Aspera Enterprise Server 2.6

Isilon IQ OneFS 6

Document Revision: 5

Contents

Introduction

Aspera Enterprise Server for Isilon is a web-based file transfer server built upon Aspera's *fasp* transport. It offers the following features:

Feature	Description
fasp transport technology	File transfer protocol that dramatically speeds transfers over IP networks by eliminating the fundamental bottlenecks in conventional technologies. *fasp* features bandwidth control, resume, transfer encryption, content protection, and data integrity validation.
Transfer server	Allows an unlimited number of concurrent client transfers. Uses virtual links to manage aggregate bandwidth usage.
Web UI	A web-based interface that enables transfers for Aspera Connect clients. Includes the HTTP Fallback Server to allow clients without *fasp* connectivity to transfer using HTTP or HTTPS.
Aspera Sync	A command-line synchronization program.
Database Logger	A MySQL adapter that logs the server's transfer activity to a database.
Pre- and Post-Processing (Prepost)	Executes customizable actions when transfer events - start and end of sessions and files - occur. *An email notification script is included.*
ascp command	The command-line file transfer program.

Installation

Install the Aspera transfer product and set up your computer for *fasp* file transfers.

Requirements

Software and hardware requirements for optimal product functionality

System requirements for Aspera Enterprise Server:

- Isilon OneFS version 6 or higher.
- For Database Logging - A MySQL Database.

Clients can access Web UI with the following web browsers:

Supported OS	Supported Browsers
Windows XP (SP2 and above), 2003, 2008, Vista, 7	Internet Explorer 7+, Firefox 3+
Mac OS X 10.4, 10.5, 10.6	Safari 3+, Firefox 3+
Linux 32/64-bit	Firefox 3+

Upgrade Existing Installation

Upgrade your existing Enterprise Server installation.

If you have a running Enterprise Server on your Isilon, and would like to upgrade it to the latest version, follow these steps:

1. Backup Aspera configuration files

   Back up these files:

File type	Path
All configurations	/ifs/.ifsvar/aspera/etc/
Customized Aspera Web	/ifs/.ifsvar/aspera/www/
Pre- and Post-Processing	• /usr/local/aspera/var/aspera-prepost • Custom scripts refer to the previous file.

2. Execute the installer

Execute the command to set up Aspera Enterprise Server on all Isilon clusters:

```
$ isi_for_array -s -q sh /usr/local/aspera/var/install.sh
```

3. Install the license

   To install the license, create the following file and paste your license key string into it:

   ```
   /ifs/.ifsvar/aspera/etc/aspera-license
   ```

   When finished, save and close the file. Use this command to verify the license info:

   ```
   $ ascp -A
   ```

   If you are updating your product license after the installation, refer to *Updating Product License* on page 74.

4. Convert the old aspera.conf file manually (Optional)

   The new application uses new file format in the configuration file aspera.conf, and moved the document root settings from the file docroot to aspera.conf. The installer converts your old configuration files to the new format, using a "strict" method. If the old aspera.conf file has errors or unrecognized directives, the conversion will fail.

   To review the errors, execute a manual strict conversion. Change the aspera.conf's path if it is not in the default location:

   ```
   $ cd /ifs/.ifsvar/aspera/etc/
   $ sudo asconfigurator -T -F convert_conf_V1_data  ./aspera.conf
   ```

Install with Pre-loaded Installer

Install Enterprise Server on Isilon OneFS 5.5.4 or higher, using pre-loaded installer.

The Aspera Enterprise Server installer is pre-loaded on Isilon 5.5.4 or higher. To install from this pre-loaded installer, log into your Isilon server as an administrator and follow these steps:

1. Execute the installer

   Execute the command to set up Aspera Enterprise Server on all Isilon clusters:

   ```
   $ isi_for_array -s -q sh /usr/local/aspera/var/install.sh
   ```

2. Install the license

   To install the license, create the following file and paste your license key string into it:

   ```
   /ifs/.ifsvar/aspera/etc/aspera-license
   ```

When finished, save and close the file. Use this command to verify the license info:

```
$ ascp -A
```

If you are updating your product license after the installation, refer to *Updating Product License* on page 74.

At this point, the Aspera transfer product is installed, but additional configuration steps are required to set up the application. Continue to the following topics in this chapter.

Configuring the Firewall

Firewall settings required by the product.

The Aspera transfer products require access through the ports listed in the table below. If you cannot establish the connection, review your local corporate firewall settings and remove the port restrictions accordingly:

Product	Firewall Configuration
Isilon server	• Allow inbound and outbound connections for SSH. (TCP/22) • Allow inbound and outbound connections for *fasp* transfers. (UDP/33001+, see notice below) • For the HTTP Fallback Server, allow inbound and outbound for HTTP or HTTPS. (TCP/8080, TCP/8443) • For Web UI, allow inbound connection for HTTP or HTTPS web access. (TCP/80, TCP/443) • To add it as a Aspera Console's managed node, allow inbound connection for Aspera Central. (TCP/40001)
Client	• Allow outbound connection for SSH. (TCP/22) • Allow outbound connection for *fasp* transfers. (All UDP ports)

When multiple clients are connecting from the same IP address (e.g. subnet) using different user accounts, an equivalent number of UDP ports should be opened, starting incrementally from UDP/33001. For example, to allow 10 concurrent incoming connections from the same IP address, open ports from UDP/33001 to UDP/33010.

Testing Transfer

Test client functionality by transferring with the Aspera Demo Server.

To make sure that the software is working properly, follow these steps to test download and upload transfers between your system and the Aspera Demo Server (Demo Server):

1. Download test files from the Demo Server

The first test is to download a test file from the Demo Server. The transfer command is based on the following settings:

Item	Value
Demo Server address	demo.asperasoft.com
Login account	aspera
password	demoaspera
Test file	/aspera-test-dir-large/100MB
Download location	/tmp/
Transfer settings	Fair transfer policy, target rate 10M, minimum rate 1M, encryption disabled.

Use the following command to download, press y to accept the server's key, and enter the password *demoaspera* when prompted:

```
$ ascp -QT -l 10M -m 1M aspera@demo.asperasoft.com:aspera-test-dir-large/100MB /tmp/
```

You should see the following session messages. The description from left to right is explained below:

Item	Description
100MB	The name of the file that is being transferred.
23%	The percentage completed.
23MB	The amount transferred.
509Kb/s	Current transfer rate.
11:59 ETA	Estimated time remaining.

2. Upload test files to the Demo Server

When the file is downloaded, try uploading the same file back to the Demo Server. Use the command to upload the file (100MB) to the Demo Server's */Upload* directory. Enter the password *demoaspera* when prompted:

```
$ ascp -QT -l 10M -m 1M /tmp/100MB aspera@demo.asperasoft.com:Upload/
```

Testing Remotely-Initiated Transfer

Test functionality.

To test *fasp* transfers initiated from another computer, prepare a client machine and follow these instructions:

1. *Client Machine* - Verify the connectivity to the **Enterprise Server**

   On the client machine, bring up the Command Prompt (Window) or Terminal, and use the **ping** command to verify the connectivity to the host. In this example, the address of the *Enterprise Server* is 10.0.0.2:

   ```
   $ ping 10.0.0.2
   ```

2. *Client Machine* - Initiate a transfer to your **Enterprise Server**

   To verify the *fasp* Transfer Server functionality, and the transfer users are created correctly, try to establish a connection from a client machine with it. On the client machine, bring up a Command Prompt (Window) or a Terminal.

   In this example, the *Enterprise Server* computer has the following settings:

Item	Value
Host Address	10.0.0.2
Login	asp1
Files to upload	/client-dir/files
Destination Folder	(user's docroot)/dir
Transfer Options	maximum rate 10 Mbps (-l 10000), minimum at 1 Mbps (-m 1000), without encryption

   Execute the following command on the client machine:

   ```
   $ ascp -TQ -l 10000 -m 1000 /client-dir/files asp1@10.0.0.2:/dir
   ```

If you are having difficulties establishing *fasp* transfers in your environment, refer to *Clients Can't Establish Connection* on page 79.

Setting Up Web UI

Set up your computer's web server to host the **Aspera Enterprise Server** Web UI.

The Web UI is a web-based file server that enables the file access through a browser, and transfer files using the Aspera Connect browser plugin. Additionally, you can set up the HTTP Fallback to establish HTTP- or HTTPS-based file transfers with clients that don't have the *fasp* connectivity.

When you have completed the *Testing Remotely-Initiated Transfer* on page 8, follow these steps to set up Web UI:

1. Add or review Web UI's settings

Aspera Enterprise Server installer adds and modifies the Apache configuration file. To review it, open the configuration file:

```
/etc/mcp/templates/apache2.conf
```

In this file, you should see the section that loads the modules mod_dir and mod_cgi:

```
LoadModule dir_module libexec/apache/mod_dir.so
LoadModule cgi_module libexec/apache/mod_cgi.so
AddModule mod_dir.c
AddModule mod_cgi.c
```

The *UseCanonicalName* should be set **off**:

```
UseCanonicalName off
```

Review the following section at the end of the configuration file:

```
#BEGIN_ASPERA
<Directory /usr/local/aspera/var/webtools>
   AllowOverride All
   Allow from all
</Directory>
<Directory /usr/local/aspera/var/webtools/scripts>
   AddHandler cgi-script .pl
   SetHandler cgi-script
   Options +ExecCGI
   AllowOverride All
</Directory>
ScriptAlias /aspera/scripts/ "/usr/local/aspera/var/webtools/scripts/"
Alias /aspera/ "/usr/local/aspera/var/webtools/"
#END_ASPERA
```

2. Enable SSL certificate (Optional)

To set up the SSL certificate for Web UI, first, execute these commands to copy the certificates:

```
$ isi_for_array -s -q cp /usr/local/etc/apache/ssl.crt/server.crt /usr/local/
apache2/conf/server.crt
$ isi_for_array -s -q cp /usr/local/etc/apache/ssl.key/server.key /usr/local/
apache2/conf/server.key
```

Use a text editor to open these files on each node. Do the following:

File	Instructions

/etc/mcp/templates/apache2.conf	Add or un-comment the following lines:
	`Include conf/extra/httpd-ssl.conf`
	`...`
	`LoadModule ssl_module modules/mod_ssl.so`

/usr/local/apache2/conf/extra/httpd-ssl.conf	Set the ServerName and ServerAdmin lines to reflect your system's settings:
	`ServerName www.example.com:443`
	`ServerAdmin you@example.com`

To enforce HTTPS-only SSL connections, modify this file:

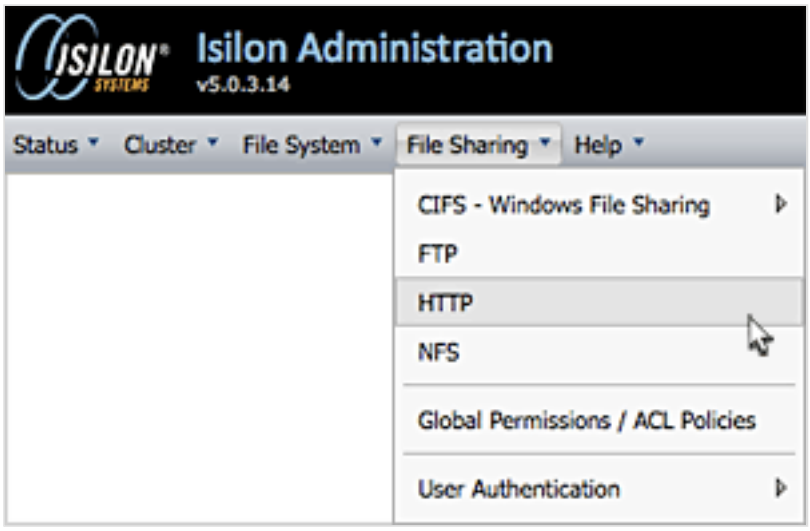
/ifs/.ifsvar/aspera/www/user/.htaccess

Add the line SSLRequireSSL at the end:

```
...
AuthName "Aspera Users"
AuthUserFile /ifs/.ifsvar/aspera/etc/webpasswd
Require valid-user
SSLRequireSSL
```

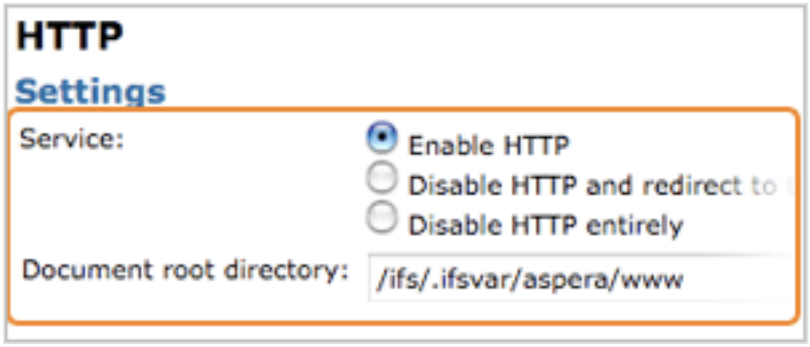
3. Set the Apache's document root, and enable the web server

   When finished, go the Isilon Web Administration. Select **File Sharing** > **HTTP**.



In the Enable HTTP Service screen, review or update the following fields with the specified values. Click **Submit** when finished:

Field	Value
Service	Enable HTTP
Document root directory	/ifs/.ifsvar/aspera/www



To restart the Apache server in command line, execute the following command to stop the service on all nodes. The *isi mcp* process will restart it automatically:

```
$ isi_for_array -s -q killall httpd
```

4. Enable the secure permissions and file access

   The secure permissions allows Web UI to accurately display the users' files, show or hide controls depending on the users' permissions, and work intuitively and securely. The most distinctive feature is the **Delete** button on the web page that allows the user to remove files. To enable this feature, execute the following command in a Terminal:

   ```
   $ isi_for_array -s -q perl /usr/local/aspera/sbin/enablesecure.pl daemon /usr/local/aspera/var/webtools/scripts
   ```

   The script will prompt you for the user that runs the Apache server (e.g.: apache, nobody) to ensure the proper file permission.

   When this feature is enabled, the user will see the **Delete** button on Web UI, allowing then to remove files on the server where permitted. You may hide the button through Web UI configuration parameter *EnableDelete*. Refer to *Web UI Configuration*.

   To reset the feature (for example, you have entered the wrong user that runs Apache), use the command:

   ```
   $ chmod 700 /usr/local/aspera/var/webtools/scripts/suiddirlist
   ```

To access Web UI, on a client machine, go to the following address with a browser:

Scope	URL
HTTP	http://<*server-ip-or-name*>/aspera/user
HTTPS	https://<*server-ip-or-name*>/aspera/user

When adding files to Web UI, avoid using the following characters in the file names: **/ \ " : ' ? > < & * |**

Testing Web UI

Test Aspera Connect client transfers through Web UI.

Follow these steps to test your server's Web UI:

The instructions involves steps on both the Enterprise Server and a client computer. Make sure that you are performing the task on the indicated machine.

1. *Enterprise Server* - Set up a test user account

On the operating system, the system user should have read and write permissions to its docroot.

Execute the command to setup the user for Web UI:

```
$ /usr/local/apache2/bin/htpasswd -c /ifs/.ifsvar/aspera/etc/webpasswd asp1
```

Open the Aspera's configuration file (aspera.conf) and set up the user's docroot information:

/ifs/.ifsvar/aspera/etc/aspera.conf

The following example uses these settings. By using the substitutable string $(name), the application will automatically replace it with user names:

Item	Value
String for generating token	secRet
Default docroot	/sandbox/$(name)

```
<CONF version="2">
   <default>
      <authorization>
         <value>allow</value> <!-- Allow token authentication for HTTP -->
         <token>
            <encryption_key>secRet</encryption_key> <!-- String for token -->
         </token>
      </authorization>
```

```
      <file_system>
         <access><paths><path>
            <absolute>/sandbox/$(name)</absolute> <!-- Default docroot -->
         </path></paths></access>
      </file_system>
   </default>
   ...
</CONF>
```

The aspera.conf sample can be used as a setup reference. Find the file in the location:

/ifs/.ifsvar/aspera/etc/aspera.conf.websample

2. *Enterprise Server* - Configure a user for *fasp* file transfer

On top of SSH authentication, Web UI uses Apache's authentication to authorize web access. To set up a system user for Apache authentication (**asp1**), use the **htpasswd** command to setup the user for Web UI. Add **-c** option if this is the first time you run the htpasswd to create the webpasswd file:

```
$ htpasswd -c /ifs/.ifsvar/aspera/etc/webpasswd asp1
```

Open the Aspera's configuration file (aspera.conf) and set up the user's docroot information:

/ifs/.ifsvar/aspera/etc/aspera.conf

The following example uses these settings. By using the substitutable string $(name), the application will automatically replace it with the login user name:

Item	Value
String for generating token	secRet
Default docroot	/sandbox/$(name)

```
<CONF version="2">
   <default>
      <authorization>
         <value>allow</value> <!-- Allow token authentication for HTTP -->
         <token>
            <encryption_key>secRet</encryption_key> <!-- String for token -->
         </token>
      </authorization>
      <file_system>
         <access><paths><path>
            <absolute>/sandbox/$(name)</absolute> <!-- Default docroot -->
```

```
        </path></paths></access>
      </file_system>
   </default>
   ...
</CONF>
```

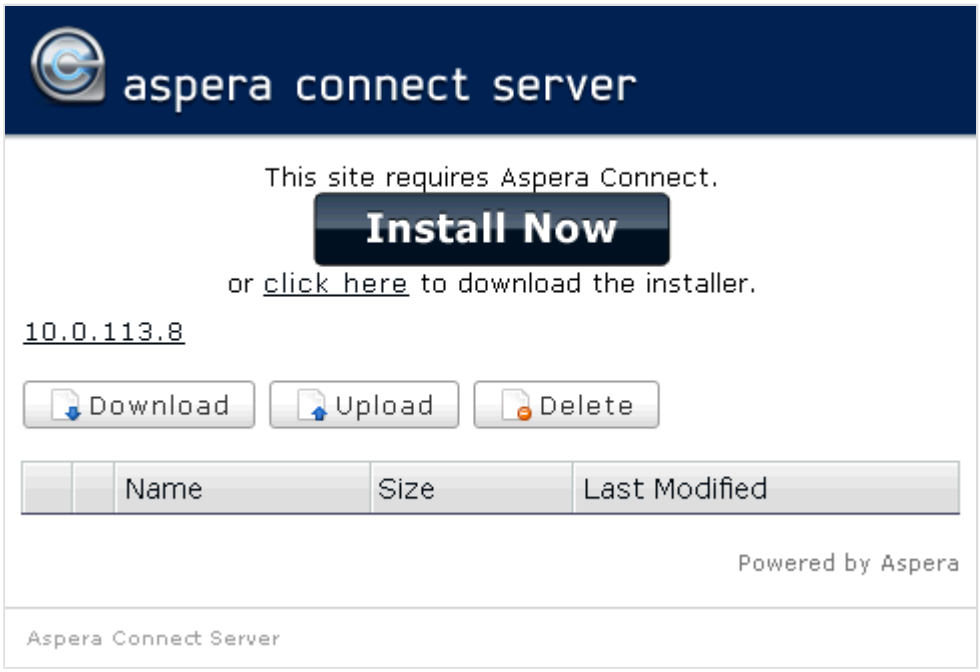
The aspera.conf sample can be used as a setup reference. Find the file in the location:

/ifs/.ifsvar/aspera/etc/aspera.conf.websample

3. *Client* - Test Web UI with the client machine

   Prepare a client computer with the supported OS and browser to test connecting to the Web UI. Refer to *Introduction* on page 4 for supported platform and browser.

   Browsing the Web UI from the client machine, you should see the Aspera Connect browser plugin installation instruction on the web page. Click either **Install Now** or **Download Aspera Connect** and follow the instructions.



   In the Web UI, click **Upload** and select one or more files to send to the *Enterprise Server*. When finished, select the uploaded files on Web UI, and click **Download**.

   For further information regarding Aspera Connect browser plugin, refer to the *Aspera Connect User Guide*.

If you are having difficulties establishing *fasp* transfers using Web UI, refer to *Clients Can't Establish Connection* on page 79.

Managing Users

Add users for the *fasp* connection authentication, and set up transfer settings for users and groups.

Setting Up Users

Add system users on your computer, and configure the account for the *fasp* transfer.

Aspera transfer products use system accounts for connection authentication, and these accounts requires additional configuration for Aspera transfers. You may specify user-based settings, such as bandwidth, document root (docroot), and file handling rules.

Follow these steps to set up transfer accounts in a Terminal:

1. Open aspera.conf with a text editor

   You need to modify the Aspera transfer product's configuration file to set up system users for *fasp* files transfers. To do so, open the file with a text editor:

   ```
   /ifs/.ifsvar/aspera/etc/aspera.conf
   ```

   You can find an aspera.conf example in this path:

   ```
   /ifs/.ifsvar/aspera/etc/samples/aspera-everything.conf
   ```

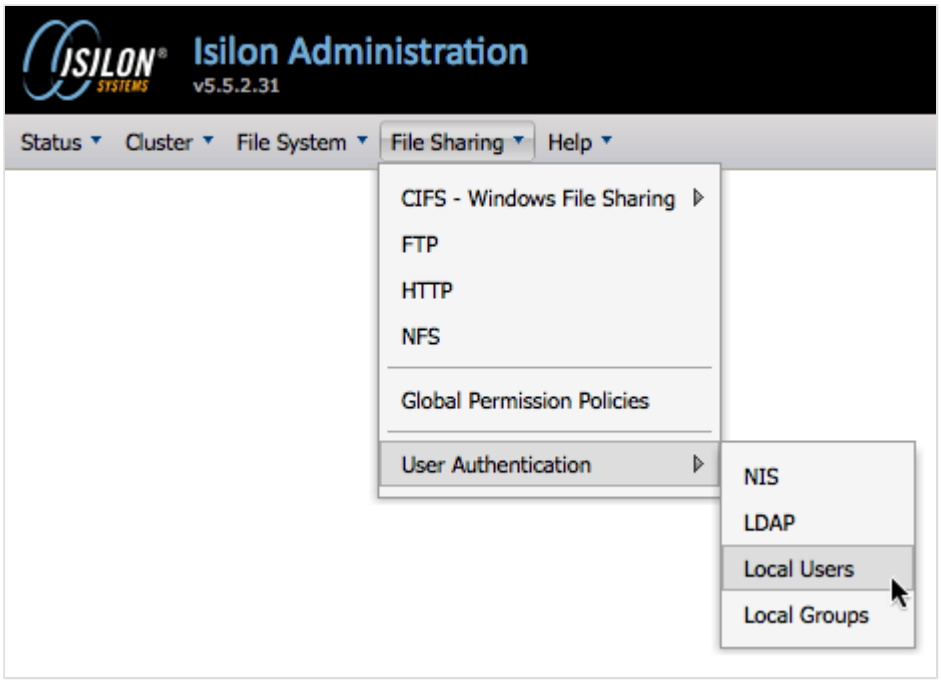
   The following steps are instructions about updating this file.

2. Restrict user permissions with Aspera Secure Shell

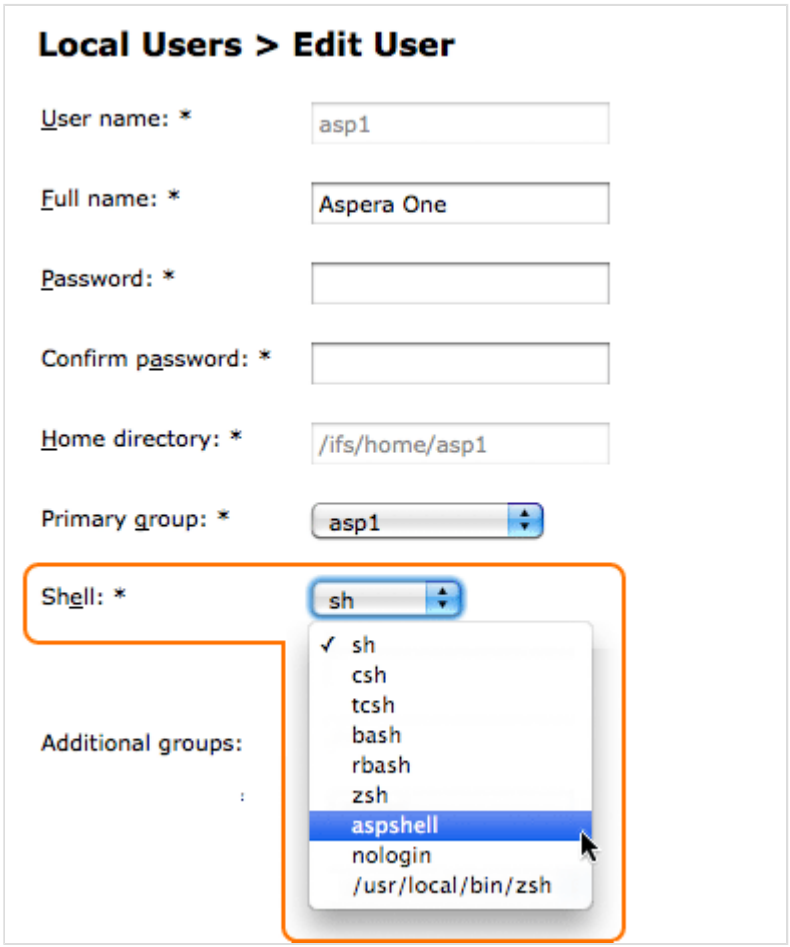
   By default, all system users can establish *fasp* connection, with full access to the system, and only restricted by file permissions. You can restrict the user's file manipulation operations through Aspera Secure Shell (aspshell). aspshell permits only the following operations:

   - Run Aspera uploads and downloads to/from this computer.
   - Establish connects in the application. Browse, create, delete, rename and list contents.
   - Make an SSH connection and run the following commands: ls, cp, mv, rm, mkdir, echo, exit.

   To set up Aspera secured shell for user accounts, go to Isilon Administration web interface, and select **File Sharing** > **User Authentication** > **Local Users**.



Click the user account to set the Aspera Secure Shell. For example, user *asp1*. In the *Edit User* page, select **aspshell** in the *Shell* field, and click **Submit**.



You can also restrict a user's file access with Document Root (docroot) settings in aspera.conf's <file_system /> (Use <absolute />, <read_allowed />, <write_allowed />, and <dir_allowed /> tags). Refer to *aspera.conf - File System* on page 66.

3. Configure a user's transfer settings

   Besides the default (global) transfer settings, you may also create user-specific and group-specific transfer settings. The user-specific settings have the highest priority, which overwrite both group and global settings.

   Add the following section to aspera.conf:

```
<?xml version='1.0' encoding='UTF-8'?>
<CONF version="2">
   <aaa>
      <realms>
         <realm>
            <users>
               <user> <!-- Each user tag contains a user's profile. -->
                  <name>asp1</name> <!-- The user name. -->
                  <authorization>...</authorization> <!-- Authorization settings. --
>
                  <transfer>...</transfer> <!-- Transfer settings. -->
                  <file_system>...</file_system> <!-- File System settings. -->
               </user>
               <user>
                  ... <!-- Another user's settings-->
               </user>
            </users>
         </realm>
      <realms>
   </aaa>
   ...
</CONF>
```

   Refer to the following sections for authorization, transfer and file_system configuration options:

Category	Description
aspera.conf - Authorization on page 55	Connection permissions, token key, and encryption requirements.
aspera.conf - Transfer on page 57	Incoming and outgoing transfer bandwidth and policy settings.
aspera.conf - File System on page 66	Network IP, port, and socket buffer settings.

4. Verify the configuration

When you have finished updating the user's settings in the aspera.conf, use the following command to verify it (In this example, verify the user asp1's settings):

```
$ /usr/local/aspera/bin/asuserdata -b -u asp1
```

Setting Up Groups

Create system groups on your computer, and set up transfer settings for the group and its members.

You can set up transfer settings based on system groups. Users in the specified groups without individual transfer settings are applied with their group's settings.

Follow these steps to set up and configure transfer groups in command line:

1. Configure a group's transfer settings

   Prepare a system group before adding it to Aspera products. The Aspera transfer product reads the system group information from the following file, make sure that the file exists before proceeding the configuration:

   ```
   /etc/group
   ```

   When a transfer group is specified, it overwrites global settings and apply configurations to its users. To add group-specific transfer settings, open aspera.conf with a text editor:

   ```
   /ifs/.ifsvar/aspera/etc/aspera.conf
   ```

   You can find an aspera.conf example in this path:

   ```
   /ifs/.ifsvar/aspera/etc/samples/aspera-everything.conf
   ```

   Add the following section to aspera.conf:

   ```
   <?xml version='1.0' encoding='UTF-8'?>
   <CONF version="2">
      <aaa>
         <realms>
            <realm>
               <users>
                  ... <!-- user-specific settings -->
               </users>
               <groups>
                  <group> <!-- Each group tag contains a group's profile. -->
                     <name>aspgroup</name> <!-- The group name. -->
                     <precedence>0</precedence> <!-- Group precedence. -->
   ```

```
                  <authorization>...</authorization> <!-- Authorization settings. --
>
                  <transfer>...</transfer> <!-- Transfer settings. -->
                  <file_system>...</file_system> <!-- File System settings. -->
               </group>
               <group>
                  ... <!-- Another group's settings-->
               </group>
            </groups>
         </realm>
      <realms>
   </aaa>
   ...
</CONF>
```

Refer to following sections for authorization, transfer and file_system configuration options:

Category	Description
Configuration Precedence on page 20	Group precedence.
aspera.conf - Authorization on page 55	Connection permissions, token key, and encryption requirements.
aspera.conf - Transfer on page 57	Incoming and outgoing transfer bandwidth and policy settings.
aspera.conf - File System on page 66	Network IP, port, and socket buffer settings.

2. Verify the configuration

When you have finished updating the group's settings in the aspera.conf, use the following command to verify it (In this example, verify the group asp-group's settings):

```
$ /usr/local/aspera/bin/asuserdata -g asp-group
```

Configuration Precedence

The priority of user, global-level and default settings.

Enterprise Server picks up settings in the order of user, groups, global and default. Within groups, a <precedence /> tag determines the group precedence if a user is in multiple groups. The following example shows the values that the user **asp1** picks up in **bold**. In this example, the user asp1 is a member of both *admin* and *xfer* groups. *admin*'s precedence setting is 0, *xfer* is 1:

Options	User asp1's Settings	Group admin's Settings	Group xfer's Settings	Global Settings	Default Settings
Target rate	**5M**	10M	15M	40M	45M
Min rate	n/a	**2M**	8M	3M	0
Policy	n/a	n/a	**Trickle**	Fair	Fair
Docroot	n/a	n/a	n/a	**/pod/$(name)**	n/a
Encryption	n/a	n/a	n/a	n/a	**any**

To determine groups' priority, use <precedence /> in group setting:

Field	Description	Values	Value Example
<precedence >	The group's precedence. Smaller number has higher priority.	0, or a positive double-digit	0, 5, 7.6, 10

To add group precedence, open **aspera.conf** with a text editor:

```
/ifs/.ifsvar/aspera/etc/aspera.conf
```

Add the option <precedence /> with a value in each group:

```
<groups>
   <group>
      <name>admin</name>
      <precedence>0</precedence>
      ...
   </group>
   <group>
      <name>xfer</name>
      <precedence>1</precedence>
      ...
   </group>
</groups>
```

Setting Up a User's Public Key

Install the public key provided by the clients to their user account.

Public key authentication is an alternative to password authentication, providing a more secure authentication method that allows users to avoid entering or storing a password, or sending it over the network.

Public key authentication is done by using the client computer to generate the key-pair (a public key and a private key), provide the public key to the server or the point-to-point, and have the public key installed on that machine.

1. Obtain the client's public key

   The client should send you an e-mail with the public key, either a text string attached in the secure e-mail, or saved as a text file. In this example, the client's login user account is *asp1*.

2. Install the client's public key to its login user account

   To install the account's public key, create a folder called .ssh in the user's home directory. This example sets up the public key for the following user:

Item	Value
User name	asp1
Key file	/ifs/id_rsa.pub
Public key install location	/ifs/home/asp1/.ssh/authorized_keys

```
$ mkdir /ifs/home/asp1/.ssh
$ cat /ifs/id_rsa.pub  >>  /ifs/home/asp1/.ssh/authorized_keys
$ chown -R asp1:asp1  /ifs/home/asp1/.ssh
```

Global Transfer Settings

The system-wide and default *fasp* transfer settings for your computer.

Setting Global Bandwidth

Allocate the global bandwidth for *fasp* file transfer.

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. This topic shows you how to optimize the transfer rate by setting up the global rate settings.

To create global bandwidth profile, open the aspera.conf (**/ifs/.ifsvar/aspera/etc/aspera.conf**) with a text editor. The following example sets the global bandwidth with these value:

Item	Value
Upload bandwidth limit (Outgoing):	88 Mbps (88000 Kbps)
Download bandwidth limit (Incoming):	99 Mbps (99000 Kbps)

```
<?xml version='1.0' encoding='UTF-8'?>
<CONF version="2">
  ...
  <trunks>
    <trunk>          <!-- Create a Vlink with 88000 Kbps bandwidth cap. -->
      <id>108</id>   <!-- ID: 108 -->
      <capacity><value>88000</value></capacity>
      <on>true</on>
    </trunk>
    <trunk>          <!-- Create a Vlink with 99000 Kbps bandwidth cap. -->
      <id>109</id>   <!-- ID: 109 -->
      <capacity><value>99000</value></capacity>
      <on>true</on>
    </trunk>
  </trunks>

  <default>   <!-- Global settings.-->
    <transfer>
      <out>   <!-- Use Vlink ID: 108 for global outgoing bandwidth. -->
        <bandwidth><aggregate><trunk_id>108</trunk_id></aggregate></bandwidth>
      </out>
      <in>    <!-- Use Vlink ID: 109 for global incoming bandwidth. -->
        <bandwidth><aggregate><trunk_id>109</trunk_id></aggregate></bandwidth>
```

```
      </in>
    </transfer>
  </default>
</CONF>
```

Setting Up Virtual Links

Create and apply the aggregate bandwidth cap.

Virtual link (Vlink) is a feature that allows "virtual" bandwidth caps. Transfer sessions assigned to the same "virtual" link conform to the aggregate bandwidth cap and attain an equal share of it. This section first shows you how to set up Vlinks, then explains how to apply it to computers or users.

Follow these steps to configure Vlinks:

1. Create Vlinks in aspera.conf

   To create Vlinks, open aspera.conf with a text editor:

   ```
   /ifs/.ifsvar/aspera/etc/aspera.conf
   ```

   You can refer to the configuration example:

   ```
   /ifs/.ifsvar/aspera/etc/samples/aspera-everything.conf
   ```

   Locate or create the section *<trunks>...</trunks>*. For each vlink, add a **<trunk>...</trunk>**:

   ```
   <CONF version="2">
     ...
     <trunks>
       <trunk>
         <id>108</id>                    <-- Vlink ID -->
         <name>50Mbps cap</name>         <-- Vlink Name -->
         <capacity>
            <value>50000</value>         <-- Capacity -->
         <capacity>
         <on>true</on>                   <-- On -->
         <mcast_port>55001</mcast_port> <-- Multicast Port -->
       </trunk>
     </trunks>
   </CONF>
   ```

   Here is a description of the Vlink tags:

#	Tag	Description	Values	Default
1	Vlink ID	The Vlink ID. Sessions assigned with the same trunk ID share the same bandwidth cap.	positive integer between 1 and 255.	N/A
2	Vlink Name	The Vlink name. This value has no impact on actual bandwidth capping.	text string	blank
3	Capacity	This value reflects the virtual bandwidth cap in Kbps. When applying this Vlink to a transfer (e.g. Default outgoing), the transfer's bandwidth will be restricted by this value.	positive integer in Kbps	50000
4	On	Select **true** to activate this Vlink; select **false** to deactivate it.	true/false	false
5	Multicast Port	This sets the UDP port through which virtual link sends and receives multicast communication messages. Sessions sharing the same virtual bandwidth cap needs to have the same port number. To avoid port conflicts, it is recommended to use the default UDP port 55001. Do NOT set the port number to the same one used by fasp data transfer (33001).	positive integer between 1 and 65535	55001

2. Apply a Vlink to a transfer

   You can assign a Vlink to a global, a user, or a group settings in the aspera.conf.

   In this example, assuming we have created three vlinks: 108, 109 and 110, apply these vlinks to the outgoing bandwidth of global and a user:

```
<CONF version="2">
  ...
  <default>
    <transfer>
      <out>
        <bandwidth><aggregate>
          <trunk_id>108</trunk_id> <!-- Vlink #108 for the default outgoing
 sessions. -->
        </aggregate></bandwidth>
      </out>
      <in>
```

```
      ...
    </in>
  </transfer>
 </default>
 <aaa><realms><realm>
  <users>
    <user>
      <name>asp1</name>
      <transfer>
        <out>
          <bandwidth><aggregate>
            <trunk_id>109</trunk_id> <!-- Vlink #109 to the user asp1's outgoing
 sessions. -->
          </aggregate></bandwidth>
        </out>
        <in>
         ...
        </in>
      </transfer>
    </user>
  </users>
 <realm></realms></aaa>
</CONF>
```

Transfer Server Configuration

Set up the transfer server and more global/default settings.

To configure the Aspera Central transfer server, open aspera.conf with a text editor (**/ifs/.ifsvar/aspera/etc/aspera.conf**), locate or create the transfer server's section *<central_server>...</central_server>*:

```
<CONF version="2">
   ...
   <central_server>
      <address>127.0.0.1</address> <!-- Address -->
      <port>40001</port>           <!-- Port -->
   </central_server>
</CONF>
```

The Aspera Central transfer server's configuration options:

Field	Description	Values	Default
Address	This is the network interface address on which the transfer server listens. The default value 127.0.0.1 enables the transfer server to accept transfer requests from the local computer; The value 0.0.0.0 allows the transfer server to accept requests on all network interfaces for this node. Alternatively, a specific network interface address may be specified.	valid IPv4 address	127.0.0.1
Port	The port at which the transfer server will accept transfer requests.	positive integer between 1 and 65535	40001

For the general configuration options (Authorization, Bandwidth, Network, File Handling, and Docroot), refer to the following sections:

For additional Enterprise Server features (Database Logger), refer to the following section:

For more configuration options, refer to these sections:

Category	Description
aspera.conf - Authorization on page 55	Connection permissions, token key, and encryption requirements.
aspera.conf - Transfer on page 57	Incoming and outgoing transfer bandwidth and policy settings.
aspera.conf - File System on page 66	Network IP, port, and socket buffer settings.

```
$ isi_for_array -s -q /etc/rc.d/asperacentral restart
$ isi_for_array -s -q /etc/rc.d/asperahttpd restart
```

Aspera Sync

The asperasync file synchronization command.

asperasync Syntax

Configure the asperasync to perform file synchronization.

Aspera Sync is a command-line tool that can be used to monitor configured "hot folders" for changes, automatically transferring any new or modified files. It can be used for one-way replication between two locations or simply as a way of forwarding files in your work-flow. Sync runs as a service in the background.

The following shell script is used to execute Aspera Sync:

```
/usr/local/aspera/bin/asperasync.sh
```

You can modify the Aspera Sync transfer settings directly in this file. For example, the Aspera Sync is initiated every 10 seconds with target rate 100Mbps (100000Kbps), you can change them by modifying the values of INTERVAL and TARGETRATE in this file, respectively:

```
...
INTERVAL=10          # Interval for directory sync is in seconds
TARGETRATE=100000    # The highest rate the sync will try to achieve.
...
```

To execute the Aspera Sync, use the command asperasync.sh with the following syntax (The environment variable ASPERA_SCP_PASS=pswd can be set in a separate line with export ASPERA_SCP_PASS=pswd):

```
$ ASPERA_SCP_PASS=pswd  /usr/local/aspera/bin/asperasync.sh  src-dir  des-dir   arg
```

Parameter	Description
pswd	The password for remote login. Use the environment variable **ASPERA_SCP_PASS** to set it.
src-dir	The source folder.
des-dir	The destination folder. Either the source or the destination folder has to be a local directory on your computer. The parameter on the remote-side takes the form user@remote-address.
arg	The ascp command options for this synchronization. See *ascp Usage* on page 47.

asperasync Examples

Examples of the asperasync settings.

This topic demonstrates the asperasync configuration settings with the following examples:

1. Start a regular synchronization

Item	Value
Remote Login	asp1 / 1234
Source	(Local)
Source folder	/local-src
Destination	10.0.0.5 (Remote)
Destination Folder	/remote-desc

```
$ export ASPERA_SCP_PASS=1234
$ /usr/local/aspera/bin/asperasync.sh /local-src asp1@10.0.0.5:/remote-dest
```

2. Start the same synchronize with additional ascp command options

Item	Value
ascp Options	Target rate 10Mbps (-l 10000), resume with a full file checksum(-k3)

```
$ export ASPERA_SCP_PASS=1234
$ /usr/local/aspera/bin/asperasync.sh /local-src asp1@10.0.0.5:/remote-dest -l 10000
 -k3
```

3. Keep the script running when the Terminal session is closed

   To keep the script running when the Terminal session is closed, start the command with *nohup*, and add a **&** at the end:

```
$ export ASPERA_SCP_PASS=1234
$ nohup  /usr/local/aspera/bin/asperasync.sh  /local-src asp1@10.0.0.5:/remote-dest
 &
```

4. Synchronize from a network shares location

   To synchronize from a network shares location, through a remote host, to a local directory:

Item	Value
Remote Login	asp1 / 1234
Source	10.0.0.10 (Remote)
Source folder	\\1.2.3.4\nw-share-src (Network Shares drive)
Destination	(Local)

Item	Value
Destination Folder	/inbox

```
$ export ASPERA_SCP_PASS=1234
$ /usr/local/aspera/bin/asperasync.sh asp1@10.0.0.10:"//1.2.3.4/nw-share-src" /inbox
```

Database Logger

Using a MySQL database to keep track of all transfers on your server.

Setting Up Database Logger

Import Database Logger's schema to the MySQL database, and set up the proper access permissions.

The Database Logger is a feature that record all the server's Aspera transactions to a MySQL database. Follow these steps to set it up:

1. Prepare the MySQL Database Server

   The Database Logger supports MySQL Server 5 and above. Prepare a system with MySQL installed and configured. The latest MySQL software download can be found at *http://dev.mysql.com/downloads/*.

2. Create the database

   Locate the Database Logger schema file in the following location:

   ```
   /usr/local/aspera/var/create_logger_database.sql
   ```

   Copy the file to the computer that runs the MySQL Server, and use the following commands to import this file into the database. This example uses the following settings:

Item	Value
MySQL login	root

   ```
   $ mysql -u root  -p  <  /temp/create_logger_database.sql
   $ mysql  -u root  -p aspera_console
   mysql> show tables;
   ```

   When finished, the database **aspera_console** will be imported to the MySQL Server. You should see the tables of this database.

3. Set up the MySQL user for Database Logger

   A database user with proper permissions is required for Database Logger. In the following example, the user account is created with the setup:

Item	Value
MySQL login	logger
Password	logger-password

Item	Value
IP address of remote machine	10.0.0.5

```
1> CREATE USER 'logger'@'10.0.0.5' IDENTIFIED by 'logger-password';
2> GRANT SELECT, INSERT, UPDATE ON aspera_console.fasp_files TO 'logger'@'10.0.0.5';
3> GRANT SELECT, INSERT, UPDATE ON aspera_console.fasp_sessions TO
 'logger'@'10.0.0.5';
4> GRANT SELECT, INSERT, UPDATE ON aspera_console.fasp_nodes TO 'logger'@'10.0.0.5';
5> GRANT INSERT ON aspera_console.fasp_rates TO 'logger'@'10.0.0.5';
6> FLUSH PRIVILEGES;
```

4. Modify MySQL Settings (Only if MySQL server is on Windows)

   If you are running the database on a Windows machine, open the MySQL config file, for example:

   C:\Program Files\MySQL\MySQL Server (Version)\my.ini

   Find the line that says **[mysqld]**, and add the line immediately under it:

```
skip-name-resolve
```

The Database Logger's schema can be found in the document *Aspera Database Logger Schema*.

Configuring the Database Logger

Update the settings in the Aspera configuration to establish connections with the MySQL database.

To configure Database Logger, open aspera.conf with a text editor (**/ifs/.ifsvar/aspera/etc/aspera.conf**). Locate or create the section *<database>...</database>*:

```
<CONF version="2">
   ...
   <database>
      <server>            <!-- Host IP -->
        127.0.0.1
      </server>
      <port>              <!-- Port -->
        4406
      </port>
      <user>              <!-- User -->
        logger
      </user>
      <password>          <!-- Password -->
```

```
      logger-password
    </password>
    <database_name>     <!-- Database Name -->
      aspera_console
    </database_name>
    <threads>           <!-- Threads -->
      10
    </threads>
    <exit_on_database_error>     <!-- Stop Transfers on Database Error -->
      false
    </exit_on_database_error>
    <session_progress>           <!-- Show Session Progress -->
      true
    </session_progress>
    <session_progress_interval>  <!-- Session Progress Interval -->
      1
    </session_progress_interval>
    <file_events>                <!-- Show File Events -->
      true
    </file_events>
    <file_progress>              <!-- Show File Progress -->
      true
    </file_progress>
    <files_progress_interval>    <!-- File Progress Interval -->
      1
    </files_progress_interval>
    <files_per_session>          <!-- File Per Session -->
      0
    </files_per_session>
    <ignore_empty_files>         <!-- Ignore Empty Files -->
      false
    </ignore_empty_files>
    <ignore_no_transfer_files>   <!-- Ignore No-transfer Files -->
      false
    </ignore_no_transfer_files>
    <rate_events>                <!-- Show Rate Events -->
      true
    </rate_events>

</database>
...
```

```
</CONF>
```

You can find a Database Logger configuration example in this file:

/ifs/.ifsvar/aspera/etc/samples/aspera-everything.conf

If you have modified these settings, execute these commands to restart Aspera Central and HTTP Fallback Server:

```
$ isi_for_array -s -q /etc/rc.d/asperacentral restart
$ isi_for_array -s -q /etc/rc.d/asperahttpd restart
```

Here is a list of all the Database Logger configuration options:

#	Field	Description	Values	Default
1	Host IP	The MySQL server's IP address.	valid IPv4 address	127.0.0.1
2	Port	The MySQL server's port number.	integer between 1 and 65535	4406
3	User	User login for the database server.	text string	blank
4	Password	The database user account's password.	text string	blank
5	Database Name	Name of the database used to store Aspera transfer data.	text string	blank
6	Threads	The number of parallel connections used for database logging. A higher value may be useful when a large number of files are being transferred within a given timeframe.	integer between 1 and 40	10
7	Stop Transfers on Database Error	Quits all ongoing transfers and no new transfers are permitted when a database error prevents data from being written to the database. Set this to true if all transfers must be logged by your organization.	• true • false	false
8	Show Session Progress	Setting this value to true will log transfer status such as number of files transferred, and bytes transferred, at a given interval.	• true • false	true
9	Session Progress Interval	The frequency at which an Aspera node logs transfer session information, in seconds. up to 65535 seconds.	integer between 1 and 65535	1
10	Show File Events	Setting this value to true enables the logging of complete file paths and file names. Performance may	• true	true

#	Field	Description	Values	Default
		be improved when transferring datasets containing thousands of files. Also see File Per Session for setting a threshold for the number of files to log per session.	• false	
11	Show File Progress	Setting this value to true will log file status such as bytes transferred, at a given interval.	• true • false	true
12	File Progress Interval	The frequency at which an Aspera node logs file transfer information, in seconds.	integer between 1 and 65535	1
13	Files Per Session	The value set will be the cut-off point for file names logged in a given session. For instance, if the value is set to 50, the first 50 file names will be recorded for any session. The session will still record the number of files transferred along with the number of files completed, failed or skipped. The default setting of 0 logs all file names for a given session.	positive integer or zero (all file names)	0
14	Ignore Empty Files	Setting this to true will block the logging of zero-byte files.	• true • false	false
15	Ignore No-transfer Files	Setting this to true will block the logging of files that have not been transferred because they exist at the destination at the time the transfer started.	• true • false	false
16	Show Rate Events	Setting this to true will log changes made to the Target Rate, Minimum Rate, and Transfer Policy by any user or Aspera node administrator during a transfer.	• true • false	true

Pre- and Post-Processing (Prepost)

Execute scripts before and after the *fasp* file transfers on your server.

Setting Up Prepost

Enable the pre- and post-processing on your server.

The Aspera server executes a shell script at a pre-defined location (). This script is executed on four events during the transfer: start of session, end of session, and start and end of each file in the session.

aspera-prepost can execute other shell scripts, Perl scripts, native executables or Java programs. Aspera sets several environment variables that aspera-prepost can use, and that can be used by your own customer scripts. Those environment variables are described in detail in *Prepost Variables* on page 37.

Depending on usage, Pre- and Post-Processing may consume a great amount of system resources. Please evaluate the system performance and apply this feature appropriately.

Follow these steps to set up the Pre- and Post-Processing:

1. Set up the shell script file

   Locate this file:

   ```
   /usr/local/aspera/var/aspera-prepost-disable
   ```

   Copy it to the following file. Make sure the execute privileges is enabled (At least r-xr-xr-x):

   ```
   /usr/local/aspera/var/aspera-prepost
   ```

2. Create the scripts

   The Pre- and Post-Processing script, aspera-prepost, can contain the prepost processing steps, can execute other programs including shell scripts, or a combination of both. Often, aspera-prepost will check for certain conditions (based on the environment variables) and then call a specified external executable based on the conditions.

   is executed four times during a transfer: session start, file start, file stop, and session stop. You can use both the variables *TYPE* and *STARTSTOP* to specify a particular state.

   For the complete list of all variables, refer to *Prepost Variables* on page 37.

3. Include commands or **shell** scripts into aspera-prepost

   ```
   ...
   perl  script1.pl
   ...
   ```

Prepost Variables

The pre-defined variables for setting up the pre- and post-processing.

The following tables list all variables:

> The prepost variables are case-sensitive.

For Type Session and Type File

Variable	Description	Values	Example
TYPE	The event type.	• Session • File	
STARTSTOP	The status start or stop.	• Start • Stop	
DIRECTION	The transfer direction.	• send • recv	
SESSIONID	The session id.	string	
USERSTR	The user string, such as additional variables.	string	
STATE	The transfer state.	• started • success • failed	
ERRCODE	The error code.	string	
ERRSTR	The error string.	string	

For Type Session

Variable	Description	Values	Example
SOURCE	The full path of the source file.	string	
TARGET	The full path of the target directory.	string	
PEER	The peer name or IP address.	string or valid IPv4 address	
USERID	The user ID	string	

Variable	Description	Values	Example
USER	The user name	string	
TARGETRATE	The initial target rate, in bps.	positive integer	
MINRATE	The initial minimum rate, in bps.	positive integer	
RATEMODE	The transfer policy.	• adapt • fixed	
SECURE	Transfer encryption.	• yes • no	
LICENSE	The license account and serial number.	string	
PEERLICENSE	The peer's license account and serial number.	string	
FILECOUNT	The number of files.	positive integer	
TOTALBYTES	The total bytes transferred.	positive integer	
TOTALSIZE	The total size of files being transferred in bytes.	positive integer	
FILE1	The first file.	string	
FILE2	The second file.	string	
FILELAST	The last file.	string	
TOKEN	The user-defined security token.	string	
COOKIE	The user-defined cookie string.	string	
MANIFESTFILE	The full path to the manifest file.	string	

For Type File

Variable	Description	Values	Example
FILE	The file name.	string	
SIZE	The file size in bytes.	positive integer	
STARTBYTE	The start byte if resumed.	positive integer	

Variable	Description	Values	Example
RATE	The transfer rate in Mbps.	double-digit fixed point value	
DELAY	The measured network delay, in ms.	positive integer	
LOSS	The network loss in percentage.	double-digit fixed point value	
REXREQS	The total number of retransmission requests.	positive integer	
OVERHEAD	The total number of duplicate packets.	positive integer	

Prepost Examples

Pre- and post-processing script examples.

Here are a few examples of common prepost processing:

1. **Shell** - Change file and directory permissions

   In shell script, change file and directory permissions after receiving, and log into the file */tmp/p.log*:

```
if [ "$TYPE" == File ]; then
  if [ "$STARTSTOP" == Stop ]; then
    echo "The file is: $FILE" >> /tmp/p.log
    chmod 777 $FILE
  fi
fi
```

2. **Shell** - Forward files to another computer

   In shell script, transfer received files to a third computer *10.10.10.10*, and remove the local copy:

```
TARGET = aspera@10.10.10.10:/tmp
RATE = 10m
export ASPERA_SCP_PASS=aspera
if [ "$TYPE" == File ]; then
  if [ "$STARTSTOP" == Stop ]; then
    if [ "$STATE" == success ]; then
      if [ "$DIRECTION" == recv ]; then
        logger -plocal2.info "Move file $FILE to $TARGET"
        ascp -T -o RemoveAfterTransfer=yes -l $RATE $FILE $TARGET
      fi
```

```
    fi
  fi
fi
```

3. **Shell** - Create a log of successfully-transferred files

   In shell script, store successfully-transferred file as a list into the file */tmp/aspera.transfer.log*:

```
if [ "$TYPE" == File ]; then
  if [ "$STARTSTOP" == Stop ]; then
    if [ "$SIZE" -gt 0 ]; then
      if [ `expr "$SIZE" - "$STARTBYTE"` -gt 0 ]; then
        echo `date` >> /tmp/aspera.transfer.log
        echo "$STATE $FILE $SIZE bits transferred" >> /tmp/aspera.transfer.log
      fi
    fi
  fi
fi
```

Setting Up Email Notification

Configure the email notification, a prepost application.

Email Notification is a built-in Pre- and Post-Processing application that generates customized e-mails based on transfer events. Your server should have the Pre- and Post-Processing configured in order to run this application. Refer to *Setting Up Prepost* on page 36.

Email Notification requires a SMTP server that matches the following configurations:

- An open SMTP server you can reach on your network
- The SMTP Server must not use any external authentication or SSL.

Follow these steps to set it up:

1. Prepare the Email Notification configuration template

   Open the aspera.conf:

   ```
   /ifs/.ifsvar/aspera/etc/aspera.conf
   ```

   Locate or create the section *<EMAILNOTIF>...</EMAILNOTIF>*:

```
<CONF version="2">
   ...
   <EMAILNOTIF>
```

```
    <MAILLISTS
       mylist = "asperausers@example.com, admin@example.com"
       myadminlist = "admin@example.com"
    />

    <FILTER
       MAILLISTS = "mylist"
       TARGETDIR = "/content/users"
    />

    <MAILCONF
       DEBUG = "0"
       FROM = "asperaserver@example.com"
       MAILSERVER = "mail.example.com"
       SUBJECT = "Transfer %{SOURCE} %{TARGET} - %{STATE}"
       BODYTEXT =
        "Aspera transfer: %{STATE}%{NEWLINE}%{TOTALBYTES} bytes in
        %{FILECOUNT} files: %{FILE1}, %{FILE2}, ...%{FILELAST}."
    />
  </EMAILNOTIF>
</CONF>
```

You can find the aspera.conf example in this path:

```
/ifs/.ifsvar/aspera/etc/sample/aspera-sample.email.conf
```

2. Set up the basic Notification function in <MAILCONF />

<MAILCONF /> defines the general e-mail configuration, including the sender, the mail server, and the body text. In the SUBJECT and BODYTEXT options, the Pre- and Post-Processing variables can be used with the format %{variable}, such as %{STATE} for the variable STATE. For the complete list of the variables, Refer to *Prepost Variables* on page 37.

MAILCONF Field	Description	Values	Example
FROM	**Required** The e-mail address to send notifications from.	a valid email address	FROM="admin@example.com"
MAILSERVER	**Required** The outgoing mail server (SMTP).	A valid URL	MAILSERVER="mail.example.com"
SUBJECT	General subject of the e-mail.	text string	SUBJECT="Transfer: %{STATE}"
BODYTEXT	General body of the e-mail.	text string	BODYTEXT="Transfer has %{STATE}."

MAILCONF Field	Description	Values	Example
DEBUG	Print debugging info and write to the logs.	0 / 1	DEBUG="0"

3. Create mailing lists in <MAILLISTS />

   <MAILLISTS /> defines sets of mailing lists. For example, to create the following mailing list:

Item	Value
Mailing list name	list1
Emails to include	janedoe@companymail.com, johndoe@companymail.com

   Specify the mailing list in the form:

```
<MAILLISTS
   list1 = "janedoe@companymail.com, johndoe@companymail.com"
/>
```

4. Set up mailing filters in <FILTER />

   <FILTER /> defines E-mail Notification conditional filters. When the conditions are met, an customized e-mail will be sent to the indicated mailing list. Multiple filters are allowed.

   The values in the filter are matched as substrings, for example, USER = root means the value would match strings like root, treeroot, and root1. The Pre- and Post-Processing variables can be used with the format %{variable}, such as %{STATE} for the variable STATE. For the complete list of the variables, Refer to *Prepost Variables* on page 37.

FILTER Field	Description	Values	Example
MAILLISTS	**Required** The e-mail lists to send to. Separate lists with comma (,).	text string	MAILLISTS="mylist"
USER	Login name of the user who transferred the files.	text string	USER="asp1"
SRCIP	Source IP of the files.	a valid IPv4 address	SRCIP="10.0.1.1"
DESTIP	Destination IP of the files.	a valid IPv4 address	DESTIP="10.0.1.5"
SOURCE	The top-level directories and files that were transferred.	text string	SOURCE="/folder1"
TARGETDIR	The directory that the files were sent to.	text string	TARGETDIR="/folder2"

FILTER Field	Description	Values	Example
SUBJECTPREFIX	The Email subject, preceded by the SUBJECT in <MAILCONF />.	text string	SUBJECTPREFIX="Sub"
BODYPREFIX	The e-mail body, preceded by the BODYTEXT in <MAILCONF />.	text string	BODYPREFIX="Txt"
TOTALBYTESOVER	Send e-mail when total bytes transferred is over this number. This only applies to e-mails sent at the end of a transfer.	positive integer	TOTALBYTESOVER="9000"
SENDONSESSION	Send e-mail for the entire session.	yes / no	SENDONSESSION="yes"
SENDONSTART	Send e-mail when transfer is started. This setting is dependent on *SENDONSESSION="yes"*.	yes / no	SENDONSTART="yes"
SENDONSTOP	Send e-mail when transfer is stopped. This setting is dependent on *SENDONSESSION="yes"*.	yes / no	SENDONSTOP="yes"
SENDONFILE	Send e-mail for each file within a session.	yes / no	SENDONFILE="yes"

Email Notification Examples

Email Notification configuration examples.

This topic demonstrates the Email Notification setup with the following examples:

1. Notify when a transfer session is completed

   When a transfer session is finished, an e-mail with brief session summary will be sent to the "list1".

```
<EMAILNOTIF>
  <MAILLISTS
    list1 ="janedoe@companyemail.com, johndoe@companyemail.com"
  />

  <MAILCONF
    FROM="Aspera Notifier &lt;admin@companyemail.com&gt;"
    MAILSERVER="smtp.companyemail.com"
    BODYTEXT="%{NEWLINE}Powered by Aspera Inc."
  />

  <FILTER
    MAILLISTS="list1"
```

```
    SENDONSESSION="yes"
    SUBJECTPREFIX="Aspera Transfer - %{USER} "
    BODYPREFIX="Status: %{STATE}%{NEWLINE} File Count: %{FILECOUNT}"
  />
</EMAILNOTIF>
```

2. Notify when a session is initiated and completed

   Send a transfer notice e-mail when a transfer is initiated; send a summary e-mail when finished.

```
<EMAILNOTIF>
  <MAILLISTS
    list1 ="janedoe@companyemail.com, johndoe@companyemail.com"
  />
  <MAILCONF
    FROM="Aspera Notifier &lt;admin@companyemail.com&gt;"
    MAILSERVER="smtp.companyemail.com"
    SUBJECT=" by %{USER}"
    BODYTEXT="%{NEWLINE}Powered by Aspera Inc."
  />

  <FILTER
    MAILLISTS="list1"
    SENDONSTART="yes"
    SENDONSTOP="no"
    SUBJECTPREFIX="Transfer Started"
    BODYPREFIX="Source: %{PEER}%{NEWLINE} Target: %{TARGET}"
  />

  <FILTER
    MAILLISTS="list1"
    SENDONSTART="no"
    SENDONSTOP="yes"
    SUBJECTPREFIX="Transfer Completed"
    BODYPREFIX="
      Status: %{STATE}%{NEWLINE}
      File Count: %{FILECOUNT}%{NEWLINE}
      Source: %{PEER}%{NEWLINE}
      Target: %{TARGET}%{NEWLINE}
      Bytes Transferred: %{TOTALBYTES} Bytes%{NEWLINE}
    "
  />
```

```
</EMAILNOTIF>
```

3. Send different email for regular transfers and Aspera Sync transfers

   When Aspera Sync triggers a transfer (Assuming only Aspera Sync uses the folder /sync-folder), an e-mail will be sent to the "mediaGroup". When a regular transfer occurs (Files sent to /upload), a different notification will be sent to the "mediaLead" and the "adminGroup".

```
<EMAILNOTIF>
  <MAILLISTS
    mediaGroup ="johndoe@companyemail.com, janedoe@companyemail.com"
    mediaLead ="janedoe@companyemail.com"
    adminGroup ="admin@companyemail.com, root@companyemail.com"

  />
  <MAILCONF
    FROM="Aspera Notifier &lt;admin@companyemail.com&gt;"
    MAILSERVER="smtp.companyemail.com"
    BODYTEXT="%{NEWLINE}Powered by Aspera Inc."
  />

  <FILTER
    MAILLISTS="list1"
    SENDONSESSION="yes"
    DESTIP="192.168.1.10"
    TARGETDIR="/sync-folder"
    SUBJECTPREFIX="Aspera Sync #1 - From %{PEER}"
    BODYPREFIX="Status: %{STATE}%{NEWLINE} File Count: %{FILECOUNT}"
  />

  <FILTER
    MAILLISTS="list2,list3"
    SENDONSESSION="yes"
    TARGETDIR="/upload"
    SUBJECTPREFIX="Transfer - %{USER}"
    BODYPREFIX="
      Status: %{STATE}%{NEWLINE}
      Source: %{PEER}%{NEWLINE}
      File Count: %{FILECOUNT}%{NEWLINE}
      Bytes Transferred: %{TOTALBYTES} Bytes%{NEWLINE}
    "
    />
```

```
</EMAILNOTIF>
```

Transferring in Command-line

Initiate transfers in Command-line.

ascp Usage

The ascp command reference.

ascp is a command-line *fasp* transfer program. This topic covers the complete command usage, including the general syntax guideline, supported environment variables, synopsis, and the options.

General Syntax Guideline

Item	Decription
symbols used in the paths	Use single-quote (' ') and forward-slashes (/) on all platforms.
Characters to avoid	/ \ " : ' ? > < & * |

Environment Variables

If needed, you can use the command to set the password, token, and cookie in the environment variables. Replace the highlighted text with your own values:

Item	Initiation Command
Password	export ASPERA_SCP_PASS=the-password
Token	export ASPERA_SCP_TOKEN=the-token
Cookie	export ASPERA_SCP_COOKIE=the-cookie
Content Protection Password	export ASPERA_SCP_FILEPASS=content-protect-password

ascp Synopsis

```
ascp  [-{ATdpqv}] [-{Q|QQ}] [-l max-rate] [-m min-rate] [-w{f|r} [-K probe-rate]]
      [-k {0|1|2|3}] [-i pubkey-file] [-Z dgram-size] [-M mgmt-port]
      [-u user-string] [-X rexmsg-size] [-g read-size] [-G write-size]
      [-S remote-ascp] [-L local-logdir] [-R remote-logdir][-e pre-post]
      [-f config-file] [-C n-id:n-count] [-E pattern1 -E pattern2...]
      [-O fasp-port] [-P ssh-port] [-o Option1=x[,Option2=y..]]
      [-U {1|2}] [-W token-string] [-@[range-low:range-high]] [-6]
      [-y {0|1}] [-j {0|1}] [-Y key-file] [-I certif-file] [-t port]
      [-x proxy-server] [[user@]host1:]source-file [[user@]host2:]target-path
```

ascp Options

Option	Description
-A	Display version and license information; then exit.
-T	Disable encryption for maximum throughput.
-d	Create target directory if it doesn't already exist.
-p	Preserve file timestamp.
-q	Quiet flag, to disable progress display.
-v	Verbose mode, print connection and authentication debug messages in the log file.
-{Q|QQ}	Enable fair (-Q) or trickle (-QQ) transfer policy. Use the -l and -m to set the target and minimum rates.
-l *target_rate*	Set the target transfer rate in Kbps. *Default: 10000*
-m *min-rate*	Set the minimum transfer rate in Kbps. *Default: 0*
-w{r|f}	Test bandwidth from server to client (r) or client to server (f). Currently a beta option.
-K *probe-rate*	Set probing rate (Kbps) when measuring bottleneck bandwidth.
-k {0|1|2|3}	Enable resuming partially transferred files. *(Default: 0)*. • **0** Always retransfer the entire file. • **1** Check file attributes and resume if they match. • **2** Check file attributes and do a sparse file checksum; resume if they match. • **3** Check file attributes and do a full file checksum; resume if they match.
-i *key-file*	Use public key authentication and specify the private key file. Typically, the private key file is in the directory *$HOME/.ssh/id_[algorithm]*.
-Z *dgram-size*	Specify the datagram size (MTU) for *fasp*. By default it uses the detected path MTU.
-M *port*	Set a management port for monitoring and controlling the transfer.
-u *user-string*	Apply user string, such as variables for Pre- and Post-Processing, in the transfer.
-X *rexmsg-size*	Adjust the size in bytes of a retransmission request. *(Max: 1440)*.
-g *read-size*	Set the read block size (in bytes). E.g. 1M for 1 megabyte.
-G *write-size*	Set the write block size (in bytes), E.g. 1M for 1 megabyte.
-S *remote-ascp*	Specify the name of the remote ascp binary if different.
-L *local-log-dir*	Specify a logging directory in the local host, instead of using the default directory.
-R *remote-log-dir*	Specify a logging directory in the remote host, instead of using the default directory.

Option	Description
-e *prepost*	Specify an alternate pre-post command. Use complete path and file name.
-f *config-file*	Specify an alternate Aspera configuration file other than aspera.conf.
-C *n-id:n-count*	Use parallel transfer on a multi-node/core system. Specify the node id (nid) and count(ncount) in the format 1:2, 2:2. Assign each participant an independent UDP port.
-E *pattern*	Exclude files or directories with the specified pattern in the transfer. This option can be used multiple times to exclude many patterns. Up to 16 patterns can be used by using -E. Two symbols can be used in the pattern: • ***** (asterisk) represents zero to many characters in a string, for example "*.tmp" matches ".tmp" and "abcde.tmp". • **?** (question mark) represents one character, for example "t?p" matches "tmp" but not "temp".
-O *fasp-port*	Set the UDP port used by *fasp* for data transfer. *(Default: 33001)*
-P *ssh-port*	Set the TCP port used for *fasp* session initiation. *(Default: 22)*

-o	Advanced ascp options as listed below. Use comma "," to separate:
	• **SkipSpecialFiles=no** Skip special files such as devices and pipes. *(yes / no. Default: no)*
	• **RemoveAfterTransfer=no** Remove source file except folder when finish. *(yes / no. Default: no)*
	• **RemoveEmptyDirectories=no** Remove empty folder on the source. *(yes / no. Default: no)*
	• **PreCalculateJobSize=no** Calculate total size before transfer. *(yes / no. Default: no)*
	• **Overwrite=diff** Overwrite files with the same name. This option takes following values *(Default: diff)*
	  • **always** Always overwrite the file.
	  • **never** Never overwrite the file.
	  • **diff** Overwrite if file is different from the source.
	  • **older** Overwrite if file is older than the source.
	• **FileManifest=none** Generate a list of all transferred files information. *(none / text. Default: none.)*
	• **FileManifestPath=(path)** Specify the path to store the manifested file. *(text string, Default: blank)*
	• **FileCrypt=encrypt** Encrypt or decrypt files. Passphrase is required.
	• **RetryTimeout=(secs)** Specify the timeout duration in seconds, for a retry attempt. *(Default: blank)*
	• **SymbolicLinks=copy** Specify rule to handle symbolic links. Currently a beta feature. This option takes following values: (Default: follow)

Option	Description
	• **follow** Follow symbolic links and transfer the linked files. • **copy** Copy only the alias file. • **skip** Skip the symbolic links.
-U {1|2}	Priority when sharing physical or virtual bandwidth cap. 1 for higher priority, 2 for regular. *(Default: 2)*
-W *token-string*	Specify the token string for the transfer.
-@[*range-low*:*range-high*]	Transfer only part of a file. This option only works for downloading a single file, and does not support resuming. The argument to "-@" may omit either or both numbers, and the ":" delimiter. For example, *-@3000:6000* transfers bytes between positions 3000 to 6000; *-@1000:* transfers from 1000 to the end of the file; and *-@:1000* transfers from beginning to 1000.
-6	Enable IPv6 address support. When using IPv6, numeric host can be written inside brackets. For example, [2001:0:4137:9e50:201b:63d3:ba92:da] or [fe80::21b:21ff:fe1c:5072%eth1]

ascp HTTP Fallback Options

Option	Description
-y {0|1}	Enable HTTP Fallback transfer server when UDP connection fails. Set 1 to enable.
-j {0|1}	Encode all HTTP transfers as JPEG files. Set 1 to enable. 0 / 1. *(Default: 0)*
-Y *key-file*	The HTTPS transfer's key file name.
-I *certif-file*	The HTTPS certificate's file name.
-t *port*	Specify the port for HTTP Fallback Server.
-x *proxy-server*	Specify the proxy server address used by HTTP Fallback.

ascp Examples

Examples of initiating *fasp* file transfers using the ascp command.

This topic demonstrates the ascp command with the following examples:

1. Fair-policy transfer, without encryption

   Transfer with fair rate policy, with maximum rate 100 Mbps and minimum at 1 Mbps:

```
$ ascp -TQ -l 100m -m 1m /local-dir/files root@10.0.0.2:/remote-dir
```

2. Fixed-policy transfer, without encryption

   Transfer all files in \local-dir\files to 10.0.0.2 with target rate 100 Mbps and encryption OFF:

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

3. Specify an UDP port

   To perform a transfer with UDP port 42000:

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

4. Authenticate with public key

   To perform a transfer with public key authentication with key file *<home dir>/.ssh/asp1-key local-dir/files*:

```
$ ascp -T -l 10m -i ~/.ssh/asp1-key local-dir/files root@10.0.0.2:/remote-dir
```

5. Authenticate with a login that contains 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"
```

6. Transfer with a network shared location

   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/"
```

7. Parallel transfer on a multi-core 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
```

8. Use content protection

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

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

   Download from the server *10.0.0.2* and decrypt while transferring:

```
$ ASPERA_SCP_FILEPASS=secRet ascp -l 10m -o FileCrypt=decrypt root@10.0.0.2:/remote-
dir  /local-dir
```

If the password-protected file is downloaded without descrypting (file1.aspera-env, with aspera-env appended), on the local computer, descrypt the file as file1:

```
$ ASPERA_SCP_FILEPASS=secRet asunprotect -o file1 file1.aspera-env
```

Frequently-Asked Questions

This topic lists frequently-asked questions regarding ascp command:

1. How do I control the transfer speed?

   You can specify a transfer policy that determines how *fasp* transfer utilize the network resource, as well as maximum and minimum transfer rates where applicable.

   In *ascp* command, use the following flags to specify fixed, fair and trickle transfer policies:

Policy	Command template
Fixed	`-l target_rate`
Fair	`-Q -l target_rate -m min_rate`
Trickle	`-QQ -l target_rate -m min_rate`

2. What should I expect in terms of transfer speed? 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.

This test will typically occupy a majority of the network's bandwidth. It is recommended that this test be performed on a dedicated file transfer line or during a time of very low network activity.

   On the client machine, start a transfer with fixed policy. Start with a lower transfer rate and increase gradually toward the network bandwidth (e.g. 1m, 5m, 10m...). Monitor the transfer rate and make sure that it fulfills your bandwidth:

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

   To improve the transfer speed, you may also upgrade the related hardware components:

Component	Description
Hard disk	The I/O throughput, the disk bus architecture (e.g. 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 / fails to finish, it will resume the transfer without re-transferring the files?

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

   - **-k 0** Always retransfer the entire file.
   - **-k 1** Check file attributes and resume if they match.
   - **-k 2** Check file attributes and do a sparse file checksum; resume if they match.
   - **-k 3** Check file attributes and do a full file checksum; resume if they match.

4. How does Aspera handle symbolic links?

   ascp command follows symbolic links by default, whereas it is currently a beta feature. There is a **-o SymbolicLink** flag that offers handling options:

   - **-o SymbolicLinks=follow**: Follow symbolic links and transfer the linked files.
   - **-o SymbolicLinks=copy**: Copy only the alias file.
   - **-o SymbolicLinks=skip**: Skip the symbolic links.

5. What are my choices regarding Overwrite of files already at the destination?

   In ascp, you can specify the overwriting rule with the following flags:

   - **-o Overwrite=always**: Always overwrite the file.
   - **-o Overwrite=never**: Never overwrite the file.
   - **-o Overwrite=diff**: Overwrite if file is different from the source.
   - **-o Overwrite=older**: Overwrite if file is older than the source.

Creating SSH Keys

Create a key pair for your computer.

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.

If you are using this machine as a client to connect to other Aspera servers with public key authentication, you need to generate a key-pair for the selected user account. Follow these instructions:

1. Create .ssh folder in home directory

   Create a ".ssh" folder in your user account's home directory if it doesn't exist:

   ```
   $ mkdir /home/<user name>/.ssh
   ```

   Navigate into the .ssh folder and continue:

   ```
   $ cd <path-to-user-home-dir>/.ssh
   ```

2. Use ssh-keygen to generate SSH key

   Execute the following command in the ".ssh" folder. The program will prompt you the key-pair's file name, hit enter to use the default name **id_rsa**. For a passphrase, you can either enter a password, or press return twice to leave it blank:

   ```
   $ ssh-keygen -t rsa
   ```

3. Retrieve the public key file

   When created, the key-pair can be found in your home directory's ".ssh" folder (Assuming you generated the key with default name **id_rsa**):

   ```
   (user's home directory)/id_rsa.pub
   ```

   Provide your public key file to the administrator of the server you are connecting to.

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

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

   ```
   $ ascp -T -l 10M -m 1M  -i ~/.ssh/id_rsa  my/files  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 public key *~/.ssh/id_rsa*.

General Configuration Reference

The general transfer configuration options.

This section covers the general configuration options, which can be used for global, group, and user settings.

aspera.conf - Authorization

The configuration options in aspera.conf's <authorization/>.

This topic shows you how to modify aspera.conf's <authorization/> section in a Terminal.

1. Open aspera.conf

   /ifs/.ifsvar/aspera/etc/aspera.conf

   You can also find the configuration example in this path:

   /ifs/.ifsvar/aspera/etc/samples/aspera-everything.conf

2. Add or locate the <file_system /> section using a template

   Here is a template that includes all options:

```
<authorization>
   <encryption_type>aes-128</encryption_type> <!-- Token Encryption Cipher -->
   <encryption_key> </encryption_key>         <!-- Token Encryption Key -->
   <filename_hash> </filename_hash>           <!-- Token Filename Hash -->
   <life_seconds>1200</life_seconds>          <!-- Token Life -->
   <transfer>
      <in>
         <value>allow</allow>       <!-- Incoming Transfer -->
         <external_provider>
            <url>...</url>   <!-- Incoming External Provider URL -->
            <soap>...</soap> <!-- Incoming External Provider SOAP Action -->
         </external_provider>
      </in>
      <out>
         <value>allow</allow>       <!-- Outgoing Transfer -->
         <external_provider>
            <url>...</url>   <!-- Outgoing External Provider URL -->
            <soap>...</soap> <!-- Outgoing External Provider SOAP Action -->
         </external_provider>
      </out>
   </transfer>
```

```
</authorization>
```

3. Configuration options reference

This table explains all configuration options:

#	Field	Description	Values	Default
1	Token Encryption Cipher	The cipher used to generate encrypted authorization tokens.	• aes-128 • aes-192 • aes-256	aes-128
2	Token Encryption Key	This is the secret text phrase that will be used to authorize those transfers configured to require token. Token generation is part of the Aspera SDK. See the *Aspera Developer's Network* for more information.	text string	blank
3	Token Filename Hash	Which algorithm should filenames inside transfer tokens be hashed with. Use MD5 for backward compatibility.	• sha1 • MD5 • sha256	sha1
4	Token Life	Sets token expiration for users of web-based transfer applications.	positive integer	1200
5	Incoming Transfer	The default setting of **allow** enables users to transfer to this computer. Setting this to **deny** will prevent transfers to this computer. When set to **token**, only transfers initiated with valid tokens will be allowed to transfer to this computer. Token-based transfers are typically employed by web applications such as Faspex and require a Token Encryption Key.	• allow • deny • token	allow
6	Incoming External Provider URL	The value entered should be the URL of the external authorization provider for incoming transfers. The default empty setting disables external authorization. Aspera servers can be configured to check with an external authorization provider. This SOAP authorization mechanism can be useful	HTTP URL	blank

#	Field	Description	Values	Default
		to organizations requiring custom authorization rules.		
7	Incoming External Provider SOAP Action	The SOAP action required by the external authorization provider for incoming transfers. Required if External Authorization is enabled.	text string	blank
8	Outgoing Transfer	The default setting of **allow** enables users to transfer from this computer. Setting this to **deny** will prevent transfers from this computer. When set to **token**, only transfers initiated with valid tokens will be allowed to transfer from this computer. Token-based transfers are typically employed by web applications such as Faspex and require a Token Encryption Key.	• allow • deny • token	allow
9	Outgoing External Provider URL	The value entered should be the URL of the external authorization provider for outgoing transfers. The default empty setting disables external authorization. Aspera servers can be configured to check with an external authorization provider. This SOAP authorization mechanism can be useful to organizations requiring custom authorization rules.	HTTP URL	blank
10	Outgoing External Provider SOAP Action	The SOAP action required by the external authorization provider for outgoing transfers. Required if External Authorization is enabled.	text string	blank

4. Validate aspera.conf

   When you have finished updating aspera.conf, use this command to validate it:

```
$ /usr/local/aspera/bin/asuserdata -b -v -a
```

aspera.conf - Transfer

The configuration options in aspera.conf's <transfer/>.

This topic shows you how to modify aspera.conf's <transfer/> section in a Terminal.

1. Open aspera.conf

   /ifs/.ifsvar/aspera/etc/aspera.conf

   You can also find the configuration example in this path:

   /ifs/.ifsvar/aspera/etc/samples/aspera-everything.conf

2. Add or locate the <transfer /> section using a template

   Here is a template that includes all options:

```
<transfer>
 <protocol_options>
    <bind_ip_address></bind_ip_address>        <!--Bind IP Address-->
    <bind_udp_port>33001</bind_udp_port>       <!--Bind UDP Port-->
    <disable_batching>false</disable_batching> <!--Disable Packet Batching-->
 </protocol_options>
 <encryption>
    <content_protection_strong_pass_required>  <!--Strong Password Required for
 Content Protection-->
       false
    </content_protection_strong_pass_required>
    <content_protection_required>              <!--Content Protection Required-->
       false
    </content_protection_required>
    <allowed_cipher>any<allowed_cipher>        <!--Encryption Allowed-->
    <fips_mode>false</fips_mode>               <!--Transfer in FIPS-140-2-certified
 encryption mode-->
 </encryption>
 <in>
    <bandwidth>
       <aggregate>
          <trunk_id>109</trunk_id>        <!-- Incoming VLink ID -->
       </aggregate>
       <flow>
          <target_rate>
             <cap></cap>                  <!-- Incoming Target Rate Cap -->
             <default>10000</default>     <!-- Incoming Target Rate Default -->
             <lock>false</lock>           <!-- Incoming Target Rate Lock -->
          </target_rate>
```

```
         <min_rate>
            <cap></cap>                  <!-- Incoming Minimum Rate Cap -->
            <default></default>          <!-- Incoming Minimum Rate Default -->
            <lock>false</lock>           <!-- Incoming Minimum Rate Lock -->
         </min_rate>
         <policy>
            <cap></cap>                  <!-- Incoming Policy Allowed -->
            <default></default>          <!-- Incoming Policy Default -->
            <lock>false</lock>           <!-- Incoming Policy Lock -->
         </policy>
         <priority>
            <cap></cap>                  <!-- Incoming Priority Allowed -->
            <default></default>          <!-- Incoming Priority Default -->
            <lock>false</lock>           <!-- Incoming Priority Lock -->
         </priority>
      </flow>
   </bandwidth>
</in>
<out>
   <bandwidth>
      <aggregate>
         <trunk_id>109</trunk_id>        <!-- Outgoing VLink ID -->
      </aggregate>
      <flow>
         <target_rate>
            <cap></cap>                  <!-- Outgoing Target Rate Cap -->
            <default>10000</default>     <!-- Outgoing Target Rate Default -->
            <lock>false</lock>           <!-- Outgoing Target Rate Lock -->
         </target_rate>
         <min_rate>
            <cap></cap>                  <!-- Outgoing Minimum Rate Cap -->
            <default>0</default>         <!-- Outgoing Minimum Rate Default -->
            <lock>false</lock>           <!-- Outgoing Minimum Rate Lock -->
         </min_rate>
         <policy>
            <cap></cap>                  <!-- Outgoing Policy Allowed -->
            <default></default>          <!-- Outgoing Policy Default -->
            <lock>false</lock>           <!-- Outgoing Policy Lock -->
         </policy>
         <priority>
            <cap></cap>                  <!-- Outgoing Priority Allowed -->
            <default></default>          <!-- Outgoing Priority Default -->
```

```
            <lock>false</lock>          <!-- Outgoing Priority Lock -->
         </priority>
      </flow>
   </bandwidth>
 </out>
</transfer>
```

3. Configuration options reference

   This table explains all configuration options:

#	Field	Description	Values	Default
1	Bind IP Address	Specify an IP address for server-side ascp to bind its UDP connection. If a valid IP address is given, ascp sends and receives UDP packets ONLY on the interface corresponding to that IP address.	valid IPv4 address	blank
2	Bind UDP Port	Prevent the client-side ascp process from using the specified UDP port.	integer between 1 and 65535	33001
3	Disable Packet Batching	When set to true, send data packets back to back (no sending a batch of packets). This results in smoother data traffic at a cost of higher CPU usage.	• true • false	false
4	Strong Password Required for Content Encryption	When set to true, require the password for content encryption to contain at least one letter, one number, and one symbol.	• true • false	false
5	Content Protection Required	When set to true, users will be required on upload to enter a password to encrypt the files on the server.	• true • false	false
6	Encryption Allowed	Describes the type of transfer encryption accepted by this computer. When set to any the computer allows both encrypted and non-encrypted transfers. When set to none the computer restricts transfers to non-encrypted transfers only. When set to aes-128 the computer restricts transfers to encrypted transfers only.	• any • none • aes-128	any
7	Transfer in FIPS-140-2-certified encryption mode	When set to **true**, ascp will use a FIPS 140-2-certified encryption module. Note:	• true	false

#	Field	Description	Values	Default
		When this feature is enabled, transfer start is delayed while the FIPS module is verified.	• false	
8	Incoming VLink ID	The value sets Vlink ID for incoming transfers. Vlinks are a mechanism to define aggregate transfer policies. The default setting of 0 disables Vlinks. One Vlink—the virtual equivalent of a network trunk—represents a bandwidth allowance that may be allocated to a node, group, or user. Vlink ID are defined in each Vlink created in Aspera Console. The Vlink ID is a unique numeric identifier.	pre-defined value	0
9	Incoming Target Rate Cap	The value sets the Target Rate Cap for incoming transfers. The Target Rate Cap is the maximum target rate that a transfer can request, in kilobits per second. No transfer may be adjusted above this setting, at any time. The default setting of Unlimited signifies no Target Rate Cap. Clients requesting transfers with initial rates above the Target Rate Cap will be denied.	positive integer	unlimited
10	Incoming Target Rate Default	This value represents the initial rate for incoming transfers, in kilobits per second. Users may be able to modify this rate in real time as allowed by the software in use. This setting is not relevant to transfers with a Policy of Fixed.	positive integer	10000
11	Incoming Target Rate Lock	After an incoming transfer is started, its target rate may be modified in real time. The default setting of false gives users the ability to adjust the transfer rate. A setting of true prevents real-time modification of the transfer rate.	• true • false	false
12	Incoming Minimum Rate Cap	The value sets the Minimum Rate Cap for incoming transfers. The Minimum Rate Cap is a level specified	positive integer	unlimited

#	Field	Description	Values	Default
		in kilobits per second, below which an incoming transfer will not slow, despite network congestion or physical network availability. The default value of Unlimited effectively turns off the Minimum Rate Cap.		
13	Incoming Minimum Rate Default	This value represents the initial minimum rate for incoming transfers, in kilobits per second. Users may be able to modify this rate in real time as allowed by the software in use. This setting is not relevant to transfers with a Policy of Fixed.	positive integer	0
14	Incoming Minimum Rate Lock	After an incoming transfer is started, its minimum rate may be modified in real time. The default setting of false gives users the ability to adjust the transfer's minimum rate. A setting of true prevents real-time modification of the transfer rate. This setting is not relevant to transfers with a Policy of Fixed.	• true • false	false
15	Incoming Policy Allowed	The value chosen sets the allowed Bandwidth Policy for incoming transfers. Aspera transfers use fixed, high, fair and low policies to accommodate network-sharing requirements. When set to any, the server will not deny any transfer based on policy setting. When set to high, transfers with a Policy of high and less aggressive transfer policies (e.g. fair or low) will be permitted. Fixed transfers will be denied. When set to low, only transfers with a Bandwidth Policy of low will be allowed.	• any • high • fair • low	any
16	Incoming Policy Default	The value chosen sets the default Bandwidth Policy for incoming transfers. The default policy value may be overridden by client applications initiating transfers.	• fixed • high • fair • low	fair

#	Field	Description	Values	Default
17	Incoming Policy Lock	After an incoming transfer is started, its Policy may be modified in real time. The default setting of false gives users the ability to adjust the transfer's Policy. A setting of true prevents real-time modification of the Policy.	• true • false	false
18	Incoming Priority Allowed	The highest priority your client can request. Use the value 0 to unset this option; 1 to allow high priority, 2 to enforce normal priority.	• 0 • 1 • 2	1
19	Incoming Priority Default	The initial priority setting. Use the value 0 to unset this option, 1 to allow high priority; 2 to enforce normal priority	• 0 • 1 • 2	2
20	Incoming Priority Lock	To disallow your clients change the priority, set the value to true	• true • false	false
21	Outgoing VLink ID	The value sets Vlink ID for outgoing transfers. Vlinks are a mechanism to define aggregate transfer policies. The default setting of 0 disables Vlinks. One Vlink—the virtual equivalent of a network trunk—represents a bandwidth allowance that may be allocated to a node, group, or user. Vlink ID are defined in each Vlink created in Aspera Console. Vlink ID is a unique numeric identifier.	pre-defined value	0
22	Outgoing Target Rate Cap	The value sets the Target Rate Cap for outgoing transfers. The Target Rate Cap is the maximum target rate that a transfer can request, in kilobits per second. No transfer may be adjusted above this setting, at any time. The default setting of Unlimited signifies no Target Rate Cap. Clients requesting transfers with initial rates above the Target Rate Cap will be denied.	positive integer	unlimited

#	Field	Description	Values	Default
23	Outgoing Target Rate Default	This value represents the initial rate for outgoing transfers, in kilobits per second. Users may be able to modify this rate in real time as allowed by the software in use. This setting is not relevant to transfers with a Policy of Fixed.	positive integer	10000
24	Outgoing Target Rate Lock	After an outgoing transfer is started, its target rate may be modified in real time. The default setting of false gives users the ability to adjust the transfer rate. A setting of true prevents real-time modification of the transfer rate.	• true • false	false
25	Outgoing Minimum Rate Cap	The value sets the Minimum Rate Cap for outgoing transfers. The Minimum Rate Cap is a level specified in kilobits per second, below which an incoming transfer will not slow, despite network congestion or physical network availability. The default value of Unlimited effectively turns off the Minimum Rate Cap.	positive integer	unlimited
26	Outgoing Minimum Rate Default	This value represents the initial minimum rate for outgoing transfers, in kilobits per second. Users may be able to modify this rate in real time as allowed by the software in use. This setting is not relevant to transfers with a Policy of Fixed.	positive integer	0
27	Outgoing Minimum Rate Lock	After an outgoing transfer is started, its minimum rate may be modified in real time. The default setting of false gives users the ability to adjust the transfer's minimum rate. A setting of true prevents real-time modification of the transfer rate. This setting is not relevant to transfers with a Policy of Fixed.	• true • false	false
28	Outgoing Policy Allowed	The value chosen sets the allowed Bandwidth Policy for outgoing transfers.	• any • high	any

#	Field	Description	Values	Default
		Aspera transfers use fixed, high, fair and low policies to accommodate network-sharing requirements. When set to any, the server will not deny any transfer based on policy setting. When set to high, transfers with a Policy of high and less aggressive transfer policies (e.g. fair or low) will be permitted. Fixed transfers will be denied. When set to low, only transfers with a Bandwidth Policy of low will be allowed.	• fair • low	
29	Outgoing Policy Default	The value chosen sets the default Bandwidth Policy for outgoing transfers. The default policy value may be overridden by client applications initiating transfers.	• fixed • high • fair • low	fair
30	Outgoing Policy Lock	After an outgoing transfer is started, its Policy may be modified in real time. The default setting of false gives users the ability to adjust the transfer's Policy. A setting of true prevents real-time modification of the Policy.	• true • false	false
31	Outgoing Priority Allowed	The highest priority your client can request. Use the value 0 to unset this option; 1 to allow high priority, 2 to enforce normal priority.	• 0 • 1 • 2	1
32	Outgoing Priority Default	The initial priority setting. Use the value 0 to unset this option, 1 to allow high priority; 2 to enforce normal priority.	• 0 • 1 • 2	2
33	Outgoing Priority Lock	To disallow your clients change the priority, set the value to true	• true • false	false

4. Validate aspera.conf

When you have finished updating aspera.conf, use this command to validate it:

```
$ /usr/local/aspera/bin/asuserdata -b -v -a
```

aspera.conf - File System

The configuration options in aspera.conf's <file_system/>.

This topic shows you how to modify aspera.conf's <file_system/> section in a Terminal.

1. Open aspera.conf

   /ifs/.ifsvar/aspera/etc/aspera.conf

   You can also find the configuration example in this path:

   /ifs/.ifsvar/aspera/etc/samples/aspera-everything.conf

2. Add or locate the <file_system /> section using a template

   Here is a template that includes all options:

```
<file_system>
   <access>
      <paths>
         <path>
            <absolute>/sandbox/$(name)</absolute>      <!-- Absolute Path -->
            <read_allowed>true</read_allowed>          <!-- Read Allowed -->
            <write_allowed>true</write_allowed>        <!-- Write Allowed -->
            <dir_allowed>true</dir_allowed>            <!-- Browse Allowed -->
         <path>
      <paths>
   </access>
   <read_block_size>0</read_block_size>                  <!-- Read Block Size -->
   <write_block_size>0</write_block_size>                <!-- Write Block Size -->
   <use_file_cache>true</use_file_cache>                 <!-- Use File Cache -->
   <max_file_cache_buffer>0</max_file_cache_buffer>      <!-- Max File Cache Buffer-->
   <resume_suffix>.aspx</resume_suffix>                  <!-- Resume Suffix -->
   <preserve_attributes> </preserve_attributes>          <!-- Preserve Attributes -->
   <overwrite>allow</overwrite>                          <!-- Overwrite -->
   <file_manifest>disable</file_manifest>                <!-- File Manifest -->
   <file_manifest_path> </file_manifest_path>            <!-- File Manifest Path -->
   <pre_calculate_job_size>any</pre_calculate_job_size><!-- Pre-Calculate Job Size
 -->
   <storage_rc>
      <adaptive>true</adaptive>                          <!-- Storage Rate Control -->
   </storage_rc>
   <file_create_mode> </file_create_mode>                <!-- File Create Mode -->
```

```
   <file_create_grant_mask>644</file_create_grant_mask><!-- File Create Grant Mask
 -->
   <directory_create_mode> </directory_create_mode>    <!-- Directory Create Mode --
>
   <directory_create_grant_mask>755</directory_create_grant_mask> <!-- Directory
 Create Grant Mask -->
   <excludes>    <!-- Exclude Pattern -->
      <exclude></exclude>
      <exclude></exclude>
      ...
   </excludes>
</file_system>
```

3. Configuration options reference

   This table explains all configuration options:

#	Field	Description	Values	Default
1	Absolute Path	The Absolute Path describes the area of the file system that is accessible by Aspera users. The default empty value gives users access to the entire file system.	file path	blank
2	Read Allowed	Setting this to true allows users to transfer from the designated area of the file system as specified by the Absolute Path value.	• true • false	blank
3	Write Allowed	Setting this to true allows users to transfer to the designated area of the file system as specified by the Absolute Path value.	• true • false	blank
4	Browse Allowed	Setting this to true allows users to browse the directory.	• true • false	blank
5	Read Block Size	This is a performance tuning parameter for an Aspera sender. It represents the number of bytes an Asper sender reads at a time from the source disk drive. Only takes effect when server is sender. The default of 0 will cause the Aspera sender to use its default internal buffer	positive integer	0

#	Field	Description	Values	Default
		size, which may be different for different operating systems.		
6	Write Block Size	This is a performance tuning parameter for an Aspera receiver. Number of bytes an ascp receiver writes data at a time onto disk drive. Only takes effect when server is receiver. The default of 0 will cause the Aspera sender to use its default internal buffer size, which may be different for different operating systems.	positive integer	0
7	Use File Cache	This is a performance tuning parameter for an Aspera receiver. Enable or disable per-file memory caching at the data receiver. File level memory caching improves data write speed on Windows platforms in particular, but will use more memory. We suggest using a file cache on systems that are transferring data at speeds close to the performance of their storage device, and disable it for system with very high concurrency (because memory utilization will grow with the number of concurrent transfers).	• true • false	true
8	Max File Cache Buffer	This is a performance tuning parameter for an Aspera receiver. This value corresponds to the maximal size allocated for per-file memory cache (see Use File Cache). Unit is bytes. The default of 0 will cause the Aspera receiver to use its internal buffer size, which may be different for different operating systems.	positive integer	0
9	Resume Suffix	File name extension for temporary metadata files used for resuming incomplete transfers. Each data file in progress will have a corresponding metadata file with the same name plus the resume suffix specified by the receiver. Metadata files in the source of a	text string	.aspx

#	Field	Description	Values	Default
		directory transfer are skipped if they end with the sender's resume suffix.		
10	Preserve Attributes	Configure file creation policy. When set to none, do not preserve the timestamp of source files. When set to times, preserve the timestamp of the source files at destination.	• none • times	blank
11	Overwrite	Overwrite is an Aspera server setting that determines whether Aspera clients are allowed to overwrite files on the server. By default it is set to allow, meaning that clients uploading files to the servers will be allowed to overwrite existing files as long as file permissions allow that action. If set to deny, clients uploading files to the server will not be able to overwrite existing files, regardless of file permissions.	• allow • deny	allow
12	File Manifest	When set to text a text file "receipt" of all files within each transfer session is generated. If set to disable, no File Manifest is created. The file manifest is a file containing a list of everything that was transferred in a given transfer session. The filename of the File Manifest itself is automatically generated based on the transfer session's unique ID. The location where each manifest is written is specified by the File Manifest Path value. If no File Manifest Path is specified, the file will be generated under the destination path at the receiver, and under the first source path at the sender.	• text • disable	none
13	File Manifest Path	Specify the location to store manifest files. Can be an absolute path or a path relative to the transfer user's home.	text string	blank
14	Pre-Calculate Job Size	Configure the policy of calculating total job size before data transfer. If set	• any • yes	any

#	Field	Description	Values	Default
		to any, follow client configurations (-o PreCalculateJobSize={yes|no}). If set to no, disable calculating job size before transferring. If set to yes, enable calculating job size before transferring.	• no	
15	Storage Rate Control	Enable/Disable disk rate control. When enabled, adjust transfer rate according to the speed of receiving I/O storage, if it becomes a bottleneck.	• true • false	true
16	File Create Mode	Specify file creation mode (permissions). If specified, create files with these permissions (for example 0755), irrespective of File Create Grant Mask and permissions of the file on the source computer. Only takes effect when the server is a non-Windows receiver.	positive integer (octal)	undefined
17	File Create Grant Mask	Used to determine mode for newly created files if File Create Mode is not specified. If specified, file modes will be set to their original modes plus the Grant Mask values. Only takes effect when the server is a non-Windows receiver and when File Create Mode is not specified.	positive integer (octal)	644
18	Directory Create Mode	Specify directory creation mode (permissions). If specified, create directories with these permissions irrespective of Directory Create Grant Mask and permissions of the directory on the source computer. Only takes effect when the server is a non-Windows receiver.	positive integer (octal)	undefined
19	Directory Create Grant Mask	Used to determine mode for newly created directories if Directory Create Mode is not specified. If specified, directory modes will be set to their original modes plus the Grant Mask values. Only takes effect when the server	positive integer (octal)	755

#	Field	Description	Values	Default
		is a non-Windows receiver and when Directory Create Mode is not specified.		
20	Exclude Pattern	Exclude files or directories with the specified pattern in the transfer. Use multiple <exclude/> for more exclusion patterns. Two symbols can be used in the setting of patterns: • ***** (Asterisk) Represents zero to many characters in a string, for example, **.tmp* matches *.tmp* and *abcde.tmp*. • **?** (Question Mark) Represents one character, for example, *t?p* matches tmp but not temp.	text string	blank

4. Validate aspera.conf

   When you have finished updating aspera.conf, use this command to validate it:

```
$ /usr/local/aspera/bin/asuserdata -b -v -a
```

Appendix

fasp Transfer Policies

The character of the *fasp* transfer policies.

The transfer policy and speed determine how you utilize the network resource for *fasp* file transfers. Here is the description of all transfer policies:

Policy	Description
Fixed	*fasp* attempts to transfer at the specified target rate, regardless of the actual network capacity. This policy transfers at a constant rate and finishes in a guaranteed time. This policy will typically occupy a majority of the network's bandwidth, and is not recommended in most file transfer scenarios. In this mode, a maximum (target) rate value is required.
High	*fasp* monitors the network and adjusts the transfer rate to fully utilize the available bandwidth up to the maximum rate. When congestion occurs, a *fasp* session with high policy transfers at a rate twice of a session with fair policy. In this mode, both the maximum (target) and the minimum transfer rates are required.
Fair	*fasp* monitors the network and adjusts the transfer rate to fully utilize the available bandwidth up to the maximum rate. When other types of traffic builds up and congestion occurs, *fasp* shares bandwidth with other traffic fairly by transferring at an even rate. In this mode, both the maximum (target) and the minimum transfer rates are required.
Low	Similar to Fair mode, the low policy uses the available bandwidth up to the maximum rate, but much less aggressive when sharing bandwidth with other network traffic. When congestion builds up, the transfer rate is decreased all the way down to the minimum rate, until other traffic retreats.

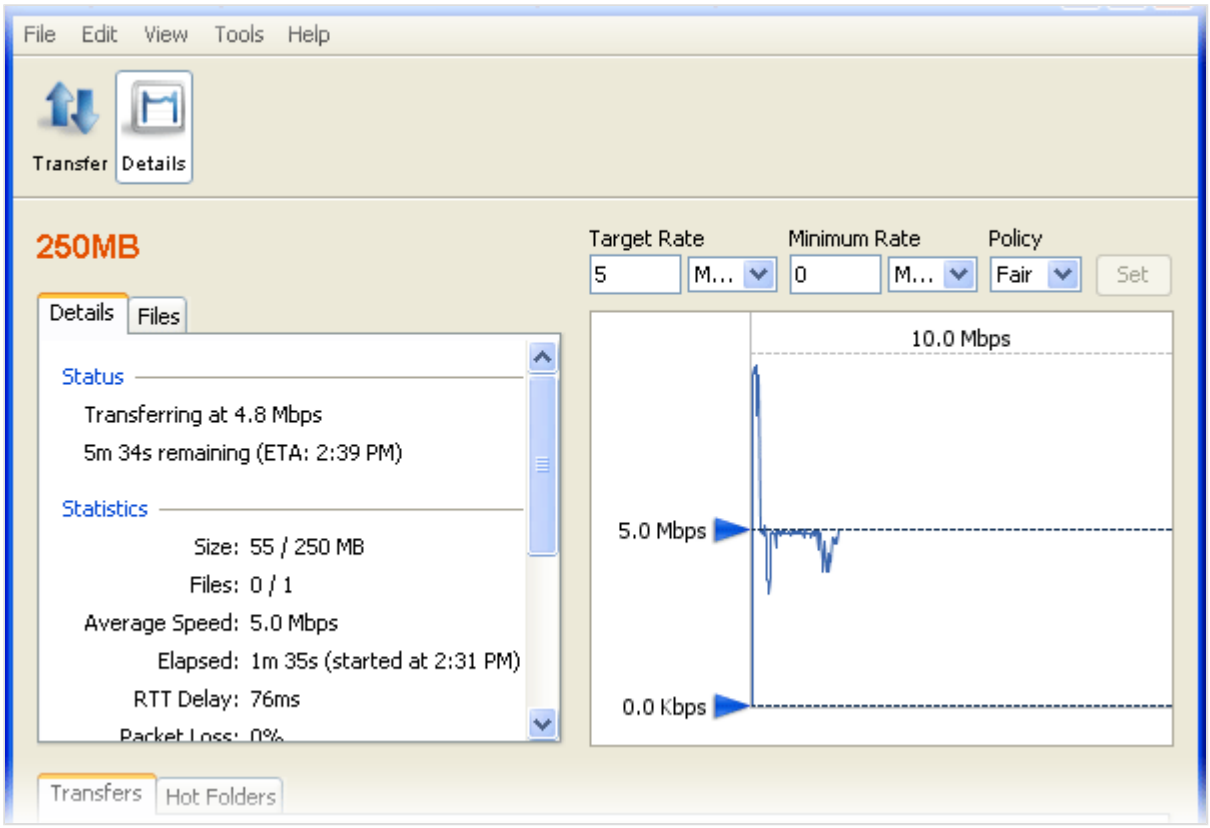
Optimizing Transfer Performance

Tips about testing and improving your computer's transfer performance.

To verify that your system's *fasp* transfer can fulfill the maximum bandwidth capacity, prepare a client machine to connect to this computer, and do the following tests:

1. Start a transfer with Fair transfer policy

   On the client machine, open the user interface and start a transfer. Go to the **Details** to open the Transfer Monitor.

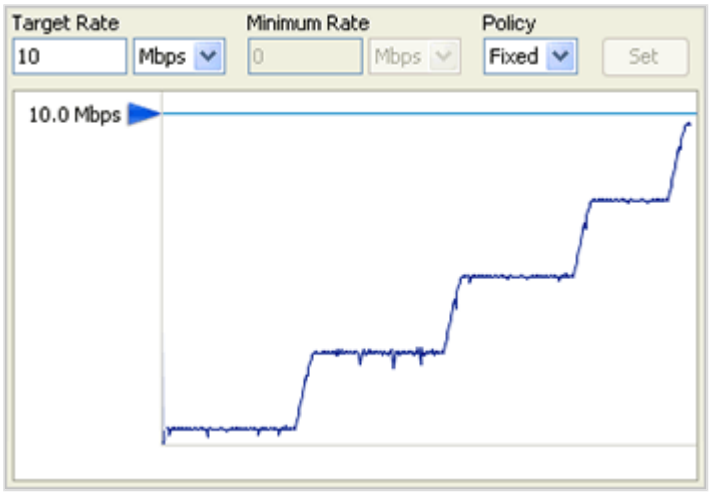


To leave more network resource for other high-priority traffics, use **Fair** policy and adjust the Target Rate and Minimum Rate rate by sliding the arrows or enter the values.

2. Test the maximum bandwidth

This test will typically occupy a majority of the network's bandwidth. It is recommended that this test be performed on a dedicated file transfer line or during a time of very low network activity.

Use **Fixed** policy for the maximum transfer speed. Start with a lower transfer rate and increase gradually toward the network bandwidth.



To improve the transfer speed, you may also upgrade the related hardware components:

Component	Description
Hard disk	The I/O throughput, the disk bus architecture (e.g. 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.

Log Files

Locate the log files related to the Aspera product.

The log file includes detailed transfer information and can be useful for review and support request.

The file can be found in the location:

```
/var/log/aspera.log
```

Updating Product License

Update your product license.

To update the license, open the following file with write permission, replace the existing license key string with the new one:

```
/ifs/.ifsvar/aspera/etc/aspera-license
```

When finished, save and close the file. Use this command to verify the new license info:

```
$ ascp -A
```

Evaluating SSH Server Security

Evaluate your SSH Server configuration and help preventing potential security risks.

Aspera transfer products use SSH for connection authentication. For security purposes, it is strongly recommended that you review SSH Server log and configure it when necessary. This topic shows you how to do it to avoid certain types of attacks.

1. Review the SSH connection log

   It is recommended to review your SSH log periodically and see if you are being attacked. Locate and open your syslog, for example, **/var/log/auth.log** or **/var/log/secure**. Depending on your system configuration, syslog's path and file name may vary.

   Look for invalid users in the log, especially a series of login attempts with common user names from the same address, usually in alphabetical order. For example:

   ```
   ...
   Mar 10 18:48:02 sku sshd[1496]: Failed password for invalid user alex from 1.2.3.4
    port 1585 ssh2
   ...
   Mar 14 23:25:52 sku sshd[1496]: Failed password for invalid user alice from 1.2.3.4
    port 1585 ssh2
   ...
   ```

   If you have identified attacks:

   - Follow SSH Server setup recommendations in this topic.
   - Report attacker to your ISP's abuse email (e.g. abuse@your-isp).

2. Locate the SSH server configuration file

   Open the SSH server configuration file with a text editor. (Depending on your system configuration, the file location varies.)

   ```
   /etc/mcp/templates/sshd_config
   ```

3. Review the authentication methods

   To allow public key authentication, add or uncomment *PubkeyAuthentication yes*; To allow password authentication, add or uncomment *PasswordAuthentication yes*. Here is a configuration example:

   ```
   ...
   PubkeyAuthentication yes
   PasswordAuthentication yes
   ```

```
...
```

4. Change SSH ports

   It is recommended to use an alternative SSH port (TCP/32768 or higher, to be outside of usual scan range, such as TCP/33001) for *fasp* transfers, instead of the default (TCP/22), to avoid security risks. For example, to remove TCP/22 and use only TCP/33001, comment-out *Port 22* and add *Port 33001*:

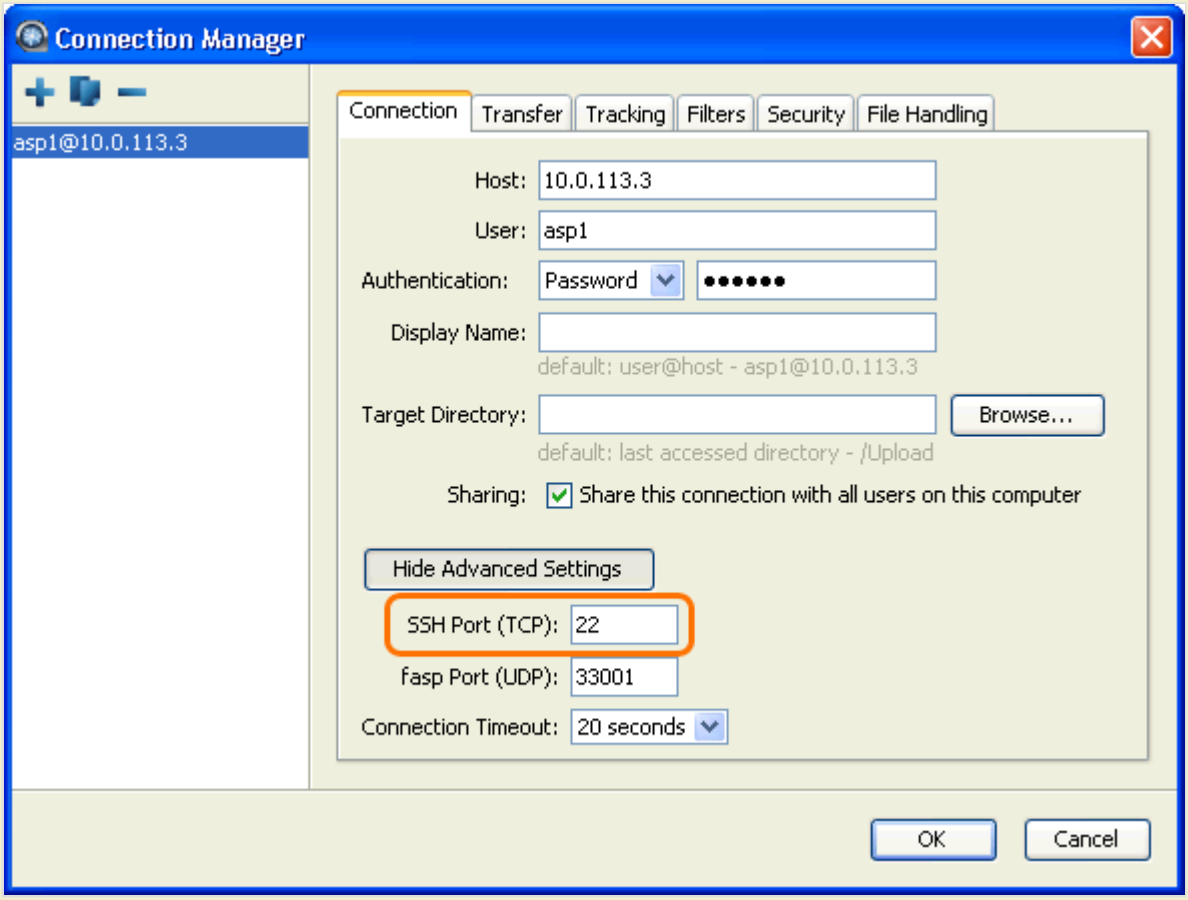
```
...
#Port 22
Port 33001
...
```

   To allow connections from both TCP/22 and TCP/33001, use this setting:

```
...
Port 22
Port 33001
...
```

Disabling the default SSH connection port (TCP/22) may affect your clients. When you change it, make sure to advise your clients of the new port number. Here are the basic instructions about specifying the SSH port for the *fasp* file transfers.

To specify the SSH port on the client side, click **Connections** on the main window, and select the entry for your computer. Under the **Connection** tab, click **Show Advanced Settings** and enter the SSH port number in the *SSH Port (TCP)* field.



If the client is connecting using the ascp command, specify the SSH port with **-P** (capital P) flag:

```
ascp -P 33001 ...
```

5. Disable Non-admin tunneling

This feature requires SSH server version 5.1 or higher.

It is recommended that you disable SSH tunneling to avoid potential attacks, and only allow tunneling from group users if this computer is managed by Aspera Console. To do so, add the following lines at the end of the configuration file:

```
...
AllowTcpForwarding no
Match Group root
AllowTcpForwarding yes
```

6. Restart SSH Server to apply new settings

When finished updating the SSH Server configuration, restart your SSH Server to apply these settings. For example, use these commands:

```
$ sudo /etc/rc.d/sshd reload
```

Uninstall

How to uninstall the Aspera product from your computer.

To uninstall Enterprise Server for Isilon, use this command:

```
$ isi_for_array -s -q /usr/local/aspera/var/uninstall.sh
```

Troubleshooting

Clients Can't Establish Connection

Troubleshoot the problem that your clients cannot connect to your

The following diagram shows the troubleshooting procedure if clients can't establish a *fasp* transfer connection to your Enterprise Server. Follow the instructions to identify and resolve problems:

1. Test SSH ports

   To verify the SSH connection port, on the client machine, open a Terminal or a Command Prompt, and use the **telnet** command to test it. For example, to test connection to a computer (10.0.1.1) through a port (TCP/33001), use this command:

   ```
   telnet 10.0.1.1 33001
   ```

   If the client cannot establish connections to your Enterprise Server, verify the port number and the firewall configuration on your Enterprise Server machine.

2. Test UDP ports

   If you can establish a SSH connection but not a *fasp* file transfer, there might be a firewall blockage of *fasp*'s UDP port. This test verifies the UDP connection.

   You can use our UDP test tools to test the UDP connections between a client and your Enterprise Server. First, download both UDP sending and receiving commands on your Enterprise Server:

Tool	Download Link
Isilon Receiving	Contact *Technical Support* on page 81.
Isilon Sending	Contact *Technical Support* on page 81.

   For clients or other platforms, locate and download both send and receive tools from this page:

   http://download.asperasoft.com/download/sw/tools/udp/

   On your Enterprise Server machine, execute the receiving tool with your UDP port specified. In this example, the Enterprise Server machine uses UDP/33001:

   You should see the following message:

   ```
   Starting reception on UDP port 33001
   ```

On the client machine, execute the sending tool with both the Enterprise Server machine's address and the port number: (Address: 10.0.1.1, UDP/33001)

You should see the following message on the client side, showing that the client is sending data:

```
Sending UDP datagram size 1460, rate=200 Kbps (ipd=57031 usec)
Set sock nonblocking successful
dgrams sent=42 drop=0
dgrams sent=95 drop=0
...
```

If the client can access your Enterprise Server, you should see a series of reception message on your Enterprise Server machine. For example:

```
dgrams rcvd=22 lost=0 (0%) interval rcvd=22 lost=0 (0%) rate=128 Kbps
dgrams rcvd=57 lost=0 (0%) interval rcvd=22 lost=0 (0%) rate=204 Kbps
...
```

Test all UDP ports for *fasp* connections if more than one are being prepared. If your Enterprise Server machine doesn't return these messages, review your firewall settings to resolve the blockage. To test the UDP transfer from your Enterprise Server to a client, run the receiving tool on the client, and execute the sending tool to connect to the client.

If you still encounter connection problems after going through these steps, contact *Technical Support* on page 81.

Technical Support

For further assistance, you may contact us through the following methods:

Contact Info	
Email	support@asperasoft.com
Phone	+1 (510) 849-2386
Request Form	*http://asperasoft.com/support/*

The technical support service hours:

Support Type	Hour (Pacific Standard Time, GMT-8)
Standard	8:00am – 6:00pm
Premium	8:00am – 12:00am

We are closed on the following days:

Support Unavailable Date	
Weekends	Saturday, Sunday
Aspera Holidays (2010)	Jan 1, Jan 18, Feb 15, May 31, Jul 5, Sept 6, Nov 25, Nov 26, Dec 24, Dec 31

Legal Notice

Aspera, the Aspera logo, and *fasp* transfer technology, are trademarks of Aspera Inc., registered in the United States.

Aspera Connect Server, Aspera Enterprise Server, Aspera Point-to-Point, Aspera Client, Aspera Connect, Aspera Cargo, Aspera Console, and Faspex are trademarks of Aspera Inc.

All other trademarks mentioned in this document are the property of their respective owners. Third-party products mentioned 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.