.3.3:3000?ttl=2&pkt_size=1316"
   ```
4. Check to see the output of `ascp4` to make sure the Rate of transfer is going up to the expected speed.

Now that your stream is running and `ascp4` shows that it is transporting the stream, check the receiver is receiving the media file. The following steps must be performed on computer B:

5. Run the `tcpdump` command to check streams are coming.

   If you do not have `tcpdump` on your computer, download the third-party, open-source WinDump from the WinDump website:

   http://www.winpcap.org/windump/

   The port number corresponds to port configured in the destination multicast URI. In the example below, the destination port was configured as `4000`.

   ```
   > tcpdump upd and port 4000
   ```
6. Play the media file over the stream.

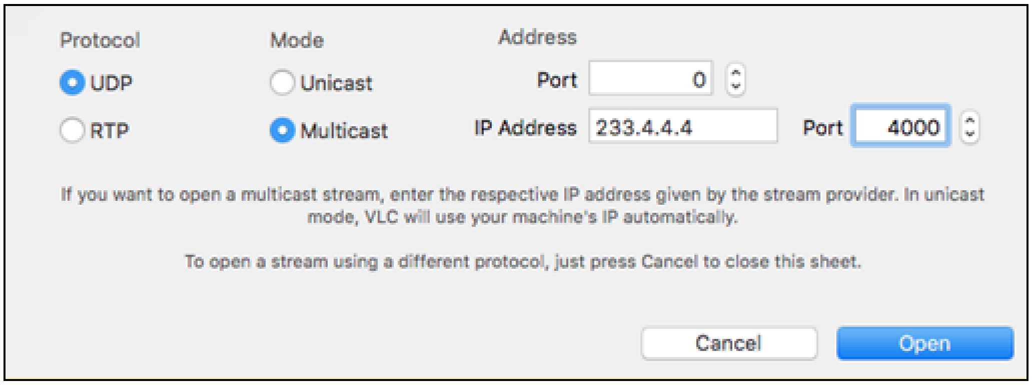
   The following example uses the third-party, open-source `vlc` command. If you do not have `vlc` on your computer, download it from:

   http://www.videolan.org/vlc/download-windows.html

   Using VLC, play the media from the stream.

   a) Open VLC.
   b) Click **Open media**. In the resulting dialog, go to the **Network** tab and click **Open RTP/UDP Stream**.
   c) Configure the settings according to your multicast URI.

Option	Value
Protocol	UDP
Mode	Multicast
IP Address	233.4.4.4
Port (for the IP Address)	4000



Protocol

● UDP

○ RTP

Mode

○ Unicast

● Multicast

Address

Port 0

IP Address 233.4.4.4 Port 4000

If you want to open a multicast stream, enter the respective IP address given by the stream provider. In unicast mode, VLC will use your machine's IP automatically.

To open a stream using a different protocol, just press Cancel to close this sheet.

Your media file should now be playing in the VLC media player.

Troubleshooting Stream Transfers

Multicast Transfer Fails with "Error: Empty file list from file/stdin"

This error might indicate that the transfer user has a docroot set in `aspera.conf`, which is not supported for multicast-to-multicast transfers.

The transfer user who authenticates the multicast-to-multicast video stream transfer must have no docroot configured in `aspera.conf`. A file restriction can be set instead to restrict access.

Run the following command to unset a docroot and set a file restriction:

```
> asconfigurator -x
 "set_user_data;user_name,username;absolute,AS_NULL;file_restriction,|restriction"
```

The restriction can be set to allow all access (*) or limited by protocol, hostname or path:

Restriction	Format Example
By protocol	udp://* tcp://*
By protocol and hostname	udp://hostname*
By protocol, hostname, and port	tcp://hostname:5000*

Transfer from macOS Fails

To diagnose the problem, re-run the stream transfer with the logging level set to debug by adding `-DD` to the command. After the transfer fails, open the log file (*homedir*`/Library/Logs/Aspera/`) and search for an `ERR` response.

`ERR udp_io_open: failed to set rcvbugsz=10485760 (e=55)(ENOBUFS)`

This error indicates that the socket buffer size on the Mac computer is too small to send or receive UDP packets. To increase the socket buffer size, specify a large buffer size by adding the following parameters to the URL:

```
url/?rcvbufsz=4000000&sndbufsz=4000000
```

With these parameters, a transfer to and from a Mac computer is written similar to the following example:

```
> ascp4 -DD -m 12m -l 15m --mode send --host 10.13.117.12 --
user root --read-threads 1 --write-threads 1 --compression none
```

```
  "udp://233.13.13.2:3002/?rcvbufsz=4000000&sndbufsz=4000000"
  "udp://233.14.14.2:4002/?rcvbufsz=4000000&sndbufsz=4000000"
```

Technical Support

Support Websites

For an overview of IBM Aspera Support services, visit https://www.ibm.com/products/aspera/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 https://www.ibm.com/mysupport/ using your IBMid (not your company Aspera credentials), or set up a new account.

Technical Support

You may contact Aspera support using the IBM Aspera Support Guide: https://www.ibm.com/support/home/pages/support-guide/?product=3712142

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.

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

Legal Notice

Aspera, the Aspera logo, and FASP transfer technology are trademarks of Aspera, Inc., registered in the United States. Aspera Drive, IBM Aspera High-Speed Transfer Server (a merger of IBM products formerly named Aspera Connect Server and Aspera Enterprise Server, 2008 and 2007), IBM Aspera High-Speed Endpoint (formerly Aspera Point-to-Point, 2006), IBM Aspera Desktop Client (formerly Aspera Client, 2005), Aspera Connect, Aspera Cargo, Aspera Console, Aspera Orchestrator, Aspera Crypt, Aspera Shares, the Aspera Add-in for Microsoft Outlook, Aspera FASPStream, and Aspera Faspex are trademarks of Aspera, Inc. All other trademarks mentioned in this document are the property of their respective owners. Mention of third-party products in this document is for informational purposes only. All understandings, agreements, or warranties, if any, take place directly between the vendors and the prospective users.