IBM

Contents

Introduction

Console is a web-based application which allows users to centrally manage, monitor and control Aspera servers (nodes) and transfers. Console offers the following features:

Feature	Description
Transfer monitoring and control	View transfers, pause / resume / cancel, change transfer rates.
Transfer Initiation	Initiate and schedule transfer jobs remotely.
Node Configuration	An administrator can configure all nodes directly from Console such for options such as bandwidth, priority, and encryption.
Email Notification	Notify users of transfer events with customizable messages.
Reporting	Create detail and summary reports of transfer activity.
Role-based access control	Manage what transfers are visible and controllable to Console users with security groups.

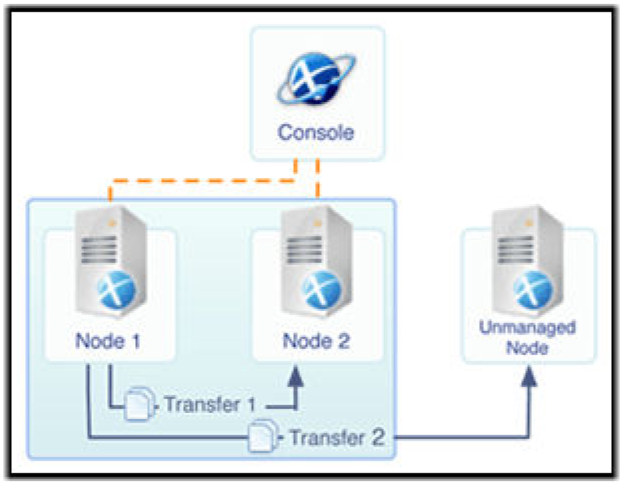
Console Nodes

Aspera servers can be added to Console as managed nodes or unmanaged nodes.

Managed nodes can be configured and their transfer activity managed from the Console UI through an SSH connection. The transfer activity of managed nodes is logged to the Console database. For nodes that run an Aspera server application version 3.4.6 or newer ("regular" nodes), Console makes REST calls to the Node API to pull transfer details into console as well as to start and control transfers. For nodes than run older versions of Aspera server applications ("legacy" nodes), Console processes data that gets pushed to the database by the node. Console also communicates with legacy nodes via SOAP calls to start and control transfers and to check for cases where nodes failed to log the end of a transfer.

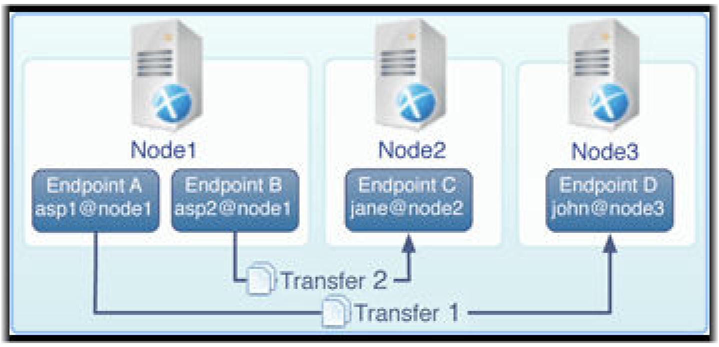
Unmanged nodes are not under control of Console. These can be used as transfer destinations, but only transfer activity with managed nodes is reported to Console.

The following figure shows the relationship between Console, two managed nodes (Node 1 and Node 2) and an unmanaged node. All transfer activity on the managed nodes, such as Transfer 1, can be monitored and controlled. In contrast, Console is only aware of transfers that are between the unmanaged node and a managed node (for example, Transfer 2 between Node 1 and the unmanaged node).



Console Endpoints

An endpoint is an individual user account on a node (managed or unmanaged) that can perform Aspera transfers without requiring the user to enter credentials in the Console UI. You can have one or multiple endpoints on a node depending on your business needs. The following figure shows a simple endpoint arrangement. Node 1 has two endpoints, Endpoint A (asp1@node1) and Endpoint B (asp2@node1), and Node 2 and Node 3 have one endpoint each (Endpoint C and Endpoint D, respectively). A Console user can run transfers between endpoints from the Consoel UI. For example, Transfer 1 between Endpoint A and Endpoint D, and Transfer 2 between Endpoint B and Endpoint C. Both transfers originate from Node 1, but from different Endpoints and have different destination Endpoints.



Installing Console

Firewall Requirements

Firewall (on the Console Machine)

Open the following ports on the Console machine:

- For the Web UI, allow inbound connections for HTTP or HTTPS Web access (for example, TCP/80, TCP/443).
- Allow outbound connections for SSH (to be used for node administration) on a non-default, configurable TCP port (for example, TCP/33001).
- Allow an outbound connection for TCP/9092 to allow Console to connect with nodes through the Node API
- Allow an outbound connection for TCP/40001 and an inbound connection for TCP/4406 to allow Console to connect with legacy nodes.

Firewall (on the Node Machines)

- To ensure that your server is secure, Aspera strongly recommends allowing inbound connections for SSH on TCP/33001 (or on another non-default, configurable TCP port), and disallowing inbound connections on TCP/22. If you have a legacy customer base using TCP/22, you can allow inbound connections on both ports. For details on securing your individual Aspera transfer server product, review the corresponding user manuals.
- Allow inbound connections for FASP transfers, which use UDP/33001 by default, although the server may also choose to run FASP transfers on another port.
- For current nodes and legacy nodes that have been converted to current nodes, allow an inbound connection on TCP 9092.
- For legacy nodes (unconverted), allow an inbound connection for Aspera Central (for example, TCP/40001).
- For legacy nodes (unconverted), allow an outbound connection for logging to Console on TCP/4406.

Note: No servers are listening on UDP ports.

When an Aspera client initiates a transfer, the client opens an SSH session to the SSH server on the designated TCP port and negotiates the UDP port over which the data transfer will occur.

For Aspera servers that have multiple concurrent clients, the Windows operating system does not allow Aspera's FASP protocol to reuse the same UDP port for multiple connections. Thus, if you have multiple concurrent clients and your Aspera server runs on Windows, then you must allow inbound connections on a range of UDP ports, where the range of ports is equal to the maximum number of concurrent FASP transfers expected. These UDP ports should be opened incrementally from the base port, which is UDP/33001, by default. For example, to allow 10 concurrent FASP transfers, allow inbound traffic from UDP/33001 to UDP/33010.

Data Storage Recommendation

Console generates data for every transfer session. Plan the size-growth of your databae depending on the number of transfer sessions each day.

In terms of planning for the size growth of the database, the per-file records generate 1-2KB per file transfer, and the session records generate 8-12KB per session. For some size estimates, here are a few examples:

- 100 sessions per day of 1000 files each, all external transfers between managed and unmanaged nodes = approx 201 MB per day db growth, 6.03 GB per month, 73.4 GB per year.
- 1000 sessions per day of 1 file each, all internal between managed nodes = approx 28 MB per day, 840 MB per month, 10 GB per year.
- 1000 sessions per day, 10,000 files each, 50% internal between managed nodes, 50% external with unmanaged node = approx 30 GB per day, 900 GB per month, 11 TB per year.

Installing Console

⚠ **Warning:** Due to incompatible common components, IBM Aspera Console and IBM Aspera Faspex *cannot* be installed on the same machine. IBM Aspera does not support this combination.

Perform the following steps as an Administrator (or Domain Administrator if in an Active Directory environment).

1. Upgrade to Windows Installer 4 and later.

   The Console installer requires Windows Installer 4+ for a successful configuration. You may download the latest version of Windows Installer from the Microsoft website: www.microsoft.com/en-us/download/details.aspx?id=8483
2. Download Console installation components. Use the credentials provided by Aspera to download the installer.
3. Run the downloaded installation components.

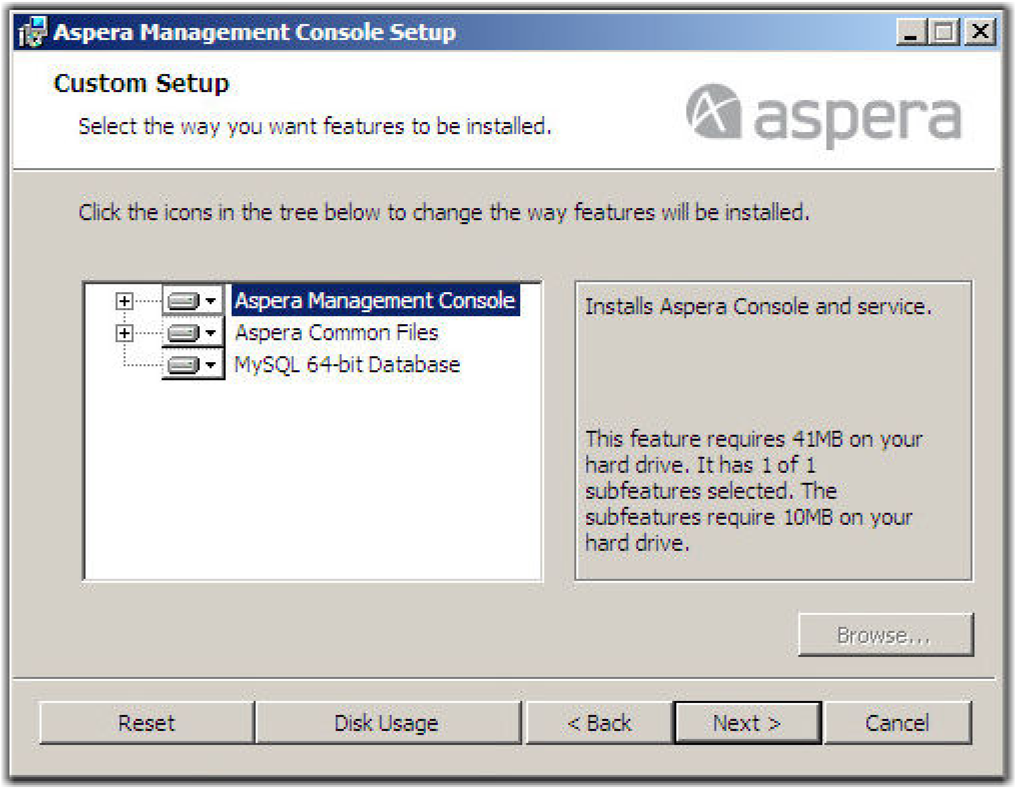
   Double-click the Console installer to start the installation.

   Note: If you are running Windows Server with User Account Control (UAC) enabled, run the installation as an administrator. Right-click the installer and click **Run as administrator**.
4. Choose the setup type.

   After the Console's End-User License Agreement screen, you should see "Choose Setup Type" with the following two options:

Option	Description
Typical	Install all components required by Console, including the Console application, Ruby, MySQL common files, and Console's MySQL database.
Custom	Select the components to install. You can use your existing installation of Ruby, MySQL or Console's MySQL database for the new installation.

If you select custom install, you can choose components to install in the following screen.



Aspera Management Console Setup

Custom Setup

Select the way you want features to be installed.

Click the icons in the tree below to change the way features will be installed.

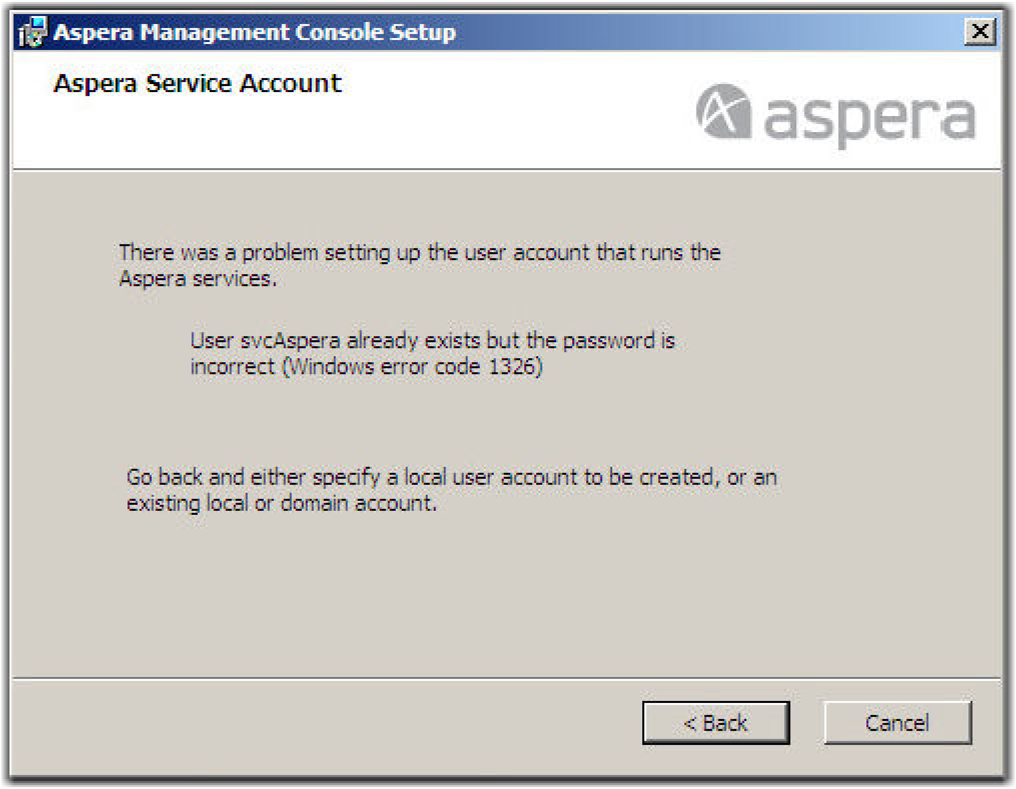
- Aspera Management Console
- Aspera Common Files
- MySQL 64-bit Database

Installs Aspera Console and service.

This feature requires 41MB on your hard drive. It has 1 of 1 subfeatures selected. The subfeatures require 10MB on your hard drive.

5. Enter the user name and password of the system account used as the Aspera service account.

   This account can be either a local account or an Active Directory account. If the Aspera service account is an existing user, enter the user's password. Otherwise, create a new user name and password. By default, the user name is **svcAspera**. If the existing user's password you have entered is incorrect, or you want to change the Aspera service user, see .



Aspera Management Console Setup

Aspera Service Account

There was a problem setting up the user account that runs the Aspera services.

User svcAspera already exists but the password is incorrect (Windows error code 1326)

Go back and either specify a local user account to be created, or an existing local or domain account.

6. Run the **asctl** setup command.

   Once installation is complete, click **Finish**. By default, the installer automatically runs the asctlsetup command. If you do not want to run the setup command automatically, clear **Launch asctl to continue the Console setup** and click **Finish**.

   Note: If Console doesn't automatically run the setup command or an error halts the process, then you can run the command manually, as shown below.

   ```
   > asctl console:setup
   ```

7. When prompted, enter the IP addresses and host names (separated by commas) that are allowed to access Console.

   You can only access the Console web application from an accepted IP address or host name. If you do not include the current IP address, you cannot log into Console.

   Note: After installation, you can edit the list of accepted IP addresses and host names by modifying `AcceptedHosts` in the `console.rb.yml` (`C:\Program Files (x86)\Aspera\Management Console\config\console.yml`) configuration file. For more information, see "Allowing Access to Console at Defined Hostnames" on page 14.

8. Update the Aspera license from the command line.

   Important: For purchasers of Aspera Enterprise, a license enabling Console as part of Enterprise can be downloaded from IBM Fix Central.

   Console administrators can update the license in the Console web UI, see "Logging Into Console for the First Time" on page 5. However, if you are automating Console installation, you can use a rake command to set the license without logging into the web UI.

   a. Set the license text as an environment variable.

      ```
      > set LICENSE_TEXT='<ASPERA_LICENSE> <DETAILS expiration_date=... </KEY> </
      ASPERA_LICENSE>'
      ```

      In this example, only part of the license text is shown. You must paste the entire license text for the license to be valid.

   b. Update the Aspera license:

      ```
      > asctl console:rake aspera:update_license
      ```

To access the Console interface, go to the following address with a browser: `http://`*`server_ip_or_name`*`:`*`port`*`/aspera/console`. For instructions on logging in for the first time, see "Logging Into Console for the First Time" on page 5.

Logging Into Console for the First Time

1. Access the Console interface by entering its hostname or IP address followed by **/aspera/console** in your web browser. For example, enter `https://`*`IP_address`*`/aspera/console`.

   Console requires a valid license key before a user can access and interact with the Console interface. An administrator must log in to paste a valid license or upload a valid license file before other users can log into Console. No other user interaction is permitted until a valid license is installed.

2. Enter the username and password and click **Login**. At this point, Console prompts you to change your password.

   Verify the old password. Then, enter and confirm a new password. Click **Change Password** to save your new password and login.

   Note: Passwords must be at least six characters long, with at least one letter, one number, and one symbol.

3. Console directs you to the Console configuration page to update your Console license with the license Aspera provided to you. Click **Upload a license file** to upload a license file or paste the license text into the text window.

Upgrading Console to the Current Version

Important: IBM Aspera supports direct upgrades to the current General Availability (GA) version from only two GA versions prior to the current release. To upgrade to the latest version, you must be within two GA versions of the current version. Upgrading from older version requires upgrading in steps. For example, if you are four GA versions behind, upgrade to two GA versions behind (GA - 2), and then upgrade to the current GA version.

⚠ **Warning:**

Prior to performing any upgrade, IBM Aspera strongly recommends customers:

1. Perform a full environment back up and ensure the back up is successful. In case the upgrade fails, the only reliable, short-term fix is to roll back the environment using the back up.
2. Test the upgrade in a test environment comparable to the production environment.
3. If upgrading the test environment is successful, upgrade the production environment, but do not bring the production environment back online.
4. Prior to bringing the production environment back online, the customer must test the application to determine if an immediate rollback is needed. Otherwise, customers risk losing all data generated between upgrade and rollback.

Note: As of Console 3.0.3 and later, Console stores license information in the database, rather than in a file located at `C:\Program File (x86)\Aspera\Management Console\config`. If a license is added to the directory before the upgrade script is run, the script migrates the license information from the file into the database and deletes the file. To modify your license, see "Updating your Console License" on page 98.

1. Upgrade to Windows Installer 4 and later.

   The Console installer requires Windows Installer 4+ for a successful configuration. You may download the latest version of Windows Installer from the Microsoft website: www.microsoft.com/en-us/download/details.aspx?id=8483
2. Back up the Console database using an **`asctl`** command. Open a Command Prompt (**Start > All Programs > Accessories >Command Prompt**) and execute the following command:

   ```
   > asctl console:backup_database
   ```
3. Stop Console and all related services.

   Run the following **`asctl`** command:

   ```
   > asctl all:stop
   ```
4. Back up `my.cnf` if you have customized it.

   Upgrading the Aspera Common Components overwrites the existing `my.cnf` with default values.

   Create a backup of the existing file, which can be found in the following location:

   C:\Program Files (x86)\Common Files\Aspera\Common\mysql\my.conf

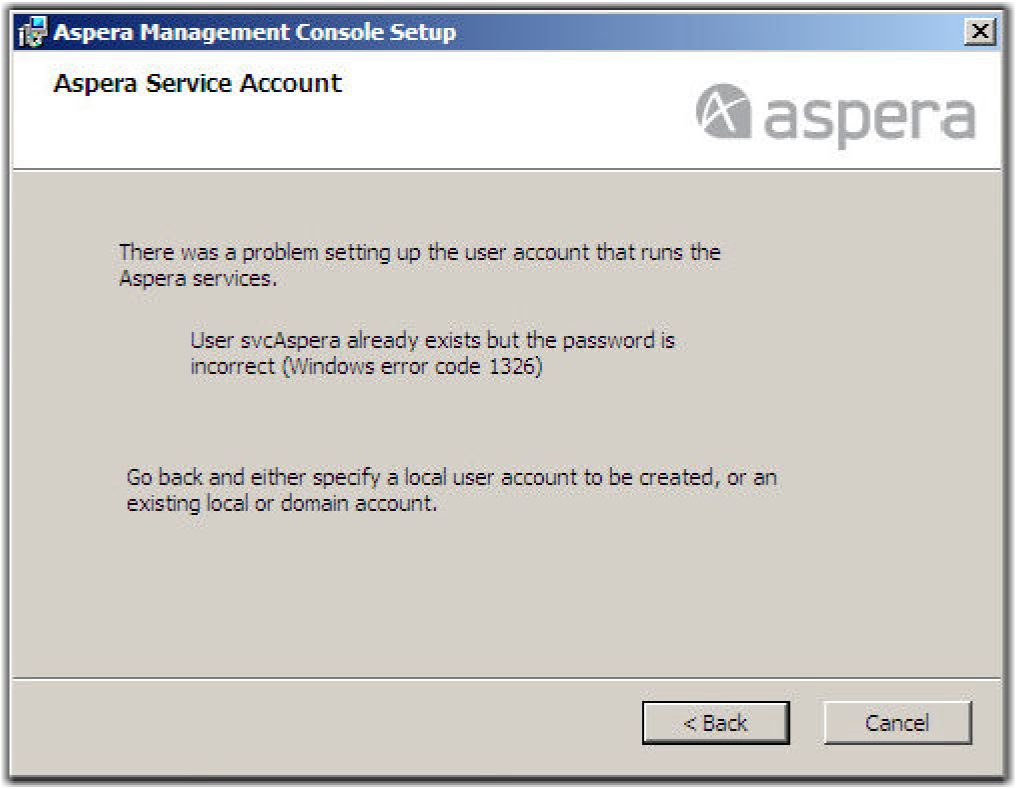
   Save the backup outside the `aspera` directory.
5. Download Console installation components. Use the credentials provided by Aspera to download the installer.
6. Install components.

   Important: If you have set a custom database backup path with the **SQL_EXPORT_DIR** environment variable, execute the installer in the same command prompt (set `SQL_EXPORT_DIR=db-export-path`), so that the installer can use the variable.

Double-click the Console installer to start the installation.

7. Enter the user name and password of the system account used as the Aspera service account.

   This account can be either a local account or an Active Directory account. If the Aspera service account is an existing user, enter the user's password. Otherwise, create a new user name and password. By default, the user name is **svcAspera**. If the existing user's password you have entered is incorrect, or you want to change the Aspera service user, see .

   Aspera Management Console Setup

   Aspera Service Account

   There was a problem setting up the user account that runs the Aspera services.

   User svcAspera already exists but the password is incorrect (Windows error code 1326)

   Go back and either specify a local user account to be created, or an existing local or domain account.

8. Click **Install**.
9. Upon completion, the installer will prompt you to execute the **asctl** upgrade command. If you did not select the option, run the **asctl** upgrade command in a Command Prompt:

   ```
   > asctl console:upgrade
   ```

10. When prompted, enter a list of hostnames at which users can access the Console UI or API. If you have SAML configured, include the SAML server hostname to allow redirection.

    For example, to allow access to Console at:

    - `localhost`
    - `console.example.com`
    - `10.0.1.128` (the server hostname)
    - `shib-idp.example.com` (SAML server hostname)

    ```
    AcceptedHosts: localhost,console.example.com,10.0.1.128,shib-idp.example.com
    ```

    In this configuration, clients can access Console at those three addresses, but cannot access Console from any other address.

    Important: You must whitelist the hostname of your SAML redirect URL to allow SAML users to authenticate through SAML and access Console.

11. Restore any `my.cnf` customizations.

    Copy the custom configurations from your backup to: `C:\Program Files (x86)\Common Files\Aspera\Common\my.cnf`.

12. Aspera highly recommends upgrading all nodes to the latest version of IBM Aspera High-Speed Transfer Server.

    For more information, see "Converting Legacy Nodes" on page 33.

Upgrading Console When MySQL is Remote

Important: IBM Aspera supports direct upgrades to the current General Availability (GA) version from only two GA versions prior to the current release. To upgrade to the latest version, you must be within two GA versions of the current version. Upgrading from older version requires upgrading in steps. For example, if you are four GA versions behind, upgrade to two GA versions behind (GA - 2), and then upgrade to the current GA version.

Warning:

Prior to performing any upgrade, IBM Aspera strongly recommends customers:

1. Perform a full environment back up and ensure the back up is successful. In case the upgrade fails, the only reliable, short-term fix is to roll back the environment using the back up.
2. Test the upgrade in a test environment comparable to the production environment.
3. If upgrading the test environment is successful, upgrade the production environment, but do not bring the production environment back online.
4. Prior to bringing the production environment back online, the customer must test the application to determine if an immediate rollback is needed. Otherwise, customers risk losing all data generated between upgrade and rollback.

Note: As of Console 3.0.3 and later, Console stores license information in the database, rather than in a file located at `C:\Program File (x86)\Aspera\Management Console\config`. If a license is added to the directory before the upgrade script is run, the script migrates the license information from the file into the database and deletes the file. To modify your license, see "Updating your Console License" on page 98.

1. Back up the MySQL database.

   On the machine with the MySQL server, run:

   ```
   > mysqldump -u mysql_username -p mysql_user_password --all-databases > /path/to/backup_name
   ```

2. Back up `my.cnf` if you have customized it.

   Upgrading the Aspera Common Components overwrites the existing `my.cnf` with default values.

   Create a backup of the existing file, which can be found at:

   C:\Program Files (x86)\Common Files\Aspera\Common\mysql\my.conf

   Save the backup outside the `aspera` directory.

3. Back up the Console database using an **`asctl`** command. Open a Command Prompt (**Start > All Programs > Accessories >Command Prompt**) and execute the following command:

   ```
   > asctl console:backup_database
   ```

4. Stop Console and all related services.

   Run the following **`asctl`** command:

   ```
   > asctl all:stop
   ```

5. Download Console installation components. Use the credentials provided by Aspera to download the installer.

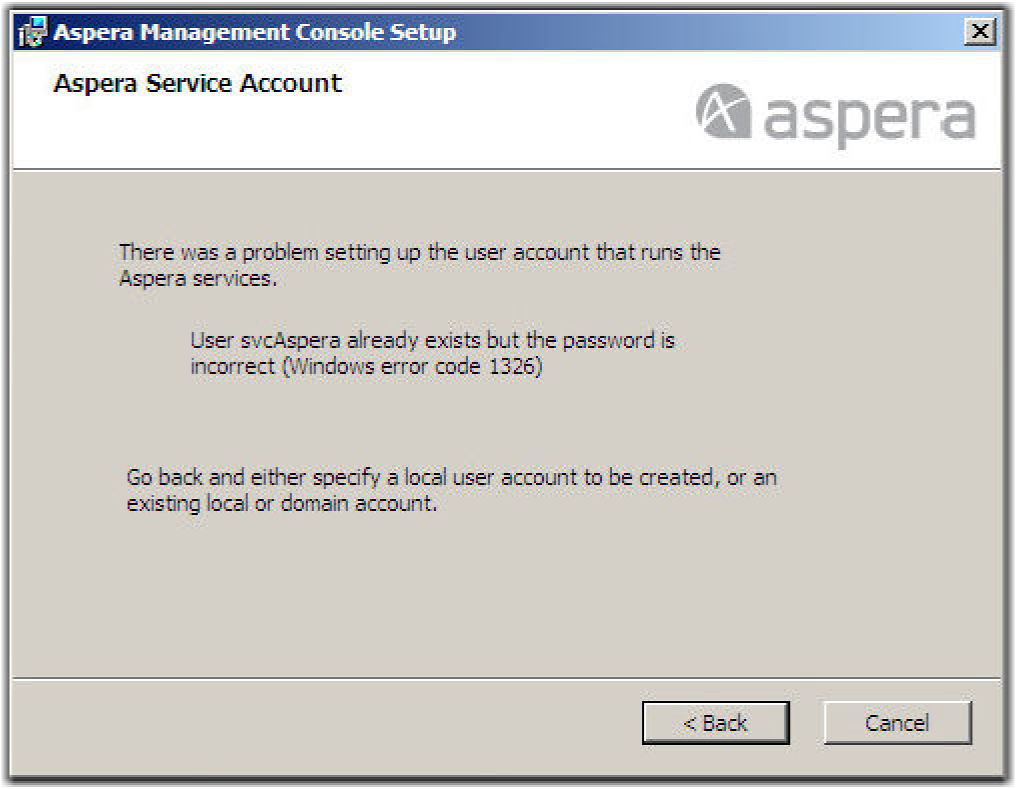
6. Install components.

   Important: If you have set a custom database backup path with the **SQL_EXPORT_DIR** environment variable, execute the installer in the same command prompt (set `SQL_EXPORT_DIR=db-export-path`), so that the installer can use the variable.

Double-click the Console installer to start the installation.

7. Enter the user name and password of the system account used as the Aspera service account.

   This account can be either a local account or an Active Directory account. If the Aspera service account is an existing user, enter the user's password. Otherwise, create a new user name and password. By default, the user name is **svcAspera**. If the existing user's password you have entered is incorrect, or you want to change the Aspera service user, see .

   There was a problem setting up the user account that runs the Aspera services.

   User svcAspera already exists but the password is incorrect (Windows error code 1326)

   Go back and either specify a local user account to be created, or an existing local or domain account.

8. Click **Install**.
9. Upon completion, the installer will prompt you to execute the **asctl** upgrade command. If you did not select the option, run the **asctl** upgrade command in a Command Prompt:

   ```
   > asctl console:upgrade
   ```

   When prompted, choose the detailed setup. Configure the Console database to be your remote MySQL database.

   Adapt your answers to your current configuration in details:

   - The user must have write access to the `aspera_console` DB. This info can be found by looking inside the file `/opt/aspera/console/config/database.yml`.
   - Be sure to set the database server hostname and IP address correctly.
10. When prompted, enter a list of hostnames at which users can access the Console UI or API. If you have SAML configured, include the SAML server hostname to allow redirection.

    For example, to allow access to Console at:

    - `localhost`
    - `console.example.com`
    - `10.0.1.128` (the server hostname)
    - `shib-idp.example.com` (SAML server hostname)

    ```
    AcceptedHosts: localhost,console.example.com,10.0.1.128,shib-idp.example.com
    ```

In this configuration, clients can access Console at those three addresses, but cannot access Console from any other address.

Important: You must whitelist the hostname of your SAML redirect URL to allow SAML users to authenticate through SAML and access Console.

11. Restore any `my.cnf` customizations.

    Copy the custom configurations from your backup to: `C:\Program Files (x86)\Common Files\Aspera\Common\my.cnf`.

12. Aspera highly recommends upgrading all nodes to the latest version of IBM Aspera High-Speed Transfer Server.

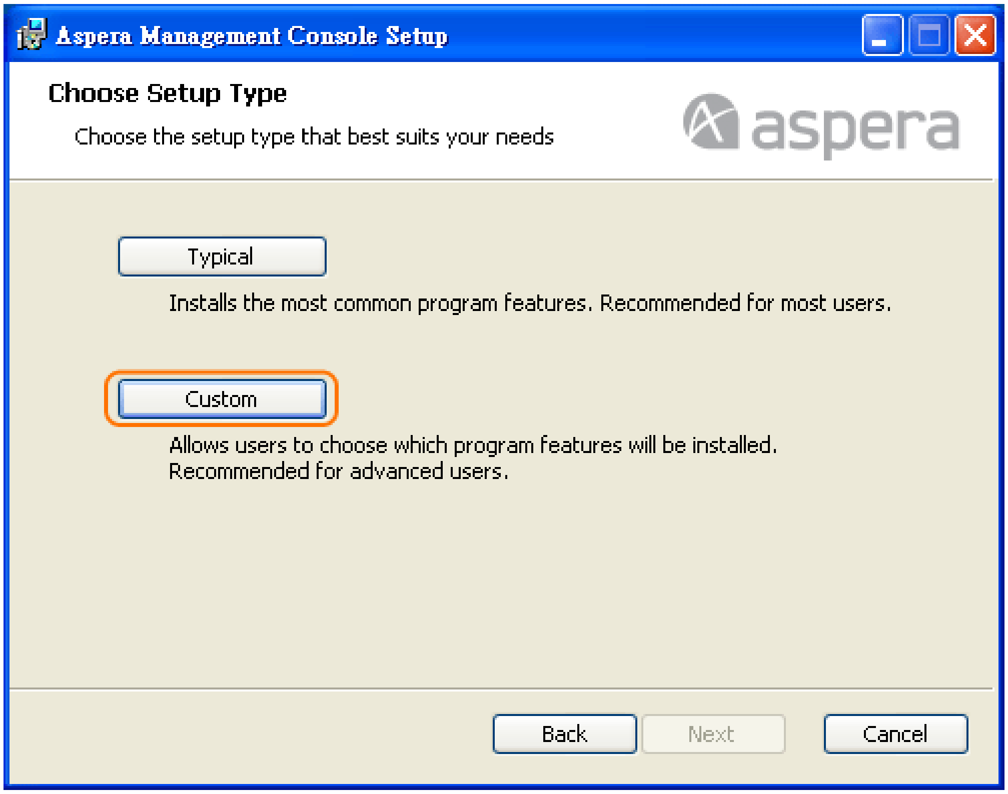
    For more information, see "Converting Legacy Nodes" on page 33.

Using IBM Aspera MySQL on a Separate Machine

Install the MySQL database included in the IBM Aspera Common Components on a remote server, and configure Console to connect to the remote MySQL database.

1. On the remote server, download and install only the IBM Aspera Common Components.

   a) Select **Custom**.



Aspera Management Console Setup

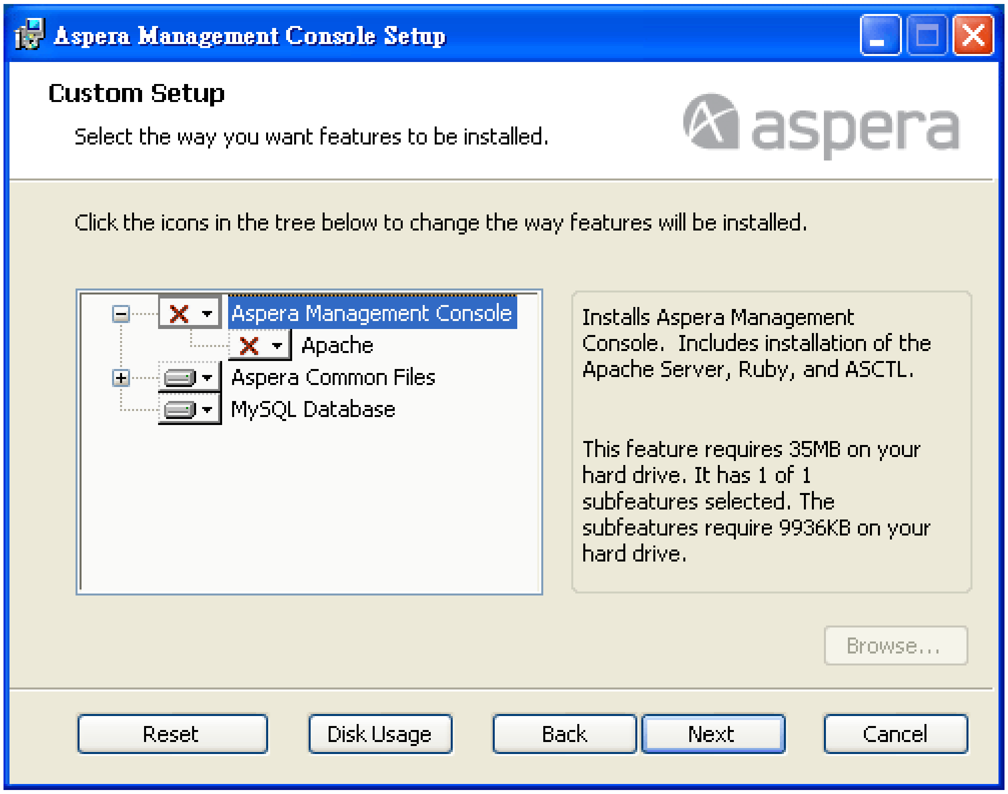
Choose Setup Type

Choose the setup type that best suits your needs

Installs the most common program features. Recommended for most users.

Allows users to choose which program features will be installed. Recommended for advanced users.

   b) Do not install **Aspera Management Console**.



Aspera Management Console Setup

Custom Setup

Select the way you want features to be installed.

Click the icons in the tree below to change the way features will be installed.

- Aspera Management Console
  - Apache
- Aspera Common Files
- MySQL Database

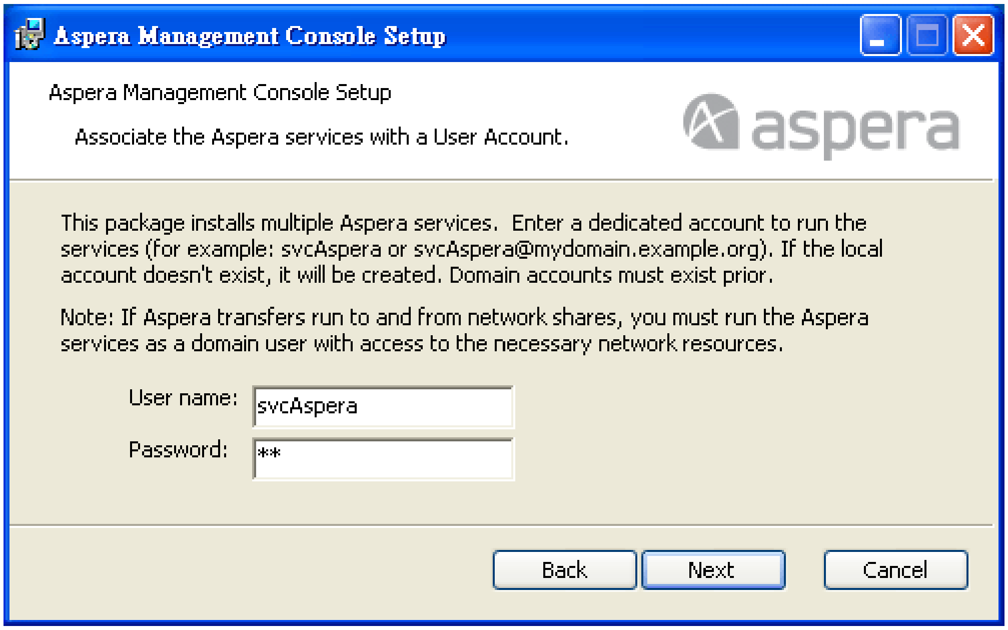
Installs Aspera Management Console. Includes installation of the Apache Server, Ruby, and ASCTL.

This feature requires 35MB on your hard drive. It has 1 of 1 subfeatures selected. The subfeatures require 9936KB on your hard drive.

c) Create or update the Aspera service account.

By default, the user name is *svcAspera*. If the server is configured to accept the domain user login, use a domain account that has been added to the local administrator's group to run the services.

If the local account does not already exist, enter new credentials and click **Next**. The installer will create an account with the information you have entered. If the account exists (created through the previous installation), enter the account's password and click **Next**.



Aspera Management Console Setup

Aspera Management Console Setup

Associate the Aspera services with a User Account.

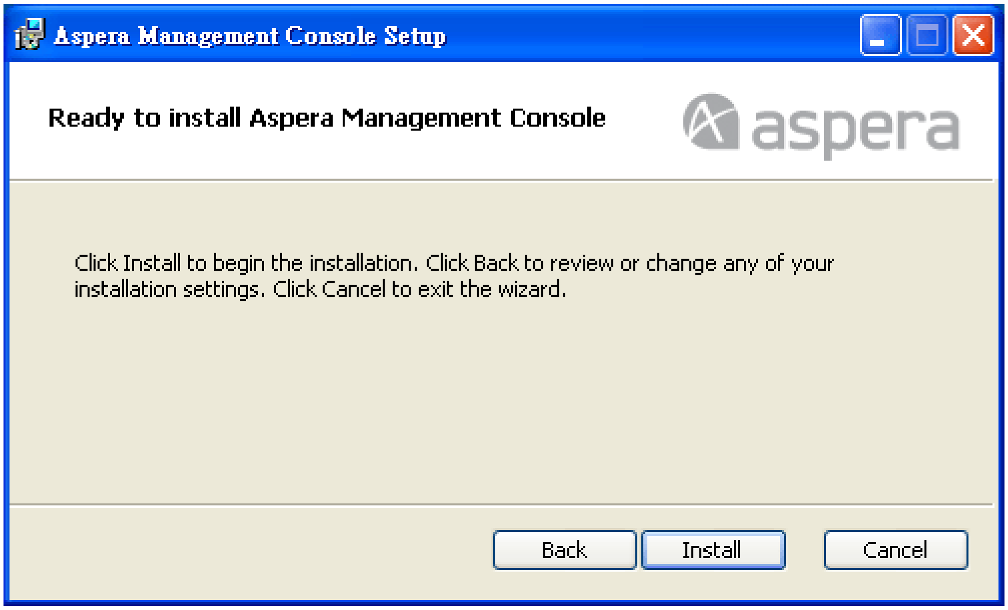
This package installs multiple Aspera services. Enter a dedicated account to run the services (for example: svcAspera or svcAspera@mydomain.example.org). If the local account doesn't exist, it will be created. Domain accounts must exist prior.

Note: If Aspera transfers run to and from network shares, you must run the Aspera services as a domain user with access to the necessary network resources.

User name: svcAspera

Password: **

d) Select **Install** to start the installation.



Ready to install Aspera Management Console

Click Install to begin the installation. Click Back to review or change any of your installation settings. Click Cancel to exit the wizard.

When finished, clear the **Launch Asctl to configure Console** option and click **Finish** to finish the installation.

2. On the remote server, set up the MySQL database:

   a) Run:

   ```
   > asctl mysql:setup
   ```

   b) Choose streamlined or detailed setup. The prompts for each setup are listed in this table:

Item	Streamlined	Detailed
`MySQL will run on this machine (y/n)? (default: y)`		**X**
`What port would you like MySQL to listen on? (default: 4406)`	**X**	**X**
`Where would you like MySQL to store data: (default: C:/Program files/Common Files/Aspera/Common/myql/data)`		**X**
`MySQL will need to start/restart during configuration. Continue (y/n)? (Current: y)`	**X**	**X**

   c) Lastly, a setup summary shows your settings. Enter **y** to confirm, **n** to change settings, or **x** to quit the program without saving.

3. When finished, execute this command to give Console access to the database using the given MySQL credentials:

   ```
   > asctl mysql:grant_remote_access console_server_ip mysql_username mysql_password
   ```

   Note: The default username is `root`.

4. On the Console server, configure Console to use a remote MySQL database.

   ```
   > asctl console:setup
   ```

   Answer **n** to the following question:

   ```
   MySQL will run on this machine (y/n)? (default: y)
   ```

5. On the Console server, configure Console to use the remote database.

   a) Back up the `C:\Program Files [(x86)]\Aspera\Common\mysql\database.rb.yml` files.

b) Edit `C:\Program Files [(x86)]\Aspera\Common\mysql\database.rb.yml`.

Change:

- `host` to the IP address of the remote database.
- `port` to the MySQL port (`4406`, by default).
- `password` to the remote MySQL database password.
- `user` to the remote MySQL database user.

  Note: By default, there is no `user` field. Console defaults to the `root` user. Add a new line to configure a different, non-root user. For example, `:user: remote_console_user`.

For example:

```
---
...
:hostname: 54.182.111.111
:port: 4406
:task status:
   ...
   ...
:user: remote_console_user
:password: XRs9sJFF5ja1BGlKHYLwzQ==
:setup_complete: true
```

Save your changes.

c) Edit `C:\Program Files [(x86)]\Aspera\Console\config\database.yml`.

Locate the `production` and `production_reports` sections and change:

- `host` to the IP address of the remote database.
- `port` to the MySQL port (`4406`, by default).
- `username` to the remote MySQL database user.
- `password` to the remote MySQL database password.

Note: By default, Console uses different users and passwords for the `production` and `production_reports` environments. To follow the same design, repeat the steps to grant access for a separate user that is dedicated to the `production_reports` environment (for example, `aspera_console_reports` in the following example).

For example,

```
...
production:
  reconnect: true
  encoding: utf8
  port: 4406
  adapter: mysql
  username: remote_console_user
  charset: utf8
  database: aspera_console
  host: 127.0.0.1
  collation: utf8_general_ci
  password: XRs9sJFF5ja1BGlKHYLwzQ==
production_reports:
  reconnect: true
  encoding: utf8
  port: 4406
  adapter: mysql
  username: remote_console_reports
  charset: utf8
  database: aspera_console_reports
  host: 127.0.0.1
  collation: utf8_general_ci
  password: DDfUMH+f3FAdHbwvRt+BQR==
```

Save your changes.

6. Shut down the local MySQL database and restart all other Console services.

```
> asctl mysql:disable
> asctl all:restart
```

Uninstalling Console

1. Prior to removal, stop the Console application and its services in a Command Prompt window using the **asctl** command.

   ```
   > asctl all:stop
   ```

2. Go to **Control Panel > Programs and Features**. Right-click **Aspera Management Console** and click **Uninstall**.

Allowing Access to Console at Defined Hostnames

Define a list of hostnames at which users can access the Console UI or API. To allow clients to access Console at an alternate hostname, whitelist the alternate hostname by adding it to AcceptedHosts in the C:\Program Files (x86)\Aspera\Management Console\config\console.yml file.

The list of hostnames does not refer to the client's hostname or IP address, but to the server's host name. AcceptedHosts does not restrict which hostnames can access Console, but does restrict the address at which clients can access Console.

Important: You must whitelist the hostname of your SAML redirect URL to allow SAML users to authenticate through SAML and access Console.

For example, to allow access to Console at:

- localhost
- console.example.com
- 10.0.1.128 (the server hostname)
- shib-idp.example.com (SAML server hostname)

```
AcceptedHosts: localhost,console.example.com,10.0.1.128,shib-idp.example.com
```

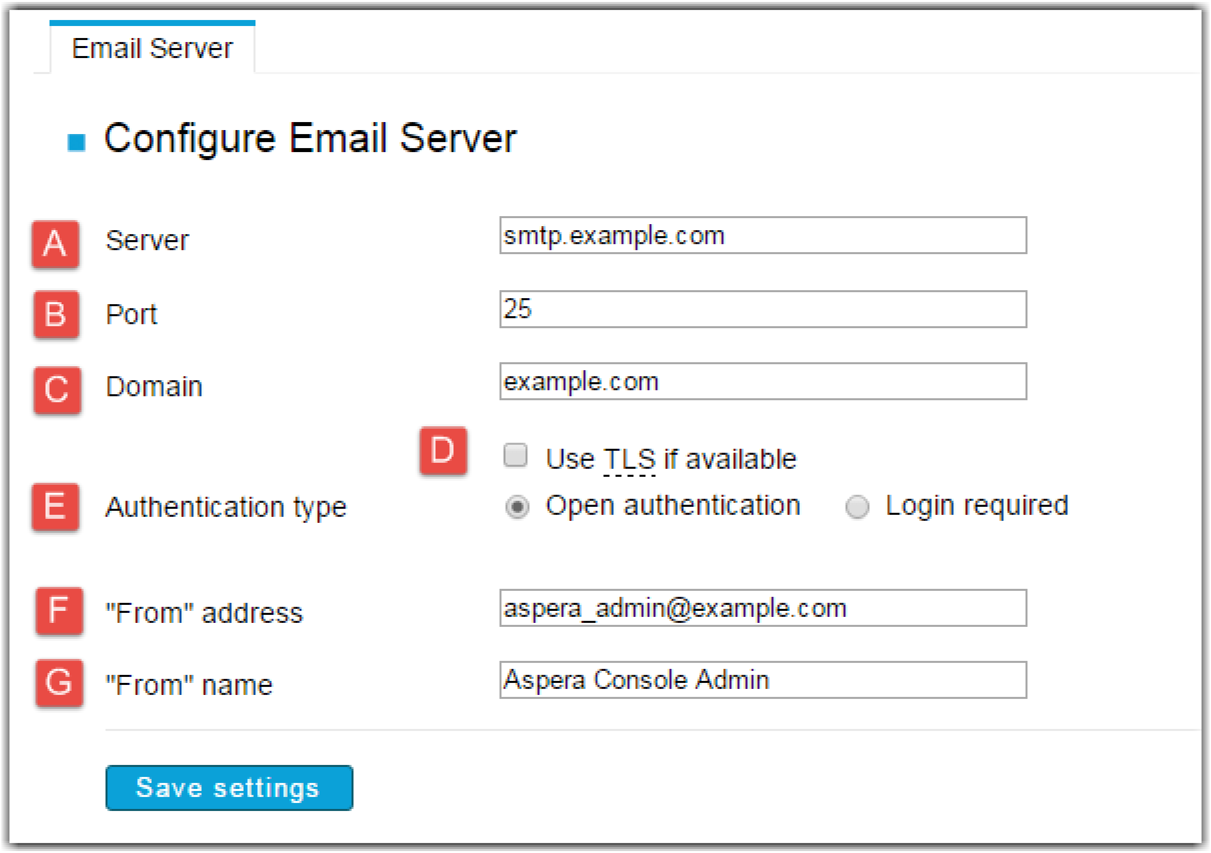
In this configuration, clients can access Console at those three addresses, but cannot access Console from any other address.

Note: The hostnames are case sensitive.

Configuring Email Notifications

Email Server Configuration

IBM Aspera Console needs to connect to a Simple Mail Transfer Protocol (SMTP) server to send email notifications.



Email Server

Configure Email Server

Server: smtp.example.com

Port: 25

Domain: example.com

☐ Use TLS if available

Authentication type: ● Open authentication ○ Login required

"From" address: aspera_admin@example.com

"From" name: Aspera Console Admin

1. Go to **Notifications** > **Email Server**.
2. Enter the SMTP server information [A, B, C].
3. Optional: Enable Transport Layer Security (TLS) [D] if available.
4. Choose the authentication type [E] of your email server.

   If your SMTP server requires login credentials, select **Login required** under **Authentication type** and enter your login credentials. Otherwise, select **Open authentication**.
5. In the **'From' address** [F] and **'From' name** [G] fields, enter the default sender email address and sender name that appear in email notifications when they receive an email notification.

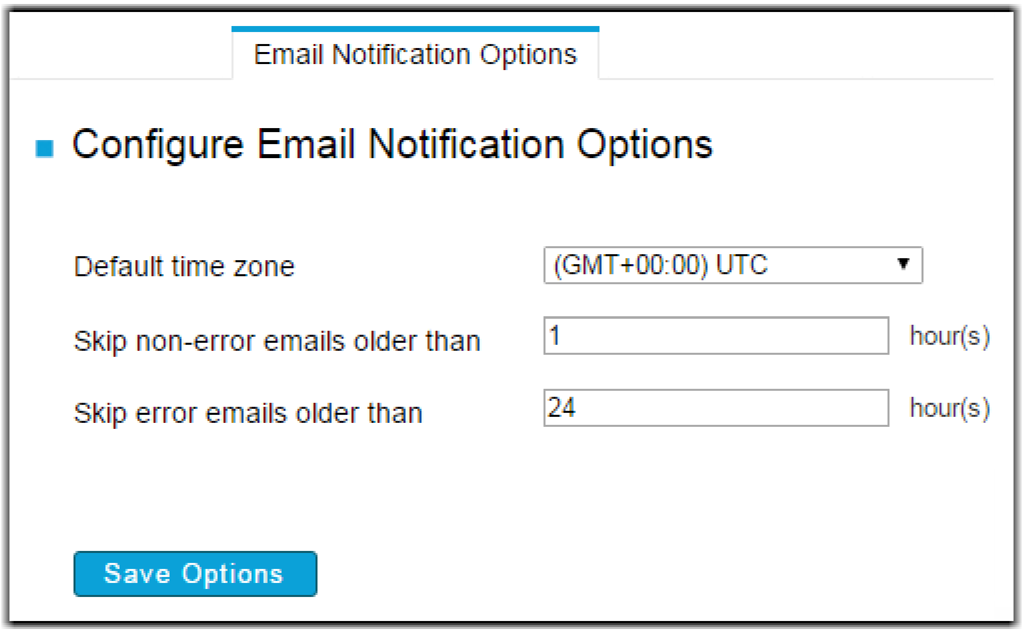
   Note:

   Expect the email address set in the **'From' address** field to receive every email notification sent by Console. Console sends email notifications to that email address and only CC's (or BCC's) recipients to optimize the email sending process.
6. Enter your email address and select **Save settings and send test email**.

   Check your email inbox for the confirmation email titled *Email settings test*. If you do not receive the email, review your settings or check your spam folder.

Configuring Notification Time Zones and Cutoff Times

1. Go to **Notifications > Email Notification Options**.



Configure Email Notification Options

Default time zone: (GMT+00:00) UTC

Skip non-error emails older than: 1 hour(s)

Skip error emails older than: 24 hour(s)

2. Select a default time zone for email timestamps.
3. Enter a cut-off time for delivering older emails.

Configuring Advanced Rulesets for Email Notifications

Configure advanced rulesets for automated generation of additional email notifications beyond the simple announcement of transfer events. Console checks configured rulesets whenever a transfer starts, completes successfully, or errors out for the final time (the transfer runs out of retries or Console detects a transfer that was supposed to retry but never did). If the transfer matches the ruleset, Console sends an email notification to the designated recipients.

1. Go to **Notifications > Advanced Rulesets** and click **Create New Ruleset**.
2. Enter a description of the ruleset.
3. Optional: Disable the ruleset to control when the ruleset comes into effect.
4. Select a filter.

Filter	Description
Address	Filter by the IP address of a node machine.
Cookie	Filter by information in a transfer cookie. For more information on transfer cookies, see "Creating a Cookie Parsing Rule" on page 62.
Contact	Filter by the contact assigned by Console. A contact can be a Console user name, a Faspex user name, a SSH account, or a customized value obtained from a transfer cookie. For example, a contact can be "admin console", "aspera ssh", or "aspera faspex" and so on.
Failover Group Name	Filter by the failover group name of the node. For more information about failover groups, see "Configure Failover Groups" on page 61.
Faspex Metadata	Filter by metadata found in a Faspex file package.
File Path	Filter by the file path of the transfer.
SSH User	Filter by the username of the SSH user that started the transfer.

Filter	Description
	Note: This works only for Console-triggered transfers with endpoints matching the SSH User.
Tags	Filter by the JSON hash used to tag the transfer. For more information on transfer tags, see "Working with Tags" on page 106.

5. Select the side to apply the filter.

Side	Description
Either	Apply the filter to both sides.
Source	Apply the filter to the source node.
Destination	Apply the filter to the destination node.
Client	Apply the filter to the node initiating the transfer request.
Server	Apply the filter to the node receiving the transfer request.

6. Select the comparator and enter the value.

   Note: Select **NOT** to exclude entries matching the value.

   For example, set the following parameters to send an email notification every time a node with the defined IP address participates in a transfer.

MATCH	SIDE	NOT	COMPARISON	VALUE
Address	Either		=	10.0.0.1

7. Designate email recipients.

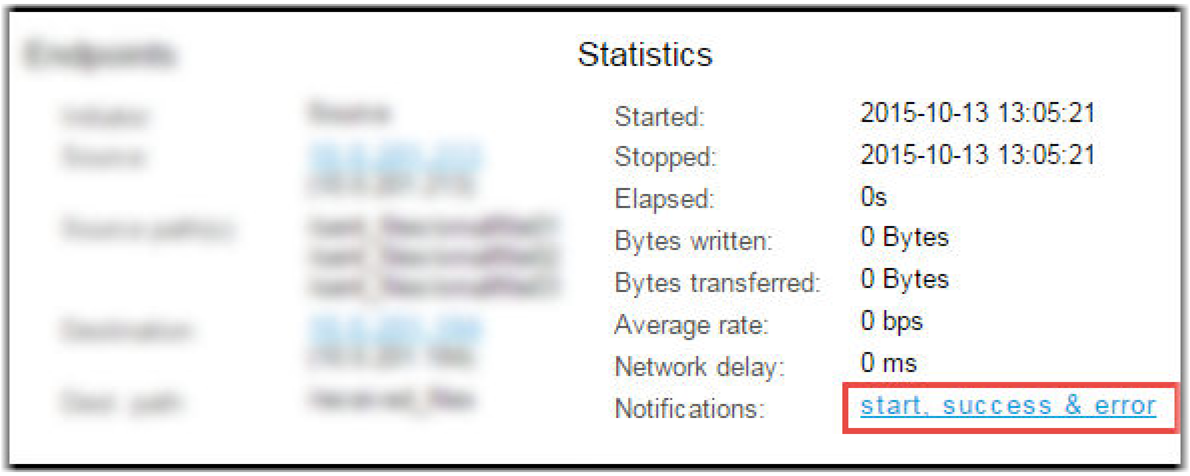
   Enter an email address and click **Add**. Select an email template for each transfer event.

8. Click **Create**.

The newly created template appears on the Advanced Rulesets page where you can **disable** or **enable**, **edit**, **copy**, and **delete** rules.

View Email Notification Statistics

You can monitor notification activity in the Session Notifications report for each transfer. To view the report, go to a transfer's Sessions Details page. Click The Statistics column contains either a link describing the type of notifications configured for that session or **None** if no notifications were configured. Click the link to display the Session Notifications page.



Statistics

Started:	2015-10-13 13:05:21
Stopped:	2015-10-13 13:05:21
Elapsed:	0s
Bytes written:	0 Bytes
Bytes transferred:	0 Bytes
Average rate:	0 bps
Network delay:	0 ms
Notifications:	start, success & error

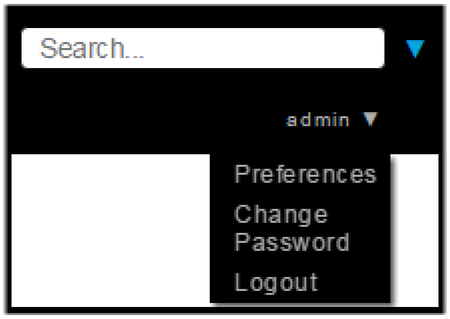
The Session Notifications page provides the following information about the transfer:

- **Session Details:** This section gives basic information about the transfer, such as its name, status, and start and stop times
- **Configured Notifications:** This section shows which types of notification were configured for this transfer (start, success, or error) and the name of the template configured for each.
- **Email Messages Sent (or Attempted)** This section shows which types of notification were actually sent or attempted for this transfer (start, success, or error) and the name of the template used for each. You can see more detail about a message by clicking on it to launch the Email Message Details page, which provides more detail about a message, including its content. You can also resend messages listed in this section by clicking **resend**. This may be useful in cases where recipients are not receiving messages due to email server or configuration issues.

Configuring Personal Email Notifications

Individual users can manage personal email notifications from their Preferences menu.

1. Open the Preferences page and select **Email Notifications**.



2. Select email templates for notifications that are triggered by the following events: transfer start, transfer success, or transfer error.

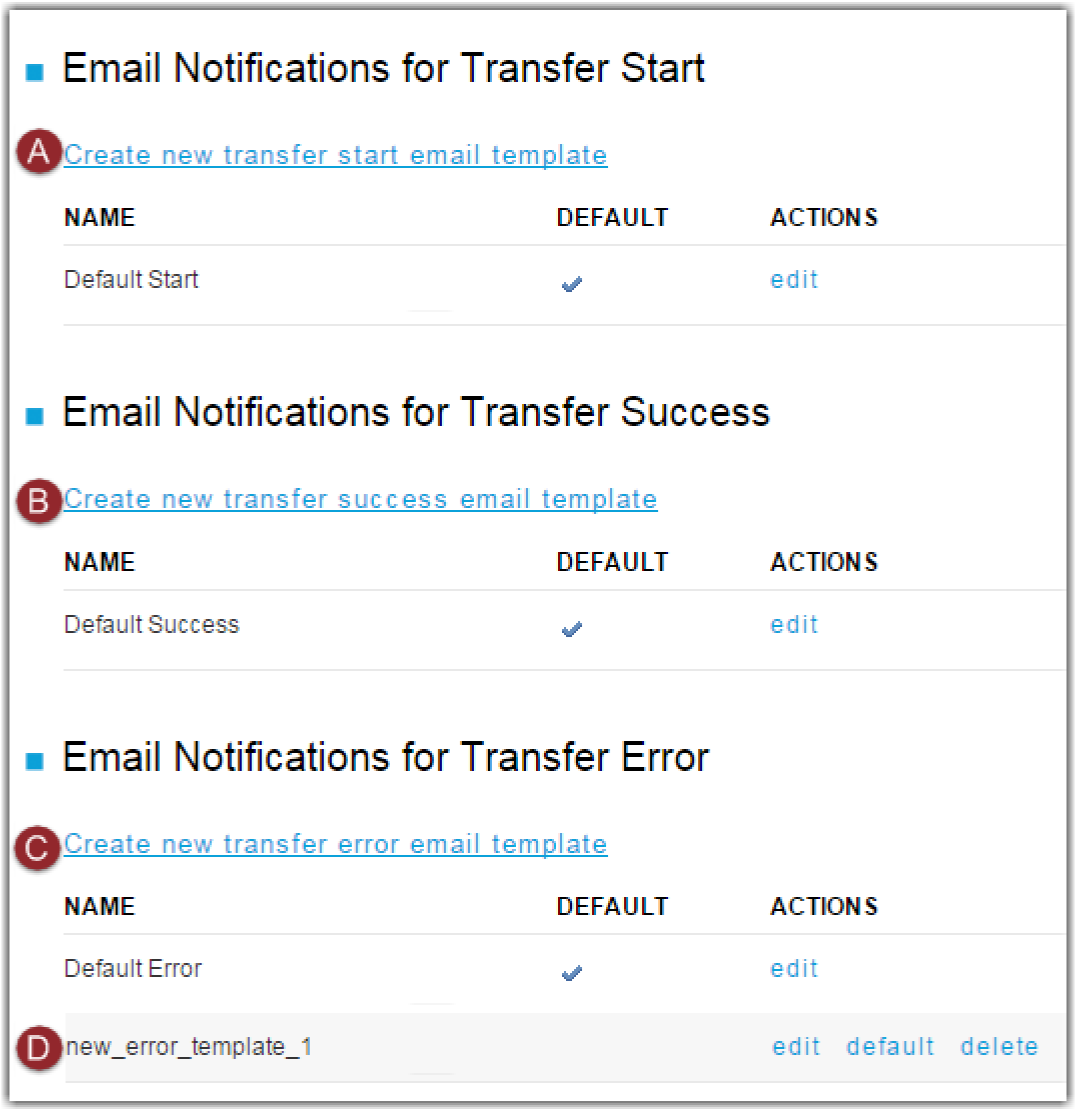
   You can create new templates or modify existing templates by going to **Notifications > Email Templates**. For more information on how to create and modify email templates, see "Editing Email Templates" on page 20.
3. Select or clear global email notifications. By default, Console notifies you for transfers that you start when those transfers start, succeed, or fail.
4. For each specific transfer path listed, select or clear notifications for transfer path. These notifications are disabled by default.
5. Click **Update**.

Configuring Email Notification Templates

Creating a New Email Notification Template

Console allows you to create and modify email notification templates based on three transfer events: transfer start, transfer success, and transfer error. You can customize emails based on recipient needs by creating a new template. For example, an error notification email to an internal admin typically contains as much information as possible, while a notice to an outside party might contain a bare minimum of information. You can edit the included default templates, create and edit new templates, and change which templates are used as defaults.

1. Go to **Notifications** > **Email Templates**.
2. Click on the appropriate "Create new..." link.



■ Email Notifications for Transfer Start

Create new transfer start email template

NAME	DEFAULT	ACTIONS
Default Start	✓	edit

■ Email Notifications for Transfer Success

Create new transfer success email template

NAME	DEFAULT	ACTIONS
Default Success	✓	edit

■ Email Notifications for Transfer Error

Create new transfer error email template

NAME	DEFAULT	ACTIONS
Default Error	✓	edit
new_error_template_1		edit default delete

To create a new template, click **Create new transfer start email template** (A), **Create new transfer success email template** (B), or **Create new transfer error email template** (C) depending on the situation for which you want to send an email notification.

The new template (D) appears listed under the default template.

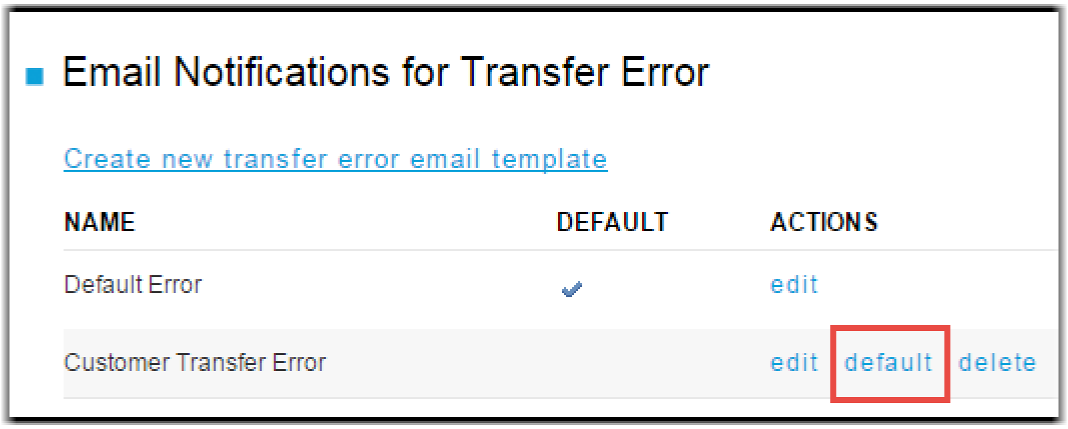
3. Rename the template.

   Click **Edit Plain Template** to open the plain text editor. Enter a descriptive name of this template in the Template name field. At this point, you can edit the template. For more information on editing templates, see "Editing Email Templates" on page 20. Otherwise, click **Save** to rename the template and return to the template preview page.

   Note: To ensure that information displays correctly in the email, edit both the plain text and HTML code versions of the template.

4. Optional: Make this template the default template.

   Return to the Email Templates page by clicking the **Email Templates** tab. Find your renamed email template and click **default**.



Email Notifications for Transfer Error

Create new transfer error email template

NAME	DEFAULT	ACTIONS
Default Error	✓	edit
Customer Transfer Error		edit default delete

Editing Email Templates

Console allows you to create and modify email notification templates based on three transfer events: transfer start, transfer success, and transfer error. You can customize emails based on recipient needs by creating a new template. For example, an error notification email to an internal admin typically contains as much information as possible, while a notice to an outside party might contain a bare minimum of information. You can edit the included default templates, create and edit new templates, and change which templates are used as defaults.

1. Go to **Notifications > Email Templates**.
2. Click **edit** for the email template you want to configure.

Note: To ensure that information displays correctly in the email, edit both the plain text and HTML code versions of the template.

3. Click **Edit Plain Template**.

Field	Description
Template name	Modify the name of the template displayed in Console.
From Name	Enter the name displayed as the email sender.
Reply-to Address	Enter the email address receiving replies from the email recipient.
Subject	Modify the email subject line.
Body	Modify, add, or remove the default text. The yellow box at the top of the page lists special text strings you can use in the message body. Console replaces the strings with the appropriate value in the actual email. The available text strings differ depending on the type of template (transfer start, transfer success, or transfer error)

For an example of how to edit the plain text version of the template, see "Email Template Example: Creating a Simple Notification for a Successful Transfer" on page 125.

4. Click **Save**.
5. Click **Edit HTML Template**.

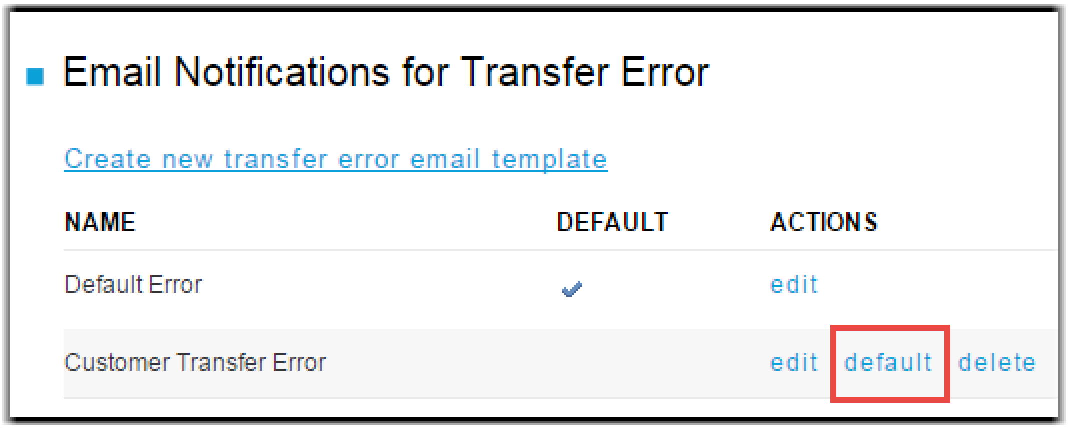
Field	Description
Template name	Modify the name of the template displayed in Console.
From Name	Enter the name displayed as the email sender.
Reply-to Address	Enter the email address receiving replies from the email recipient.
Subject	Modify the email subject line.

Field	Description
Body	Modify, add, or remove the default text. The yellow box at the top of the page lists special text strings you can use in the message body. Console replaces the strings with the appropriate value in the actual email. The available text strings differ depending on the type of template (transfer start, transfer success, or transfer error)

For an example of how to edit the HTML code of the template, see "Email Template Example: Adding Company Branding to Your Template " on page 126.

6. Click **Save**.
7. Optional: Test the email template. Enter an email address in the field and click **Send Test Email**.
8. Optional: Make this template the default template.

   Return to the Email Templates page by clicking the **Email Templates** tab. Find your renamed email template and click **default**.



Email Notifications for Transfer Error

Create new transfer error email template

NAME	DEFAULT	ACTIONS
Default Error	✓	edit
Customer Transfer Error		edit default delete

You can take the following actions for the new template:

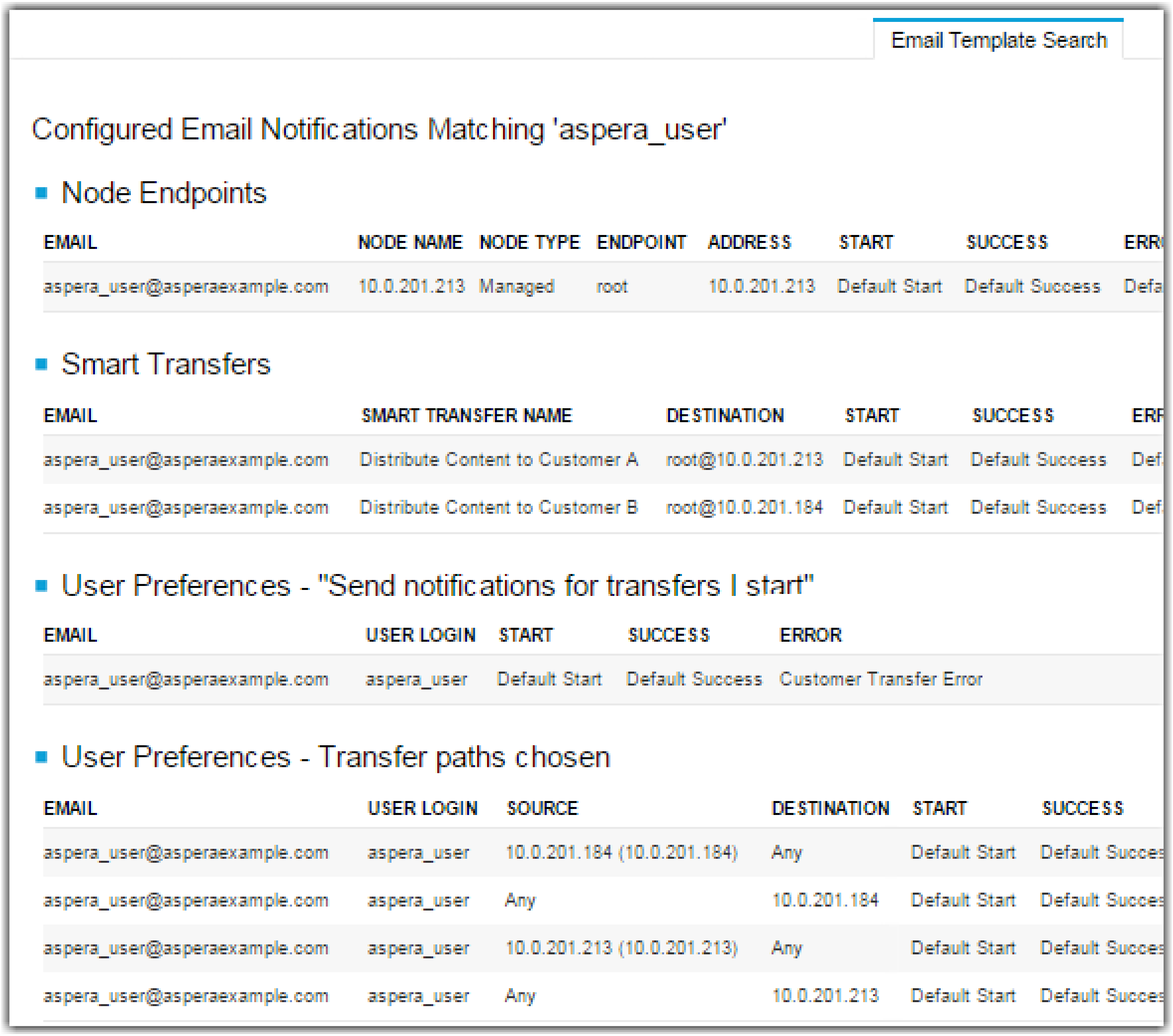
- Set this template as your personal default from your Personal Preferences page.
- Select this template when creating a transfer.
- Select this template for an Advanced Ruleset.

Displaying Email Notification Templates for an Email Address

See a list of email notifications enabled for a given email address. The results show the email recipient's node endpoints, smart transfers, and user preferences. The results also list the email templates selected as the default for each transfer event (start, success, failure).

1. Go to **Notifications > Email Templates**
2. In the Search field, enter the email address (full or partial) and click **Search**.

In the example below, the results display all pre-configured email notification templates related to the email address "jdean".



Configured Email Notifications Matching 'aspera_user'

- Node Endpoints

EMAIL	NODE NAME	NODE TYPE	ENDPOINT	ADDRESS	START	SUCCESS	ERR
aspera_user@asperaexample.com	10.0.201.213	Managed	root	10.0.201.213	Default Start	Default Success	Defa

- Smart Transfers

EMAIL	SMART TRANSFER NAME	DESTINATION	START	SUCCESS	ERR
aspera_user@asperaexample.com	Distribute Content to Customer A	root@10.0.201.213	Default Start	Default Success	Def
aspera_user@asperaexample.com	Distribute Content to Customer B	root@10.0.201.184	Default Start	Default Success	Def

- User Preferences - "Send notifications for transfers I start"

EMAIL	USER LOGIN	START	SUCCESS	ERROR
aspera_user@asperaexample.com	aspera_user	Default Start	Default Success	Customer Transfer Error

- User Preferences - Transfer paths chosen

EMAIL	USER LOGIN	SOURCE	DESTINATION	START	SUCCESS
aspera_user@asperaexample.com	aspera_user	10.0.201.184 (10.0.201.184)	Any	Default Start	Default Succes
aspera_user@asperaexample.com	aspera_user	Any	10.0.201.184	Default Start	Default Succes
aspera_user@asperaexample.com	aspera_user	10.0.201.213 (10.0.201.213)	Any	Default Start	Default Succes
aspera_user@asperaexample.com	aspera_user	Any	10.0.201.213	Default Start	Default Succes

Adding Nodes to Console

Adding Nodes to Console

In Console, a *node* is a computer that has an Aspera transfer product installed and is enabled to make transfers. A node can be a managed node or an unmanaged node.

- Managed Node: Console can initiate transfers with this node, monitor this node's activity, and configure settings for this node.
- Unmanaged Node: Console cannot initiate transfers with this node or configure its settings. Console can only monitor transfers between this node and managed nodes.

You can also add an Aspera Transfer Cluster to Console. A transfer cluster is a provisioned cluster of Aspera servers in the cloud that autoscales with traffic and use. Console can monitor transfers running on managed clusters using access key authentication for each node. You can also run reports on managed clusters. For more information, see “Adding Managed Clusters to Console” on page 31.

Adding Managed Nodes

A node must be configured for use with Console. For instructions, see:

- “Setting Up a Linux Node” on page 23
- “Setting Up a Windows Node” on page 24

- “Setting Up an OS X Node” on page 25

Once the node is configured for use with Console, add it as a managed node and configure it. This processes includes the following steps:

1. Create a new managed node in Console; see “Creating a Managed Node in Console” on page 27.
2. Authorize access to the node; see “Managing Nodes” on page 32.
3. Set up permissions for the node; see “Accounts and Permissions” on page 36.

Adding Unmanaged Nodes

To add an unmanaged node to Console, you do not need to have access to the node or configure it. As described in “Adding Unmanaged Nodes” on page 28, the process is a simpler version of adding a managed node:

1. Create a new unmanaged node in Console; see “Adding Unmanaged Nodes” on page 28.
2. Set up permissions for the node; see “Accounts and Permissions” on page 36.

Setting Up a Linux Node

A *node* is any server running IBM Aspera High-Speed Transfer Server. Aspera web applications, such as IBM Aspera Console, communicate with a node through the IBM Aspera Node API.

Console uses SSH to authenticate to the node to remotely configure nodes and uses the Node API to start and monitor transfers between nodes. Different nodes may use different Node API username and password pairs.

The instructions below assume you have already installed HSTS on your server. For instructions on installing IBM Aspera High-Speed Transfer Server Admin Guide: Installing HSTS.

1. Aspera recommends setting up the node as the `root` user. If you do not have access to the `root` user, you must give the current system user permissions to make changes to the `/opt/aspera/etc/aspera.conf` configuration file.

   Change ownership of the `aspera.conf` file to the current system user:

   ```
   # chown system_user:root /opt/aspera/etc/aspera.conf
   ```

2. Create a system user account on the node.

   Run the following command:

   ```
   # useradd username
   ```

   For example:

   ```
   # useradd xfer_user
   ```

   The examples in this topic use `xfer_user` as an example username.

The **`asconfigurator`** utility modifies the `aspera.conf` configuration file, located at: `/opt/aspera/etc/aspera.conf`.

3. Set the IP address or hostname for the node in the `aspera.conf` file with the following **`asconfigurator`** command:

   ```
   # asconfigurator -x "set_server_data;server_name,ip_or_hostname"
   ```

   For example:

   ```
   # asconfigurator -x "set_server_data;server_name,aspera.example.com"
   ```

4. Configure a HSTS transfer user account with a Node API username and password.

   Console communicates to the HSTS transfer user account through the Node API to start and monitor transfers on the node.

For instructions on adding users to HSTS, see the *IBM Aspera High-Speed Transfer Server Admin Guide: Setting Up Users*.

a) Set up the Node API user:

```
# /opt/aspera/bin/asnodeadmin -a -u node_api_username -p node_api_passwd -x
system_username --acl-set impersonation,admin
```

Note: Aspera recommends that you use different names for the system user account and transfer user account in order to minimize confusion when tracing transactions and events.

For example:

```
# /opt/aspera/bin/asnodeadmin -a -u node_user -p XF324cd28 -x xfer_user --acl-set
impersonation,admin
```

Note:

You need to escape special characters such as $ to use them in a password. For example, to use XF324$ as the password:

```
/opt/aspera/bin/asnodeadmin -a -u node_user -x xfer -p XF324\$
```

b) Run the following command to check the system user was successfully added to **asnodeadmin**:

```
# /opt/aspera/bin/asnodeadmin -l
```

Given a node user named **node_user** and a system user named **xfer_user**, the result should be similar to the following example:

```
                user       system/transfer user                    acls
====================    =======================    ====================
           node_user                  xfer_user
[impersonation,admin]
```

You can now add this node to Console.

Setting Up a Windows Node

A *node* is any server running IBM Aspera High-Speed Transfer Server. Aspera web applications, such as IBM Aspera Console, communicate with a node through the IBM Aspera Node API.

Console uses SSH to authenticate to the node to remotely configure nodes and uses the Node API to start and monitor transfers between nodes. Different nodes may use different Node API username and password pairs.

The instructions below assume you have already installed HSTS on your server. For instructions on installing IBM Aspera High-Speed Transfer Server Admin Guide: Installing HSTS.

1. Set up a system account on the node to run Aspera services.

   When you install IBM Aspera High-Speed Transfer Server, the installer automatically creates a system account to run Aspera services. Aspera recommends using this default account (**svcAspera**) to run Aspera services.

   The examples in this topic use `svcAspera` as the example transfer user.

   If you do not wish to use `svcAspera` as the transfer user, create a new Windows system user account. Log in as that user for Windows to set up the user's home folder.

The **`asconfigurator`** utility modifies the `aspera.conf` configuration file, located at: `C:\Program Files (x86)\Aspera\Enterprise Server\etc\aspera.conf`.

2. Set the IP address or hostname for the node in the `aspera.conf` file with the following **`asconfigurator`** command:

   ```
   > asconfigurator -x "set_server_data;server_name,ip_or_hostname"
   ```

For example:

```
> asconfigurator -x "set_server_data;server_name,aspera.example.com"
```

3. Configure a HSTS transfer user account with a Node API username and password.

   Console communicates to the HSTS transfer user account through the Node API to start and monitor transfers on the node.

   For instructions on adding users to HSTS, see the *IBM Aspera High-Speed Transfer Server Admin Guide: Setting Up Users*.

   a) Run the following commands to set up the Node API user:

   ```
   > asnodeadmin -a -u node_api_username -p node_api_passwd -x system_username --acl-set
   impersonation,admin
   ```

   Note: Aspera recommends that you use different names for the system user account and transfer user account in order to minimize confusion when tracing transactions and events.

   For example:

   ```
   > asnodeadmin -a -u node_user -p XF324cd28 -x svcAspera --acl-set impersonation,admin
   ```

   Note:

   You need to escape special characters such as $ to use them in a password. For example, to use XF324$ as the password:

   ```
   /opt/aspera/bin/asnodeadmin -a -u node_user -x xfer -p XF324\$
   ```

   b) Run the following command to check that the system user was successfully added to **asnodeadmin**:

   ```
   > asnodeadmin -l
   ```

   Given a node user named `node_user` and a system user named `svcAspera`, the output should be:

   ```
                   user      system/transfer user                   acls
   ====================    =======================    ====================
              node_user                  svcAspera
   [impersonation,admin]
   ```

You can now add this node to Console.

Setting Up an OS X Node

A *node* is any server running IBM Aspera High-Speed Transfer Server. Aspera web applications, such as IBM Aspera Console, communicate with a node through the IBM Aspera Node API.

Console uses SSH to authenticate to the node to remotely configure nodes and uses the Node API to start and monitor transfers between nodes. Different nodes may use different Node API username and password pairs.

The instructions below assume you have already installed HSTS on your server. For instructions on installing IBM Aspera High-Speed Transfer Server Admin Guide: Installing HSTS.

1. Create a system admin account on the node.

   a) Go to **System Preferences # Users & Groups**.

   b) Click the lock button and enter your admin credentials to make changes.

   c) Click the add button.

   d) Select **Administrator** from the New Account drop-down menu.

   e) Name the account.

f) Enter and verify a password for the account.

g) Click **Create User**.

h) Click **Login Options** in the users panel.

i) Click the **Join** button next to Network Account Server.

j) Click **Open Directory Utility**.

k) In the Directory Utility window, click the lock button and enter an administrator account and password to make changes.

l) From the menu bar, select **Edit # Enable Root User**.

m) Enter and verify the password.

n) Click **OK**.

The following examples use `xfer_user` as an example username.

The **`asconfigurator`** utility modifies the `aspera.conf` configuration file, located at: `/Library/Aspera/etc/aspera.conf`.

2. Set the IP address or hostname for the node in the `aspera.conf` file with the following **`asconfigurator`** command:

   ```
   # asconfigurator -x "set_server_data;server_name,ip_or_hostname"
   ```

   For example:

   ```
   # asconfigurator -x "set_server_data;server_name,aspera.example.com"
   ```

3. Configure a HSTS transfer user account with a Node API username and password.

   Console communicates to the HSTS transfer user account through the Node API to start and monitor transfers on the node.

   For instructions on adding users to HSTS, see the *IBM Aspera High-Speed Transfer Server Admin Guide: Setting Up Users*.

   a) Run the following commands to set up the Node API user:

   ```
   # /Library/Aspera/bin/asnodeadmin -a -u node_api_username -p node_api_passwd -x
   system_username --acl-set impersonation,admin
   ```

   Note: Aspera recommends that you use different names for the system user account and transfer user account in order to minimize confusion when tracing transactions and events.

   For example:

   ```
   # /Library/Aspera/bin/asnodeadmin -a -u node_user -p XF324cd28 -x xfer_user --acl-set
   impersonation,admin
   ```

   Note:

   You need to escape special characters such as $ to use them in a password. For example, to use XF324$ as the password:

   ```
   /opt/aspera/bin/asnodeadmin -a -u node_user -x xfer -p XF324\$
   ```

   b) Run the following command to check the system user was successfully added to **`asnodeadmin`**:

   ```
   # /Library/Aspera/bin/asnodeadmin -l
   ```

   Given a node user named **node_user** and a system user named **xfer_user**, the result should be similar to the following example:

   ```
                   user      system/transfer user                    acls
   ====================    =======================    ====================
              node_user                  xfer_user
   [impersonation,admin]
   ```

You can now add this node to Console.

Preparing a Non-Local User to Run the Aspera Central Service

Console communicates with nodes through the Aspera Central service. When you are setting up a Windows node and using an Active Directory (or non-local) user to run the Aspera Central service, the user must have special permissions configured to allow Console to manage the node.

1. Create or obtain an Active Directory account to use for running the Aspera services.
2. Add that Domain user account to the local Administrators group.
3. Go to **Control Panel > Administrative Tools > Local Security Policy** and give the user special permissions to allow Console to manage the node machine.

   Within the Local Security Policy window, select **Local Policies > User Rights Assignment** in the left panel, and add the desired user to the following policies (by double-clicking each policy):

   - Act as part of the operating system
   - Adjust memory quotas
   - Create a token object
   - Log on as a service
   - Replace a process level token
4. Go to **Control Panel > Administrative Tools > Services** and set the following services to run as the user:

   - Aspera Central
   - Aspera Sync
   - OpenSSH

   To set the service to run as the user, right-click the service and select **Properties**. Click the **Log On** tab and select your user.

Creating a Managed Node in Console

It is best practice to keep all your nodes up to date with the latest version of IBM Aspera High-Speed Transfer Server or IBM Aspera High-Speed Transfer Endpoint.

To verify your node machine's product version, run the following command on your node machine:

```
> ascp -A
```

If you have an older version, you can download the latest from http://asperasoft.com/downloads.

1. Go to **Nodes** and click **New Managed Node**.
2. Enter your managed node's **IP Address** and **Name**.

   Important: When adding the Console server itself as a managed node, don't enter **127.0.0.1** as the IP address unless it is the *only* node you are planning to add to Console. Even then it is not recommended. Do this only if your network or firewall configuration interferes with Console communicating with its own external address.
3. Configure the communication settings.

   Update **SSH Port**, **SSH Encryption**, and **Node API Port** if you do not want to use the default values. Select a default endpoint type from the drop-down menu. You can still change the endpoint type when you add an endpoint to this node.
4. Add the node to a failover group.

   Select **Enable failover and load balancing for Console-initiated transfers on this node**. Add the node to an existing group or select enter new name from the Failover Group Name drop-down menu to create a new group. If you select enter new name, enter a new failover group name in the prompt. For more information on failover groups, see “Configure Failover Groups” on page 61.

5. Optional: Create the three default Console groups. Select **Create default Console groups**.

   The default transfer Console groups have the following permissions:

   - **Transfer Admin:** Users in this group can manage transfers on this node. This includes initiating, canceling, and deleting transfers.
   - **Transfer Initiator:** Users in this group can only initiate transfers.
   - **Transfer Monitor:** Users in this group can only monitor transfers.

   For more information on Console groups, see “Creating Console Groups” on page 37

6. Click **Create**.

Console redirects you to the Credentials page. For more information on configuring admin credentials, see “Updating a Node's Admin Credentials” on page 28.

Updating a Node's Admin Credentials

If you have not already configured an administrative account on your node machine for use with Console, see the following instructions:

- Windows node: “Setting Up a Windows Node” on page 24.
- OS X node: “Setting Up an OS X Node” on page 25.
- Linux node: “Setting Up a Linux Node” on page 23.

Console connects to a managed node through SSH for file browsing and node configuration. You must update Console with the node's SSH credentials before Console can access the node.

If Console automatically redirected you to the Admin Credentials page as part of the process of adding a new managed node, skip the first step.

1. Go to **Nodes**. Click the **edit** link for the managed node, and click the **Credentials** tab.
2. Select **Edit Credentials**.
3. If you want to configure the node from Console, enter the node machine's SSH login credentials.

   Enter the administrative account username and password to allow Console to connect to the node machine. You can authenticate the account in one of two ways:

   - **Password authentication:** Enter the account password.
   - **Public key authentication:** Select **Use SSH Key** and select your uploaded key. To use public key authentication, you must have your SSH private key configured in Console. For instructions, on how to configure SSH keys in Console, see “SSH Keys” on page 83.

4. Enter the node machine's Node API credentials.
5. Click **Update**.
6. Click **Test Credentials** to make sure Console has a working connection to the node.

   If successful, the message "Successfully connected to node via SSH and Node API" appears in green at the top of the page.

A connection is established between Console and your managed node. To edit or remove a node, go to **Nodes** for a list of managed nodes and click **edit** or **delete** for the designated node.

Adding Unmanaged Nodes

It is best practice to keep all your nodes up to date with the latest version of IBM Aspera High-Speed Transfer Server or IBM Aspera High-Speed Transfer Endpoint. Verify the machine's product version with the administrator of the node.

1. Go to **Nodes**. Click **List Unmanaged Nodes**.
2. Click **New Unmanaged Node**.

3. Enter the node's **Address** (IP or domain name) and **Name**.

   Note: Console does not send notifications when an unmanaged node is using a fully qualified domain name (FQDN) instead of an IP address if the transfer is started by IBM Aspera High-Speed Transfer Server. Console still sends notifications for transfers started by Console.
4. Select the default endpoint type from the drop-down menu.

   You can still change the endpoint type when you add an endpoint to this node.
5. Configure SSH.

   Enter the **SSH Port** number and select the **SSH Encryption** method from the drop-down menu.
6. Click **Create** when finished.
7. To verify that your new node has been created, select **List Unmanaged Nodes** and look for your unmanaged node in the table.

A connection should be established between Console and your unmanaged node. To edit or remove a node, go to **Nodes** and click **List Managed Nodes** for a list of managed nodes. Click **edit** or **delete** for the designated node.

Connecting Aspera Shares to Console

If you want to initiated a Console transfer with an Aspera Shares application as an endpoint, follow the steps below:

1. Add Shares as an unmanaged node to Console with default endpoint type as `Shares` and Node API port as 443.

   Adding the Shares application as an unmanaged node makes it easier to save Shares credentials for use in transfers. Aspera does not recommend adding Shares applications as managed nodes because they lack the capabilities necessary for monitoring.

   For more instructions on adding an unmanaged node, see "Adding Unmanaged Nodes" on page 28.
2. Create a Shares endpoint for the unmanaged node.

   For instructions on adding an endpoint to the node, see "Adding Endpoints" on page 30.
3. Add nodes used by the Shares application as managed nodes in Console for transfer monitoring.

   For instructions on adding a managed node, see "Creating a Managed Node in Console" on page 27.
4. Create a simple or smart transfer with the Shares endpoint.

   For instructions on creating simple or smart transfers, see "Starting a Simple Transfer" on page 50 and "Creating a Smart Transfer" on page 52, respectively.

   Important: You cannot initiate transfers between Shares endpoints and SSH endpoints, or between two Shares endpoints. For more information about restrictions on endpoint combinations, see "Console Transfer Special Conditions" on page 49.

Understanding Endpoints

An *endpoint* serves as a transfer source or destination for transfers initiated in the Console UI between nodes (managed or unmanaged) and between nodes and clusters. It is defined by a login credential and address. These appear in the **Transfer** drop-down menus for **Source** and **Destination** as *login@address*, such as xasp1@10.0.0.2 for a node or ats-aws-us-east-1.aspera.io for a managed cluster.

Whenever a node or cluster is added to Console, Console automatically creates a "wildcard" endpoint with the format *@*address* (for example, *@192.168.0.100. The wildcard endpoint is listed as just the IP address or domain name. When a user selects the wildcard endpoint as a source or destination, they must enter credentials to authorize the transfer. Wildcard endpoints enable you to monitor all transfers on a node per user account or access key.

Console admins can add more endpoints to nodes and clusters, and configure them with credentials. The credentials required to set up and use an endpoint depend on the endpoint type:

- **SSH**: An Aspera transfer user's username and either a password or SSH key.

- **Node API**: An Aspera node username and password. (Only supported for managed nodes)
- **Access Key**: An Aspera access key and secret. (Only supported for clusters)

When you create a new endpoint, you can enter the credentials or leave the password/secret field blank (you must provide a login - a username or access key). Sharing a credentialled endpoint with a user who does not have login credentials allows that user to send or receive files without compromising the security of your nodes. When the password for the endpoint is not set, the user must enter it when initiating a transfer. These credentials are then stored in the user's **Saved Endpoints** under the **Preferences** tab.

Adding Endpoints

An *endpoint* serves as a transfer source or destination for transfers initiated in the Console UI between nodes (managed or unmanaged) and between nodes and clusters. It is defined by a login credential and address. These appear in the **Transfer** drop-down menus for **Source** and **Destination** as *login@address*, such as xasp1@10.0.0.2 for a node or ats-aws-us-east-1.aspera.io for a managed cluster.

For more information about endpoints, see "Understanding Endpoints" on page 29.

Tip: To use domain names as transfer endpoints, create an unmanaged node using a domain name, then add an endpoint to this unmanaged node.

1. Open the **Endpoint** dialog for a node or cluster.
   - To add an endpoint to a managed node or cluster, go to **Nodes** and click **edit** for the node or cluster to which you want to add an endpoint.
   - To add an endpoint to an unmanaged node, go to **Nodes > List Unmanaged Nodes** and click **edit** for the node or cluster to which you want to add an endpoint. Click the **Endpoints** tab.
2. Add a new endpoint.

   Click **Add Endpoint** and enter the following information:

   - **Endpoint type**: Select the endpoint type from the drop-down menu.
   - **Login**: The username or access key.
   - **Password**: The password, SSH public key, or secret. If left blank, users must enter the password, SSH public key, or secret to authorize a transfer with the endpoint. The credentials required to set up and use an endpoint depend on the endpoint type:
     - **SSH**: An Aspera transfer user's username and either a password or SSH key.
     - **Node API**: An Aspera node username and password. (Only supported for managed nodes)
     - **Access Key**: An Aspera access key and secret. (Only supported for clusters) To use SSH keys, the user must have their private key configured in Console. For instructions, see "SSH Keys" on page 83.

       Important: When using SSH key authentication, make sure that the key file on the node is not a shared key. On the node computer, the key file should be a "private" key in the specified user account.
   - **Label**: Optional descriptive name for the endpoint. Default is *login@node_address*
   - **Email address**: The email address to receive notifications of transfer activity on this endpoint. You can enter multiple email addresses by clicking **Add** after each one, then select which notifications to send to which email addresses from the drop-down menus.

   Click **Create**.
3. Verify that your endpoint is configured correctly and that the connection works.

   The new endpoint appears in the list of endpoints. To test the connection, click **test** and, on the following page, **Test Connecting to Host**. If successful, a confirmation message appears in green at the top of the page. If unsuccessful, a description of the error appears in red at the top of the page and the SSH Client Log appears at the bottom of the page.

The endpoint is now configured. If **Password Saved** is selected in the **Endpoints** table, the endpoint contains a password, an SSH key, or a secret, depending on the endpoint type, and permitted users are not required to enter credentials to use this endpoint. To edit or remove an endpoint, click **edit** or **delete**.

Set a Global Docroot for the Node

A document root, or docroot, is the area of a machine that a system user has permission to access. Setting docroots are important for maintaining security by keeping unqualified users from accessing confidential information.

1. Go to **Nodes** and click **edit** for the node.
2. Go to **Accounts** and click **edit** for the user or group you want to configure.
3. Expand the Docroot configuration section and click **Browse**. Choose the file directory you want to set as the docroot.

   The docroot is a security feature that allows you to restrict the area `asperawatchfolderd` can access. If you need to acces the entire file system, you can set the docroot path as `C:\` or leave it empty. The directory you choose is configured in the `aspera.conf` configuration file on the transfer node.
4. Click **Save changes**.

Adding Managed Clusters to Console

Adding a Managed Cluster to Console

A transfer cluster is a provisioned cluster of Aspera servers in the cloud that autoscales with traffic and use. Console can monitor transfers running on managed clusters using access key authentication for each node. You can also run reports on managed clusters. For information on configuring the Aspera Transfer Cluster, see the *IBM Aspera Transfer Cluster Manager Admin Guide*.

To add a cluster to Console, follow the instructions below.

1. Go to **Nodes** and click the **New Managed Cluster** button.
2. Enter the cluster's domain name address in the **Domain Name** field.
3. Name the cluster in Console.
4. Configure the cluster API port. Port 443 is the default port for communicating with Aspera clusters.
5. If you want to use HTTPS to connect to node, select **Use HTTPS to connect to node**. Aspera recommends using this feature for security reasons.
6. If you want Console to verify the SSL certificate of cluster nodes, select **Require signed SSL certifcate**.
7. Click **Create**.

Console redirects you to the Credentials page to add access keys for authentication. For more information about adding access keys, see “Adding Access Keys to a Managed Cluster” on page 31.

Adding Access Keys to a Managed Cluster

Aspera Transfer Clusters use access keys to authenticate to other Aspera products. Each transfer cluster can have multiple access keys. For more information about access keys and access key management, see the entries in the *IBM Aspera Transfer Cluster Manager Admin Guide: Managing a Cluster* section.

Follow the instructions below to add access keys to a managed cluster.

1. Go to **Nodes**. Click the **edit** link for the managed cluster and click the **Credentials** tab.

2. Click the **Add Access Key** button.
3. Give the access key a name to differentiate it from other keys.
4. Enter the ID and Secret associated with the key.
5. Click **Update**.

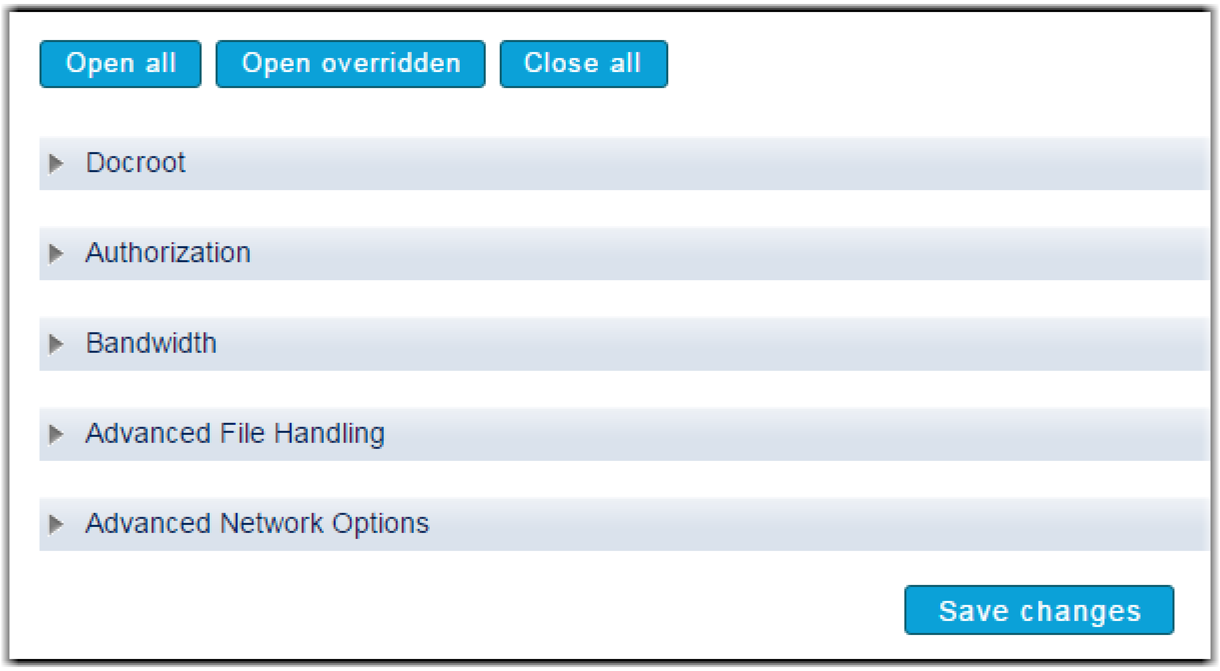
   Console redirects you to the list of Access Keys for this cluster. Verify that your new access key is in the list.
6. Test your new access key by selecting **test**.

Managing Nodes

Editing the User or Group on a Node

Console can configure user and group account settings for managed nodes that have valid admin credentials saved in Console. For more information on updating admin credentials, see "Updating a Node's Admin Credentials" on page 28.

1. Go to **Nodes** and click **edit** for the node you want to edit. Click **Accounts**.
2. Make sure that the group or user you want to configure has already been created and is available on the node machine. Console automatically detects new groups and users and lists them under the node's **Accounts** tab, but if the group or user is not listed, click **Add Group** or **Add User**.
3. Depending on whether you want to configure a node user or a node group, select **Users** or **Groups**.
4. Select the **edit** link for the user or group account you want to edit.
5. Configure the user or group account's transfer options. For more detailed information on these options, see "Node Account-Level Configuration Options" on page 137.



6. When you are finished, click **Save changes**.

Set a Docroot for a Node User or Group

A document root, or docroot, is the area of a machine that a system user has permission to access. Setting docroots are important for maintaining security by keeping unqualified users from accessing confidential information. To set a docroot for a node user or group, you must have already added them into Console.

For more information about adding node users or groups to Console, see "Editing the User or Group on a Node" on page 32.

1. Go to **Nodes** and click **edit** for the node.
2. Go to **Accounts** and click **edit** for the user or group you want to configure.
3. Expand the Docroot configuration section and click **Browse**. Choose the file directory you want to set as the docroot.

   The docroot is a security feature that allows you to restrict the area `asperawatchfolderd` can access. If you need to acces the entire file system, you can set the docroot path as `C:\` or leave it empty. The directory you choose is configured in the `aspera.conf` configuration file on the transfer node.
4. Click **Save changes**.

Converting Legacy Nodes

A node running an Aspera transfer product with a version prior to 3.4.6 is considered a legacy node and continues to report to Console using the legacy mechanism. To take full advantage of Console's architecture, upgrade the Aspera transfer products on legacy nodes to the latest version and convert legacy nodes to use the Node API in the Console 3.0+ Node Maintenance page.

Important: Once you convert from a legacy node to a regular node, you cannot revert back to a legacy node.

1. On the node machine, upgrade your Aspera transfer product to its latest version. To check the version of the running transfer product, run the following command:

   ```
   > ascp -A
   ```
2. Configure an administrative account and API user on your node machine.
   - Windows node: "Setting Up a Windows Node" on page 24.
   - OS X node: "Setting Up an OS X Node" on page 25.
   - Linux node: "Setting Up a Linux Node" on page 23.
3. From the Console menu, select **Nodes**. Select the **edit** link for a node that Console shows is a "Legacy Node".
4. Click **Node API** and enter your Node API credentials for the node machine.

   For more information on updating a node's admin credentials, see "Updating a Node's Admin Credentials" on page 28.
5. Click **Convert** and then click **Convert to use Node API**. When prompted by the browser to confirm conversion, click **OK**.

   Note: If you do not see **Convert to use Node API**, make sure you have correctly configured your Node API credentials on the machine and entered them into Console. Edit the node in Console, click **Credentials**, and click **Test Credentials**.
6. Select **Open all** to expand all available configuration options. Clear any overridden values in the **Database** and **Transfer Server** sections.
7. Select **Save changes**.
8. Restart the Aspera Central service on the node.
   - **Windows:**

     Use the Computer Management window. Go to **Manage > Services and Applications >Services**.
   - **Linux:**

     ```
     # service asperacentral restart
     ```

- **OS X:**

  ```
  $ sudo launchctl stop com.aspera.asperacentral
  $ sudo launchctl start com.aspera.asperacentral
  ```

Console now displays your legacy node as a regular node in the node list.

Configuring Virtual Links

Configure Virtual Links (Vlink) on a node to create a "virtual" bandwidth cap for the node. Transfer sessions assigned to the same Vlink take up equal shares of the capped bandiwdth.

1. Go to **Nodes**, find the desired node, and click **edit**. Click **Vlinks > New Vlink**.
2. Enter a number for the **Vlink ID** and name the Vlink. Sessions assigned with the same ID share the same bandwidth cap.
3. Select **True** to activate the Vlink.
4. Enter a value for the capacity. When applying this Vlink to a transfer, the transfer's bandwidth will be restricted by this value.
5. Click **Create**.

After creating a new Vlink, you have the option of configuring the Vlink to run on a schedule by clicking **Edit Time Varying Schedule** and then **New Schedule**. For more information on scheduling Vlinks, see "Scheduling Virtual Links" on page 34.

Scheduling Virtual Links

After creating a new virtual link, you have the option of configuring the Vlink to run on a schedule.

1. Go to **Nodes**, find the desired node, and click **edit**. Click **Vlinks**, find the desired Vlink, and click **edit**.
2. Click **Edit Time Varying Schedule** and click **New Schedule**.

   Configure the following options.

Options	**Description**
On the following days	Select the days or set of days for which the bandwidth rate cap is enforced.
From the following time	Enter a time to start the bandwidth rate cap.
To the following time	Enter a time to stop the bandwidth rate cap.
Set the rate to	Enter a value for the scheduled virtual bandwidth cap. When applying this Vlink to a transfer, the transfer's bandwidth will be restricted by this value based on the configured schedule.

   Note: Overlapping time schedules are not supported. If there are overlapping schedules, they are not accurately reflected in the Vlinks chart, and precedence is indeterminate.
3. Click **Update**.

Setting Up Cloud Storage from the Console

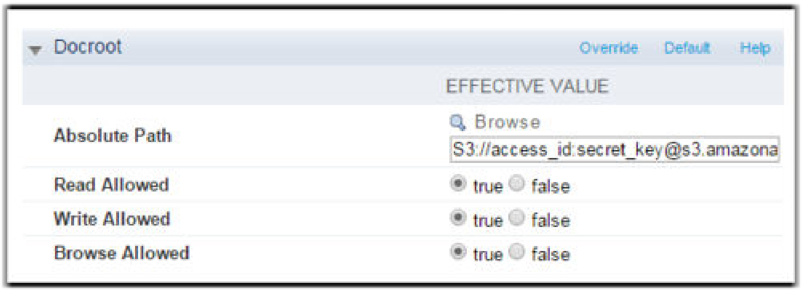
Enabling S3 Storage Using Console

IBM Aspera Console can use S3 storage for a node transfer user by specifying the storage in the user docroot. Use this user to transfer files to and from your S3 storage. The steps below assume the following:

- You have purchased and booted up your Aspera On Demand product.
- You have created an S3 bucket.
- You have permissions to create IAM roles or change the policies of your IAM.
- You know how to SSH as root to your Aspera On Demand instance.

1. In Console, select a node and edit its transfer user from the **Accounts** tab.
2. Expand **Docroot**, click **Override**, and paste the S3 docroot for that user using the following syntax:

   ```
   S3://access_id:secret_key@s3.amazonaws.com/my_bucket/my_path
   ```

   Absolute Path: S3://access_id:secret_key@s3.amazona

   Read Allowed: ● true ○ false

   Write Allowed: ● true ○ false

   Browse Allowed: ● true ○ false

   Use URL encoding for special characters in your S3 Access ID and secret key. For example, encode a slash character (**/**) by replacing it with %2F and encode a plus character (**+**) by replacing it with %2B.

   Click on the **Save Changes** button.

   For more information about setting a user's docroot, see "Editing the User or Group on a Node" on page 32.
3. Restart the Aspera NodeD service on the node.

   SSH into the node and run the following command:

   ```
   # ssh -i identity_file -p 33001 ec2-user@ec2_host_ip
   # service asperanoded restart
   ```

Configure advanced S3 storage settings and test your configuration.

4. Optional: Enable advanced S3 storage settings.

   - Enable Reduced Redundancy Storage (RRS): Append the following to the docroot:

     ```
     ?storage-class=REDUCED_REDUNDANCY
     ```

     For example, enter:

     ```
     S3://access_id:secret_key@s3.amazonaws.com/my_bucket/my_path?storage-
     class=REDUCED_REDUNDANCY
     ```
   - Enable S3 Server Side Encryption (SSE). Append the following to the docroot:

     ```
     ?server-side-encryption=AES256
     ```

     For example, enter:

     ```
     S3://access_id:secret_key@s3.amazonaws.com/my_bucket/my_path?server-side-encryption=AES256
     ```
5. Test your configuration. Perform a test transfer using an Aspera client to the account configured with the S3 docroot. For information on starting a transfer, see "Starting a Simple Transfer" on page 50.

Enabling SoftLayer Storage Using Console

IBM Aspera Console can use SoftLayer storage for a node transfer user. Use this user to transfer files to and from your SoftLayer storage.

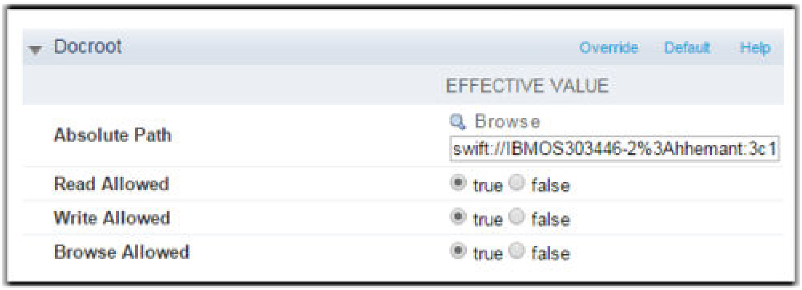
CAUTION: When transferring files larger than 64 MB to SoftLayer storage, an `.aspera-segment` directory is created at the destination. Do not move this directory or modify any files in it. Doing so may cause corruption or loss of data.

1. Go to **Nodes** and click the **edit** button for the node.
2. Go to **Accounts** and click **edit** for the account to configure with SoftLayer access.

   Note: You can also create a new account by clicking on the **Add User** button. For information on how to add a new account, see "Editing the User or Group on a Node" on page 32.
3. Enter the SoftLayer docroot.

   Expand **Docroot**, click **Override**, and paste the SoftLayer docroot for that user using the following syntax:

   ```
   swift://username:api key@Object Storage URI/bucket_name?aspera.swift.endpoint.auth-path=%2Fauth%2Fv1.0
   ```

	EFFECTIVE VALUE
Absolute Path	Browse swift://IBMOS303446-2%3Ahhemant:3c1
Read Allowed	● true ○ false
Write Allowed	● true ○ false
Browse Allowed	● true ○ false

   Use URL encoding for special characters. For example, encode the colon (:) by replacing it with %3A.
4. Click on the **Save Changes** button.
5. Restart **asperanoded** on the node.

   SSH into the node and run the following command:

   ```
   # service asperanoded restart
   ```
6. Test your configuration. Perform a test transfer using an Aspera client to the account configured with the SoftLayer object storage docroot. For information on starting a transfer, see "Starting a Simple Transfer" on page 50.

Managing User Accounts

Accounts and Permissions

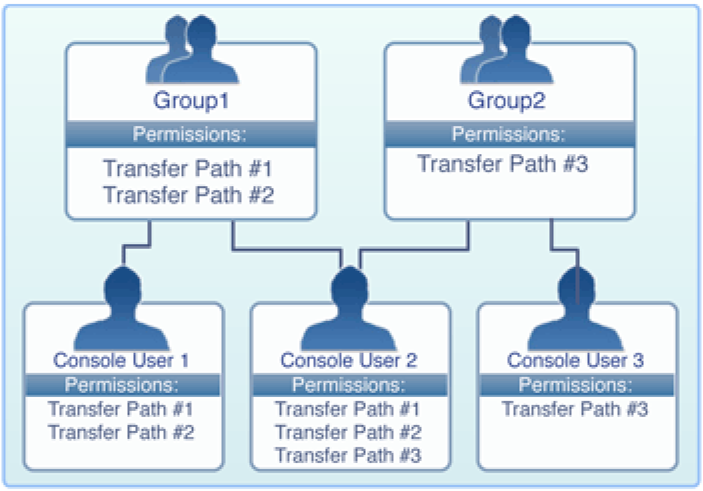
Definition of Terms

- *User*: A user is a Console login account with customizable access permissions.
- *Group*: A group defines the transfer permissions of all its users.
- *Transfer Path*: A transfer path consists of two endpoints, the transfer direction (one-way or two-way), and a set of permissions that authorize starting transfers, monitoring transfers, and enabling email notifications.

Overview

Console uses a combination of groups, transfer paths, and user accounts to manage to user permissions. A user that belongs to a group inherits permissions defined within the groups it belongs to. A group's permissions are defined by its transfer paths. If you have a non-admin user and you want them to be able to see certain transfers, you need to add them to a group. This group must have one or more transfer paths that specify the kinds of transfers that members of the group are allowed to see or control.

Each group can contain one or more transfer paths. In the figure below, Group 1 contains two transfer paths, #1 and #2. A Console user inherits transfer permissions from all of the groups he or she belongs to. For example, Console User 2 belongs to both Group 1 and Group 2, and has the permissions to use Transfer Paths #1, #2, and #3.



Tip: Console administrators are able to view and control all transfers. They automatically inherit permissions of any existing Console groups. They can add, edit, and delete any nodes, Console users, and Console groups.

Default Console Groups

When adding a new node, you have the option of creating three default groups associated with that node.

Group name	Description
Transfer Administrators	The users in this group can monitor, control, and set up email notifications of all transfers on the node. They can start simple and smart transfers between this node and any node, and share smart transfer templates with other users.
Transfer Initiators	The users in this group can start simple and smart transfers between this node and any node.
Transfer Monitors	The users in this group can monitor and set up email notifications of all transfers on this node.

Creating Console Groups and Users

For instructions on creating a new Console group, see "Creating Console Groups" on page 37.

For instructions on creating a new Console user, see "Creating Console Users" on page 39.

Creating Console Groups

Console uses a combination of groups, transfer paths, and user accounts to manage to user permissions. A user that belongs to a group inherits permissions defined within the groups it belongs to. A group's permissions are defined by its transfer paths. If you have a non-admin user and you want them to be able to see certain transfers, you need to add them to a group. This group must have one or more transfer paths that specify the kinds of transfers that members of the group are allowed to see or control.

Tip: Console administrators are able to view and control all transfers. They automatically inherit permissions of any existing Console groups. They can add, edit, and delete any nodes, Console users, and Console groups.

Important: You must first manually add a group to the node OS before you can add it in Console.

1. Go to **Accounts > Groups** and click **New Group**.
2. Enter the group name and a brief description. When finished, click **Create**.

   You are redirected to a page that allows you to configure the group.
3. Click **Add Transfer Path**.

   A transfer path determines a user's permissions to create, initiate, and monitor transfers from one endpoint to another. A transfer path consists of two endpoints, the transfer direction (one-way or two-way), and a set of permissions that authorize starting transfers, monitoring transfers, and enabling email notifications.
4. Select the endpoints for the transfer path.

   For a unidirectional transfer path, set Endpoint 1 as your source endpoint and Endpoint 2 as your destination endpoint. Order does not matter for a bidirectional transfer path. If you specify a node user in an endpoint, users in the group are limited to monitoring only transfers on the node machine that involve the specified node user. An example of such a transfer is a transfer initiated using the specified node user's credentials. Selecting "Any" grants users transfer path permissions to all nodes.

   Important: When you select **Any** as an endpoint and permit users to start simple or smart transfers, users can enter arbitrary addresses for file transfers.
5. Choose the direction of the transfer path.

   A transfer path can be unidirectional or bidirectional.

   - Unidirectional (**to**): Console users can create, initiate, and monitor transfers initiated from Endpoint 1 to Endpoint 2 (depending on the transfer path permissions) but not the other way around.
   - Bidirectional (**to/from**): Console users can create, initiate, and monitor all transfers (depending on the transfer path permissions) between Endpoint 1 or Endpoint 2.
6. Choose the permissions you want to give users in the group.

Item	**Description**
Start Simple Transfers	Users can start a simple transfer.
Start Smart Transfers	Users can start smart transfers.
Create Smart Transfers	Users can to create a smart transfer template.
Share Smart Transfers	Users can to share smart transfer templates with other users.
Control Transfers started by others	Users can control other users' transfers. For example, they can stop, pause, and set the rate of a transfer, and so on.
View Transfers started by others	Users can view other users' transfers on the same transfer paths.
Opt-in to email notifications	Users can enable email notifications for this transfer path.

7. Optional: Enter a description for this transfer path.
8. When finished, click **Create**.

   The Editing Group Details screen displays the new transfer path in the **Transfer Paths** list. To modify or remove the transfer path, click **edit** or **delete**, respectively.
9. Add users to the group.

   Select a Console user from the members drop-down and click **Add**.

   Tip: Alternatively, you can assign group members through user management. See "Creating Console Users" on page 39.

10. Click **Update**.

Creating Console Users

Console user is a Console login account with customizable access permissions. Except for administrator accounts, Console user permissions are managed through group assignment. A Console user inherits permissions from its groups.

Note: Console users are not directly related to the login account to a node.

1. Go to **Accounts > Users** and click **New User**.
2. Enter a login username, the user's first and last name and an email address. Set the user's time zone.

   Important: All activity on the Console is dated according to the user's time zone.
3. Optional: Select **Set password** to create a password for the user account. If you do not set a password, Console generates a temporary password for the account and emails the password to the user.

   Tip: You can change password requirements in the Console Password Options section. Go to **Configuration > Defaults** For more information on password requirements, see "Configuring Console Defaults" on page 102.
4. Optional: Disable user login by clearing **Active (allow user to login)**.

   If you wish to finish setting permissions for the user account before allowing the user to log in, disable the account by clearing **Active (allow user to login)**. To re-enable the account, return to these settings and select **Active (allow user to login)**. User login is enabled by default.
5. Optional: Disable reporting features for the user by clearing **Reports Allowed**.
6. When finished, click **Create**.

   The system sends an account creation notification email to the designated email with the account's username and password. If you do not set a password, Console generates a temporary password for the account and include that in the email.

The following step is only applicable when creating non-admin users. All admin users have full permissions to all groups and transfer paths. After creating a non-admin user, Console redirects you to the user permissions page.

7. Assign the user to Console groups.

   Assign the user to groups with the desired transfer-path permissions. To assign the user to a group, select a group from the drop-down menu and click **Add**. You can review the Console user's transfer permissions in a table listing all transfer paths accessible by this user

Once the Console user account is created, users can log in to Console with the proper account credentials. To deactivate this account or make other changes to it, go to **Accounts > Users**. Locate the account you want to change in the list of all Console users.

- To deactivate or reactivate the account, change it to a Console administrator, or modify any of the basic account information, click **edit**.
- To modify transfer permissions and group membership, click **permissions**.
- To remove a Console user from the system, click **delete**.

Monitoring Console

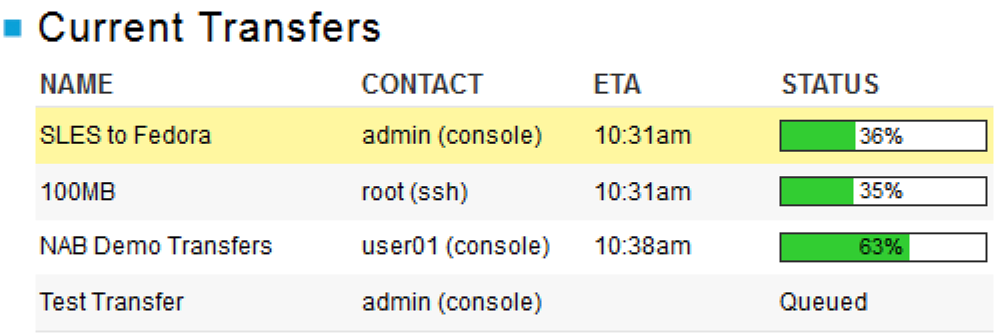
The Console Dashboard

The Dashboard provides a quick overview of all transfer activities and the statuses of nodes for which you have monitoring permissions. It gives continuous updates and helps identify transfer and node problems.

Go to **Dashboard**. The Dashboard contains the following six panels:

Current Transfers

Current Transfers lists up to ten ongoing transfers on all monitored nodes. To view all active transfers, click the **Current Transfers** header.

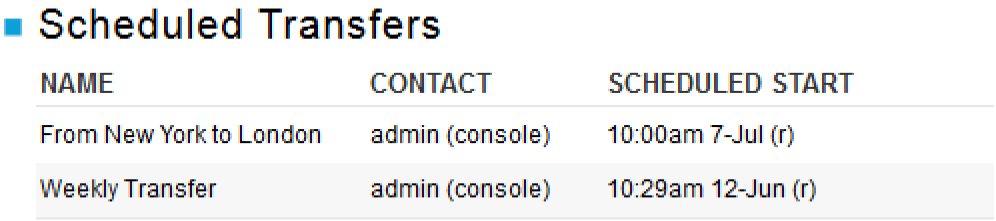


Current Transfers

NAME	CONTACT	ETA	STATUS
SLES to Fedora	admin (console)	10:31am	36%
100MB	root (ssh)	10:31am	35%
NAB Demo Transfers	user01 (console)	10:38am	63%
Test Transfer	admin (console)		Queued

Scheduled Transfers

Scheduled Transfers lists up to ten scheduled transfers on all monitored nodes. To view all scheduled transfers, click the **Scheduled Transfers** header.

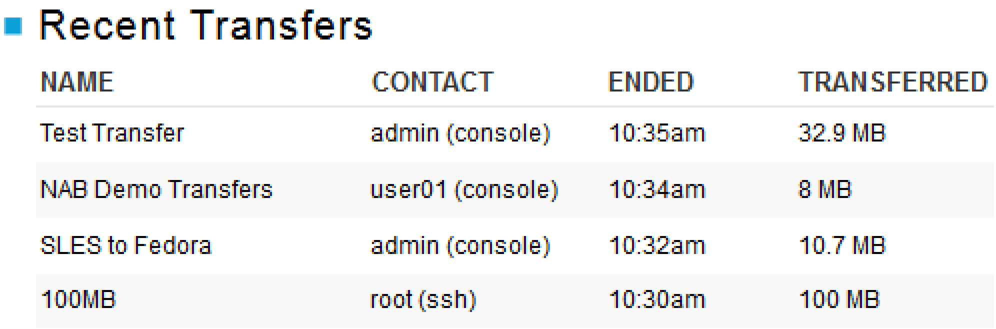


Scheduled Transfers

NAME	CONTACT	SCHEDULED START
From New York to London	admin (console)	10:00am 7-Jul (r)
Weekly Transfer	admin (console)	10:29am 12-Jun (r)

Recent Transfers

Recent Transfers lists up to ten recent transfers on all managed nodes. To view all recent transfers, click the **Recent Transfers** header.

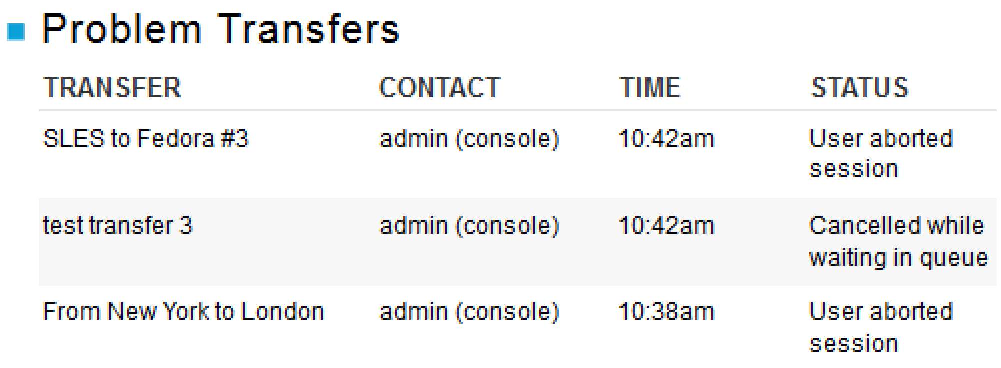


Recent Transfers

NAME	CONTACT	ENDED	TRANSFERRED
Test Transfer	admin (console)	10:35am	32.9 MB
NAB Demo Transfers	user01 (console)	10:34am	8 MB
SLES to Fedora	admin (console)	10:32am	10.7 MB
100MB	root (ssh)	10:30am	100 MB

Problem Transfers

Problem Transfers lists up to ten transfers with errors on all managed nodes. To view all transfers with errors, click the **Problem Transfers** header.



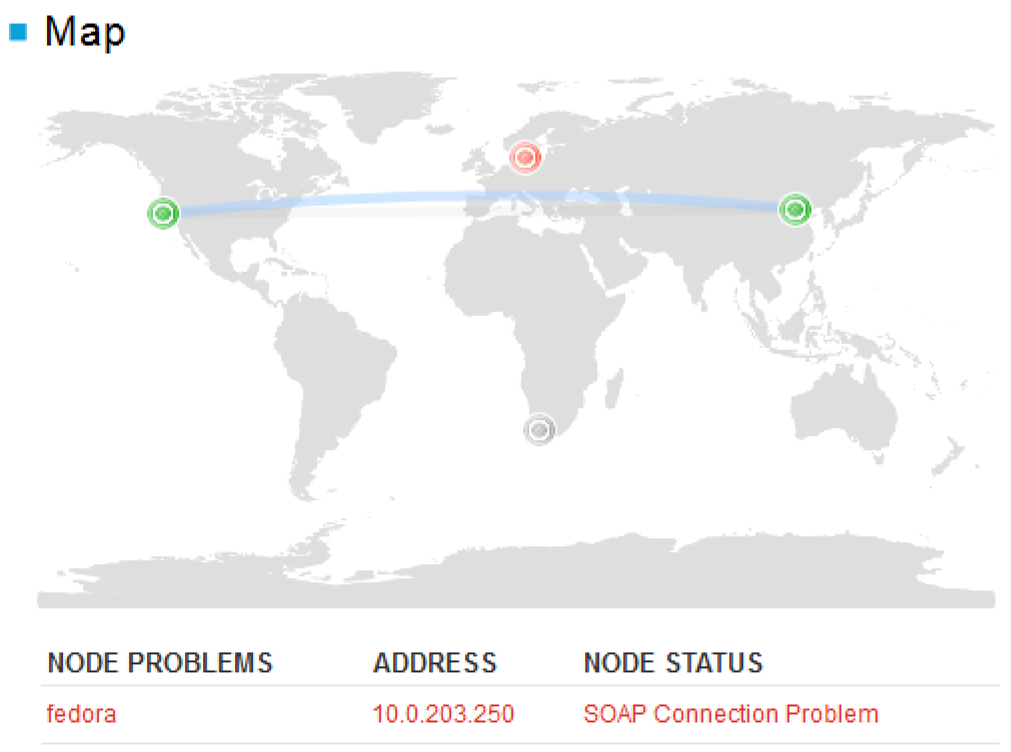
Problem Transfers

TRANSFER	CONTACT	TIME	STATUS
SLES to Fedora #3	admin (console)	10:42am	User aborted session
test transfer 3	admin (console)	10:42am	Cancelled while waiting in queue
From New York to London	admin (console)	10:38am	User aborted session

Map

The map shows the status of all your monitored nodes and shows the transfers between them. If a node fails, the icon becomes red in the map, and the node and the problem are listed in the table below the map.

Nodes are not automatically added to maps. They must be configured. For more information, see "Configuring the Map" on page 45.



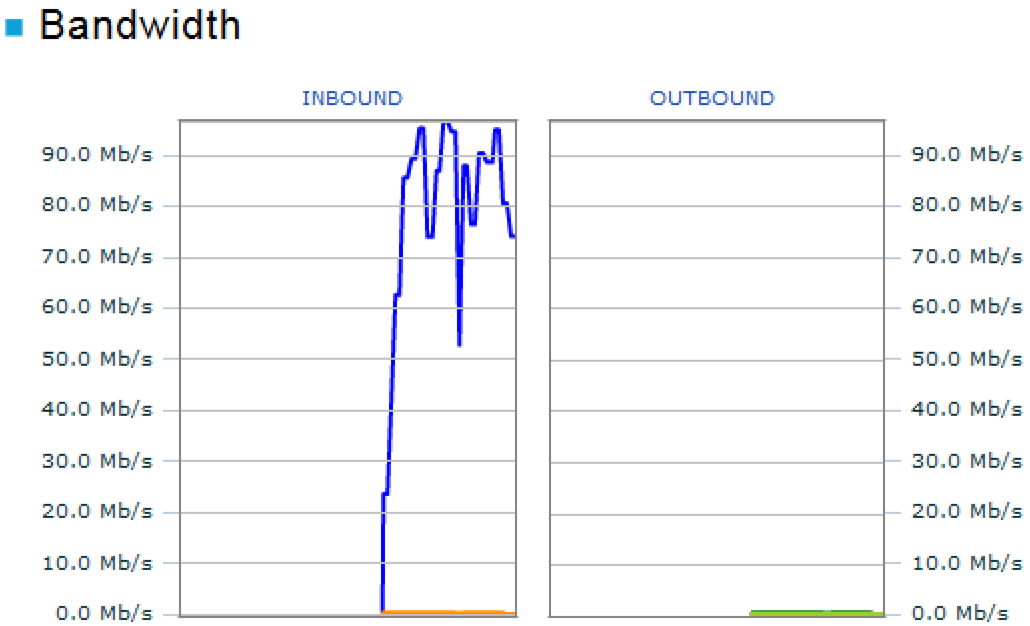
Map

NODE PROBLEMS	ADDRESS	NODE STATUS
fedora	10.0.203.250	SOAP Connection Problem

Note: You can choose to hide or display the map and bandwidth chart by clicking the blue arrow (▶) next to the map.

Bandwidth

The Bandwidth chart shows bandwidth usage of your monitored nodes. If you select one or more nodes on the map, the chart shows the cumulative bandwidth of the selected nodes.

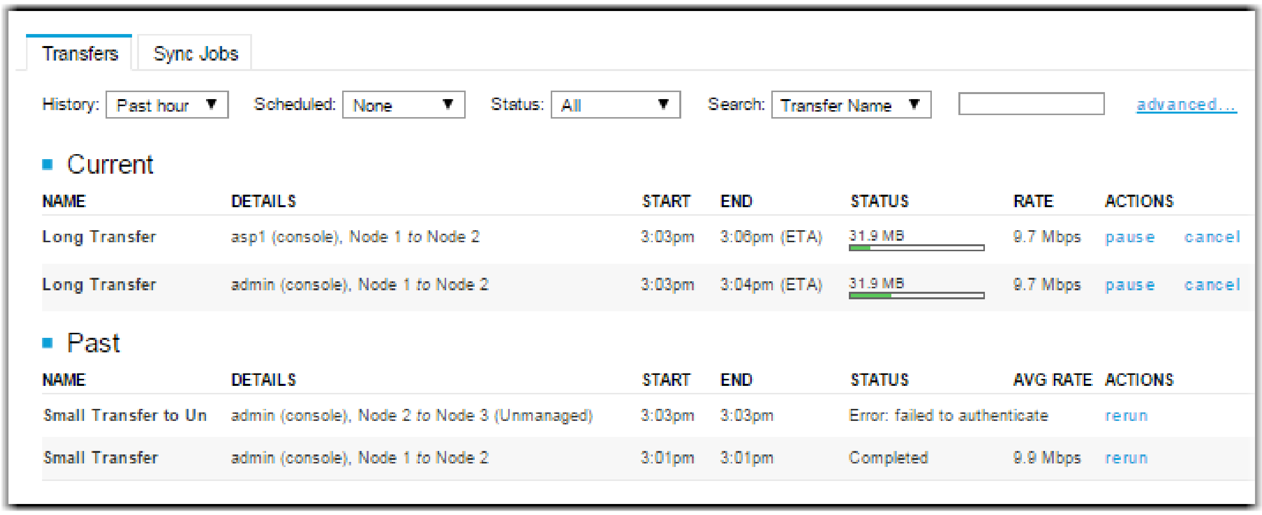


Note: You can choose to hide or display the map and bandwidth chart by clicking the blue arrow (▶) next to the map.

The Activity Overview

The Activity Overview page lists all transfers on all managed nodes. View the Activity Overview page by going to **Activity**. You can narrow down the list with the filter and advance into a transfer's session detail page. The Activity Overview screen displays the following information:

Item	**Description**
NAME	The transfer's name.
DETAILS	The transfer initiator, source, and destination.
START	This transfer's start time.
END	The estimated time of arrival, or the transfer completion time.
STATUS	Current status of this transfer.
AVG RATE	The transfer rate of the active transfer, or the average rate of a past transfer.
ACTIONS	Show all available actions. For example, **pause** and **cancel** for a running transfer or **rerun** for a past transfer.



History: Past hour ▼ Scheduled: None ▼ Status: All ▼ Search: Transfer Name ▼

■ Current

NAME	DETAILS	START	END	STATUS	RATE	ACTIONS
Long Transfer	asp1 (console), Node 1 *to* Node 2	3:03pm	3:06pm (ETA)	31.9 MB	9.7 Mbps	pause cancel
Long Transfer	admin (console), Node 1 *to* Node 2	3:03pm	3:04pm (ETA)	31.9 MB	9.7 Mbps	pause cancel

■ Past

NAME	DETAILS	START	END	STATUS	AVG RATE	ACTIONS
Small Transfer to Un	admin (console), Node 2 *to* Node 3 (Unmanaged)	3:03pm	3:03pm	Error: failed to authenticate		rerun
Small Transfer	admin (console), Node 1 *to* Node 2	3:01pm	3:01pm	Completed	9.9 Mbps	rerun

The Current panel lists all currently active transfers, including running and queued transfers. The Past panel shows previous transfers, including those that were completed, canceled, or those that generated errors.

The filter options on the top can be used to narrow down the list.

Item	Description
History	Select the time frame to display the started transfers.
Scheduled	Select the time frame to display the scheduled transfers.
Status	Select a specific transfer status to display.
Search	Search for keywords in transfer sessions.

You can also perform an advanced search by clicking on the **advanced** link. For more information on searching, see "Search for a Transfer" on page 46.

Transfer Details

Overview

Details about a particular transfer can be accessed by clicking on a transfer shown in listings of past, current, and scheduled transfers. These lists can be found in three locations:

- The **Activity Overview** page
- The Console **Dashboard**
- The Managed Node Detail page (the specific node from **Nodes** in the Console menu)

Ongoing Transfers

For an ongoing transfer, the Session Detail page provides the transfer monitor that displays current transfer status. You can control the transfer through the options shown at the top of the graph.

Important: The failed files counter may count "directories" if the network failed at some point or the user cancelled the transfer.

Finished or Failed Transfers

For a finished or failed transfer, the Session Detail page provides detailed information about the transfer's state, endpoints, and statistics.

The Session Files panel lists all files being transferred in this session. Click on a file to review its information. You can use the search box to show only specific files or groups of files.

Note: When searching for files, "*" is not a wildcard. Any string you enter is treated as a "search within". In other words, the string "foo" will match "123foo", "foo456", and "123foo456".

Console also lets you monitor notification information that includes messages about transfer starts, successes, errors, and what notification templates were used under the Statistics column. Next to **Notifications** is a link describing some combination of **start**, **success**, and **error** depending on what notifications were configured for the transfer, or **None** if no notifications were configured. Select the link to see the Session Notifications page. For more information, see "View Email Notification Statistics" on page 17.

Multiple-Session Transfer

A multiple-session transfer is a smart transfer with more than one destination. In the Activity Overview page, clicking on a multiple-session transfer reveals all sessions in the transfer. To drill down to the particulars of each session, click the **Session Detail** button to open its Session Detail page.

Monitoring File Validation

Console reports on the validation states of transferred files on nodes that are validated by IBM Aspera Validator. Validator is installed separately and has its own documentation. You can find the *IBM Aspera Validator Admin Guide* on the IBM Aspera Faspex downloads page: https://downloads.asperasoft.com/en/downloads/6.

Validation Status Monitoring

If configured to validate transferred files for a node, Validator updates the validation status on the node, which Console then reports. The possible validation states are:

- `to be validated`
- `validating`
- `completed`

Console displays the validation state for each file. Go to the "Transfer Details" on page 43 page to monitor the validation status.

Optimizing Node Reporting

By default, managed nodes report the filenames of the first 1,000,000 files of a transfer to Console. However, reporting this many filenames, especially if multiple managed nodes are reporting transfer sessions of several thousands of files, can slow transfers.

You can decrease the number of filenames that are reported by each endpoint by configuring all managed nodes. This configuration affects all transfer reporting, including transfers initiated by Aspera Hotfolders and Aspera Sync. The total number of files transferred, completed, failed, and skipped are still reported, but filenames are logged only for the files up to the specified number.

Perform the following steps on every managed node:

1. Decrease the number of files that are reported.

   Run the following command:

   ```
   > asconfigurator -x "set_central_server_data;files_per_session,1000"
   ```

   With this setting, only the first 1000 filenames are reported to Console and logged.
2. Restart Aspera Central and Aspera NodeD to activate your changes.

   You can restart the Aspera Central from the Computer Management window. Go to **Control Panel > Administrative Tools > Computer Management > Services and Applications > Services**, click **Aspera Central**, and click **Restart**.

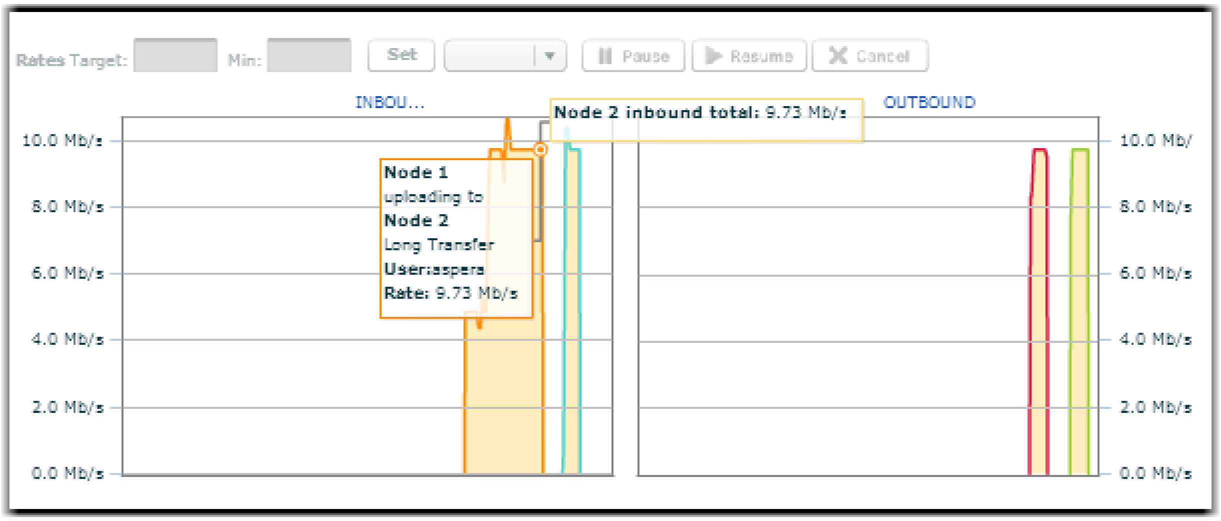
Go to **Control Panel > Administrative Tools > Computer Management > Services and Applications > Services**, click **Aspera NodeD**, and click **Restart**.

Monitoring Nodes

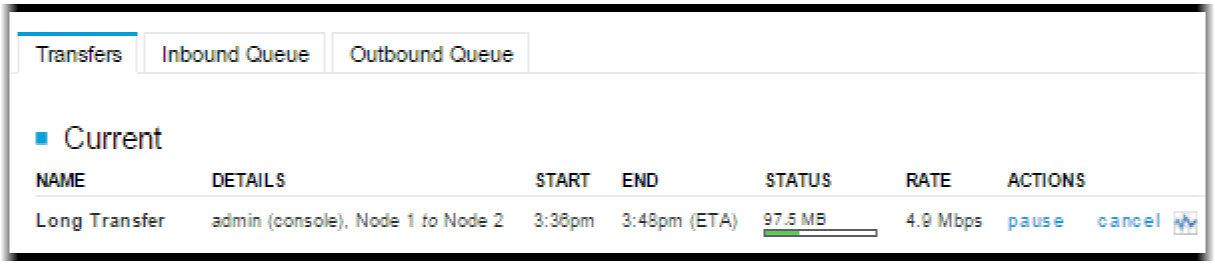
You can monitor the node status and manage the transfers on a node. Navigating to **Nodes** from the Console menu will bring you to the list of managed nodes. To view a list of unmanaged nodes, click the **List Unmanaged Nodes** button. To monitor a node, click on the node.

Monitor Transfers on a Node

On the Node Detail page, the transfer chart shows all inbound and outbound transfers on this node. To control a transfer session, select a session from the graph, and use the control options above the graph to control it.



The table lists all sessions on this node. Use **Pause** and **Cancel** to control an ongoing session.



Current

NAME	DETAILS	START	END	STATUS	RATE	ACTIONS	
Long Transfer	admin (console), Node 1 to Node 2	3:36pm	3:48pm (ETA)	97.5 MB	4.9 Mbps	pause	cancel

Configuring the Map

You can configure Console to display the locations of your nodes on the dashboard map.

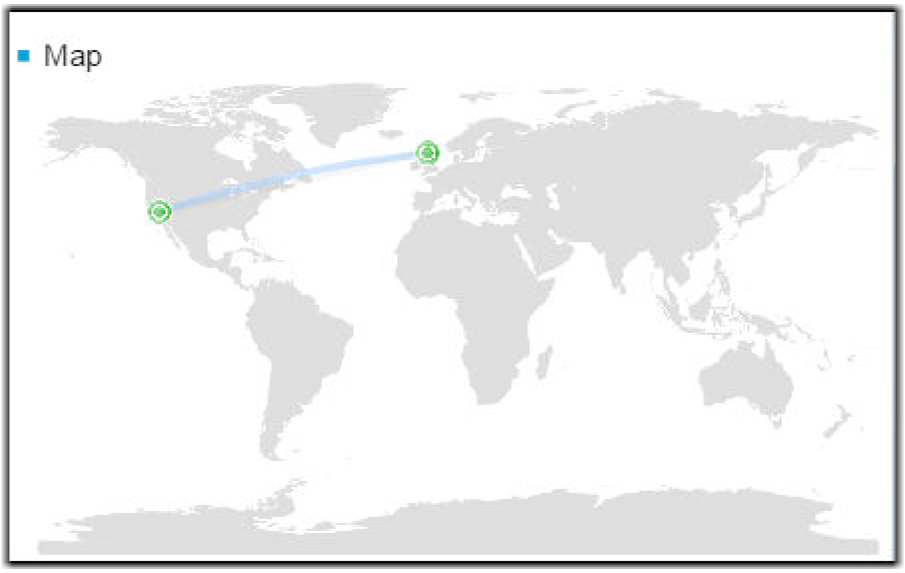
1. Go to **Configuration > Map**.
2. Select or upload a map image for use on the Console dashboard.

   - Upload a new map image: Click **Upload Map File**. Upload the file and then click **select**. For best results, Aspera strongly recommends using an image with a ratio of 16:9 (for example, 800 x 450).
   - Select existing map image: Choose one of two default map images or any previously uploaded image as the dashboard map by clicking the **select** link.

   Note: To delete a map image you have uploaded, click the **delete** link.
3. Configure node to show on map.

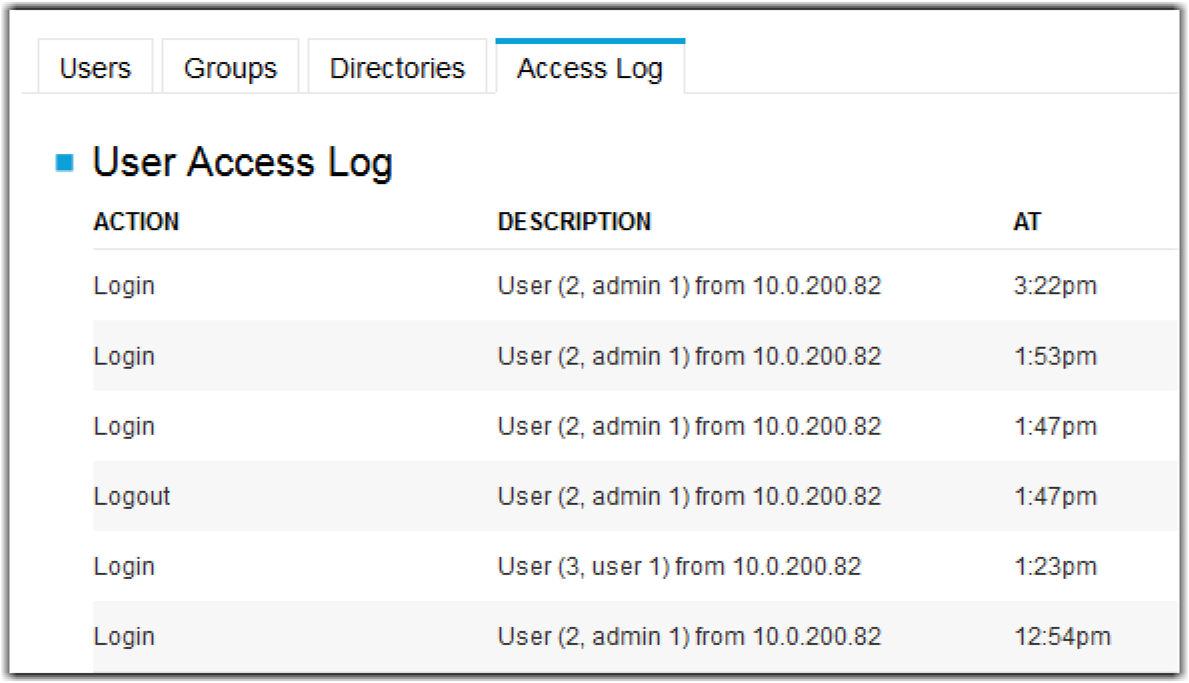
   Edit your node and click the **Map** tab. Select **Show on Map**. Click and drag the green icon to its proper location on the map.

The configured nodes appear on the map on the Dashboard. Ongoing transfers between nodes are represented by a line between the nodes.



Access Logs

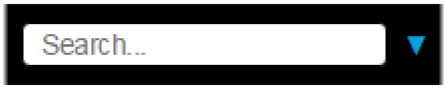
Once you have created accounts for Console users, you can monitor their activity from the **Accounts > Access Log** tabs. The User Access Log displays user logins and logouts, concurrent logins and session timeouts.



User Access Log

ACTION	DESCRIPTION	AT
Login	User (2, admin 1) from 10.0.200.82	3:22pm
Login	User (2, admin 1) from 10.0.200.82	1:53pm
Login	User (2, admin 1) from 10.0.200.82	1:47pm
Logout	User (2, admin 1) from 10.0.200.82	1:47pm
Login	User (3, user 1) from 10.0.200.82	1:23pm
Login	User (2, admin 1) from 10.0.200.82	12:54pm

Search for a Transfer



You can search for a transfer from any page in Console by using the search bar in the top right corner of the page. If you want to refine your search, you can access the Advanced Search dialog by selecting the blue drop-down arrow next to the search bar.

Console will search all transfers within the last 24 hours for transfers that match the search criteria.

For more information about the advanced search form, see "Advanced Search" on page 114.

Monitoring Sync Jobs

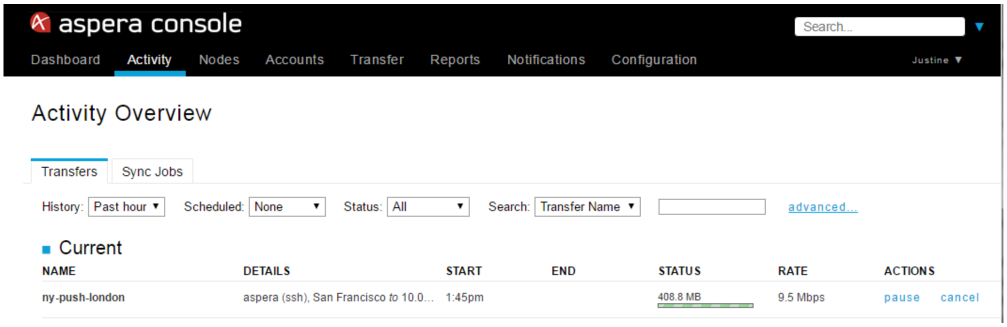
Enabling Sync Client Node Reporting

The instructions below describe how to configure the IBM Aspera Sync client reporting to Console when the client is a managed node.

Note: If both client and server involved in a Sync job are Console managed nodes, then reporting can come from either node, both nodes, or neither node using the `async_activity_logging` setting in `aspera.conf`. For example, to receive reporting from both the client and server, set `async_activity_logging` to `true` on both. The server is reported as the local host. For more information on server reporting, see "Enabling Async Server Node Reporting " on page 48.

Transfer Reporting

The Sync client reports transfers associated with Sync jobs if `<async_management_activity_logging>` is set to `true` in `aspera.conf`, which is the default configuration. The transfer name is listed as the Sync session name. In the example below, the Sync session "ny-push-london" is reported under **Transfers**.



Activity Overview

History: Past hour ▼ Scheduled: None ▼ Status: All ▼ Search: Transfer Name ▼

■ Current

NAME	DETAILS	START	END	STATUS	RATE	ACTIONS
ny-push-london	aspera (ssh), San Francisco *to* 10.0...	1:45pm		408.8 MB	9.5 Mbps	pause cancel

This setting can be modified by running the following command:

```
> asconfigurator -x "set_client_data;async_management_activity_logging,value"
```

Setting the value to `false` disables reporting transfers associated with Sync jobs to Console. You do not need to restart the Aspera Node API service for the new setting to be activated.

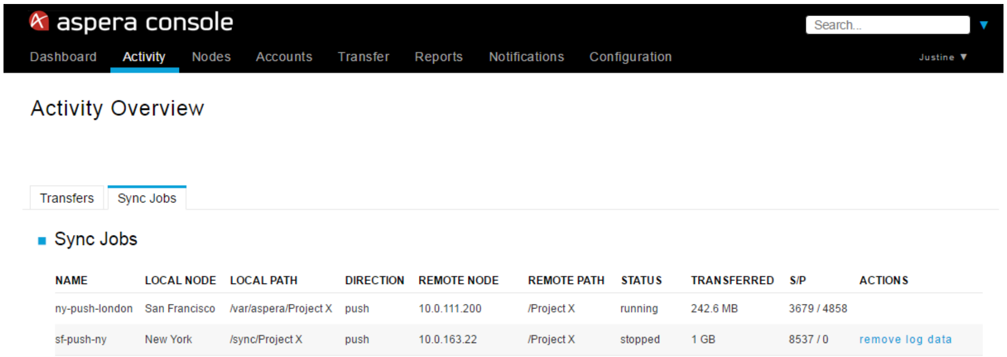
If you are syncing empty directories then no transfers are reported; the creation of the empty directories at the destination is not reported as a transfer.

Sync Job Reporting

The Sync client reports Sync jobs to Console if `<async_kvstore_activity_logging>` is set to `true` in `aspera.conf`. The default value is `false`, such that no Sync jobs are reported. To modify this setting, run the following command:

```
> asconfigurator -x "set_client_data;async_kvstore_activity_logging,value"
```

Sync jobs are listed under **Activity > Sync Jobs**.



Sync Jobs

NAME	LOCAL NODE	LOCAL PATH	DIRECTION	REMOTE NODE	REMOTE PATH	STATUS	TRANSFERRED	S/P	ACTIONS
ny-push-london	San Francisco	/var/aspera/Project X	push	10.0.111.200	/Project X	running	242.6 MB	3679 / 4858	
sf-push-ny	New York	/sync/Project X	push	10.0.163.22	/Project X	stopped	1 GB	8537 / 0	remove log data

You can click on a Sync job to view more details about the session, including the endpoints, the status of individual file transfers, and the transfer rate. For more information, see "Monitor Sync Jobs" on page 48.

Enabling Async Server Node Reporting

By default, Console does not report transfers with a server that are associated with IBM Aspera Sync jobs or the Sync job information. If the server is a Console managed node and is runnning IBM Aspera High-Speed Transfer Server version 3.1.5 or later, then it can be configured to report transfers and Sync jobs.

1. Enable server activity logging for Sync jobs.

   Run the following asconfigurator command to enable activity logging for Sync jobs:

   ```
   > asconfigurator -x "set_node_data;async_activity_logging,true"
   ```

   This command adds the following text to the `<default>` section of the `aspera.conf` file, located at: `C:\Program Files\Aspera\Enterprise Server\etc\aspera.conf`.

   ```
   <CONF version="2">
   ...
   <default>
       ...
       <async_activity_logging>true</async_activity_logging>
       ...
   </default>
   ...
   </CONF>
   ```

2. Restart the Aspera Node API service to activate your changes.

   Go to **Control Panel > Administrative Tools > Services**, right-click **Aspera NodeD** and select **Restart**.

3. Confirm Sync jobs and transfers associated with them are reported in Console.

   After initiating a sync session, go to Console and go to **Activity > Sync Jobs** page to monitor the job.

   Note: Sync job reporting (from the **Sync Jobs** screen) may not appear immediately.

Monitor Sync Jobs

To monitor Aspera Sync jobs, you must first configure Console to poll the Node API. For more information, see "Enabling Async Server Node Reporting " on page 48 to configure server reporting, and "Enabling Sync Client Node Reporting" on page 47 to configure client reporting.

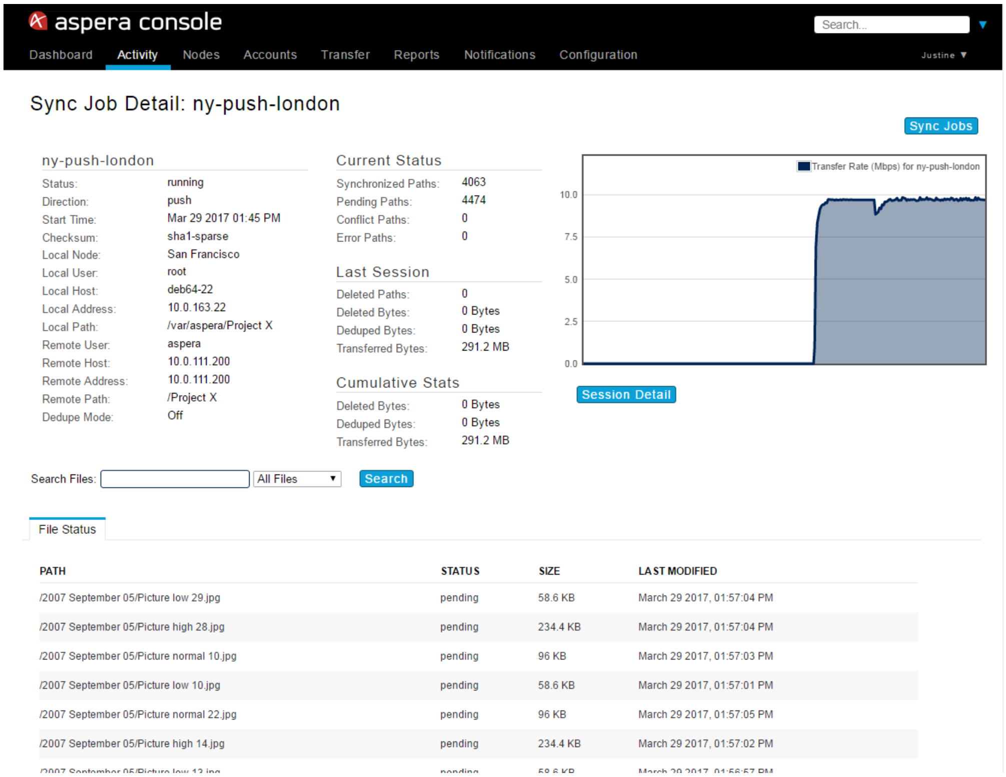
Once you have initiated a Sync transfer, you can monitor it by going to **Activity > Sync Jobs**. This shows a list of active and recently completed Sync jobs. You can also remove log data from the Sync Jobs page by clicking **remove log data**.

Note: Sync jobs may not appear immediately.

From the Sync Jobs table, you can view a job's transfer details by clicking the corresponding row. The job's transfer details page displays the following:

- Local and remote server details
- Session statistics including the number of paths that are synced, pending, conflicted, deleted, or in error state
- Transfer rate graph, which is only active during the transfer
- **Remove log data** button, which deletes the job's log data from the Console and Aspera Sync databases.

The example below shows a running Sync job.



Sync Job Detail: ny-push-london

ny-push-london

Status: running
Direction: push
Start Time: Mar 29 2017 01:45 PM
Checksum: sha1-sparse
Local Node: San Francisco
Local User: root
Local Host: deb64-22
Local Address: 10.0.163.22
Local Path: /var/aspera/Project X
Remote User: aspera
Remote Host: 10.0.111.200
Remote Address: 10.0.111.200
Remote Path: /Project X
Dedupe Mode: Off

Current Status

Synchronized Paths: 4063
Pending Paths: 4474
Conflict Paths: 0
Error Paths: 0

Last Session

Deleted Paths: 0
Deleted Bytes: 0 Bytes
Deduped Bytes: 0 Bytes
Transferred Bytes: 291.2 MB

Cumulative Stats

Deleted Bytes: 0 Bytes
Deduped Bytes: 0 Bytes
Transferred Bytes: 291.2 MB

Search Files: All Files ▼

PATH	STATUS	SIZE	LAST MODIFIED
/2007 September 05/Picture low 29.jpg	pending	58.6 KB	March 29 2017, 01:57:04 PM
/2007 September 05/Picture high 28.jpg	pending	234.4 KB	March 29 2017, 01:57:04 PM
/2007 September 05/Picture normal 10.jpg	pending	96 KB	March 29 2017, 01:57:03 PM
/2007 September 05/Picture low 10.jpg	pending	58.6 KB	March 29 2017, 01:57:01 PM
/2007 September 05/Picture normal 22.jpg	pending	96 KB	March 29 2017, 01:57:05 PM
/2007 September 05/Picture high 14.jpg	pending	234.4 KB	March 29 2017, 01:57:02 PM
/2007 September 05/Picture low 13.jpg	pending	58.6 KB	March 29 2017, 01:56:57 PM

Transferring Files

Console Transfer Special Conditions

Console transfers have restrictions for some endpoint combinations. For more information about endpoints, see "Understanding Endpoints" on page 29.

Incompatible Endpoint Combinations

Console does not support transfers with the following endpoint combinations:

- Transferring between SSH and Shares endpoints
- Transferring between two Shares endpoints

Restricted Endpoint Combinations

The following four endpoint combinations require transfers to be initiated from a specific endpoint:

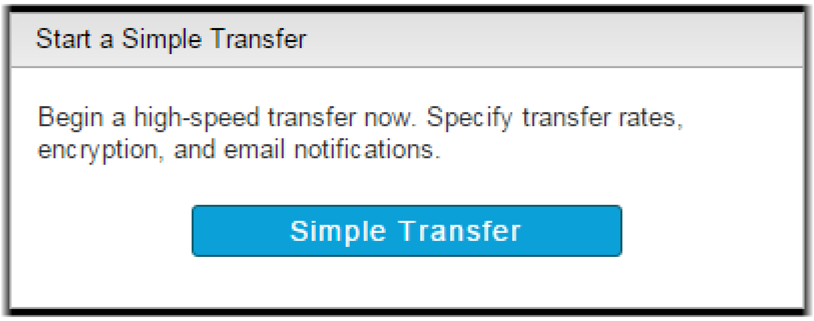
Endpoint Combination	Initiating Endpoint
SSH and Node API endpoints	The Node API endpoint must be the initiator and a managed node

Endpoint Combination	Initiating Endpoint
SSH and Access Key endpoints	The SSH endpoint must be the initiator and a managed node
Shares and Node API endpoints	The Node API endpoint must be the initiator and a managed node
Shares and Access Key endpoints	The Access Key endpoint must be the initiator and a managed node

Starting a Simple Transfer

Console can be used to initiate transfers between nodes when the Console user has the permission to start transfers. Console provides two types of transfer methods: simple transfers and smart transfers. Simple transfers are one-time transfer sessions that require entering all transfer information. Smart transfers are reusable templates with saved transfer settings.

1. Go to **Transfer**.
2. Click **Simple Transfer**.

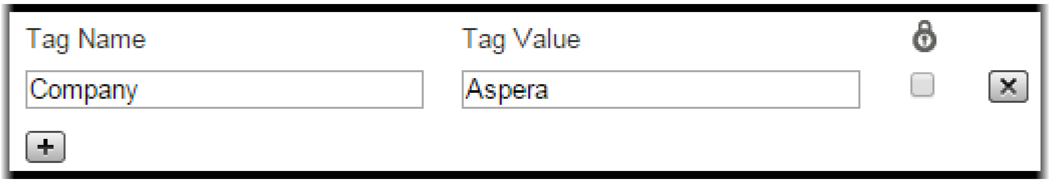


Start a Simple Transfer

Begin a high-speed transfer now. Specify transfer rates, encryption, and email notifications.

3. Enter the transfer name and optional comments. The name and comments can be helpful if you want to search for this transfer later.
4. Optional: Add new tags or modify existing tags.

   Click the ⊞ button to add a new tag. Enter the tag name and the tag value. Click the ☒ button to delete an existing tag. Select the 🔒 button to prevent a user from changing or deleting the locked tag when starting this transfer. For more information about tags, see "Working with Tags" on page 106.



Tag Name	Tag Value	🔒	
Company	Aspera	□	☒
⊞			

5. In the **Source** section, click the **Connect** drop-down menu and select the source node, cluster, or saved endpoint.
   - Node: A node is listed as the node name (by default, its IP address) and IP address. Select the **Endpoint type** from the drop-down menu and enter your credentials or select your SSH key.
   - Cluster: A cluster is listed as the domain name. Select the **Endpoint type** from the drop-down menu and enter your credentials.
   - Endpoint: A saved endpoint is listed as *login@address* and is associated with login credentials for the username or access key. Selecting a saved endpoint does not prompt you for credentials.
   - Any: Choosing **Any** allows the user to create a new endpoint instead of choosing from the list.
6. Select content to transfer by clicking **Browse**, selecting the content, and clicking **Add**.

Note: When browsing the node, you can narrow your search by applying a filter. When specifying a filter, the asterisk (*) is not a wildcard. Any string you enter as a filter is treated as a "search within". In other words, the string "foo" matches "123foo", "foo456", and "123foo456".

By default, the parent folders of the selected files and folders are not transferred. If a source item is a file, then *only* the file is transferred. If a source item is a folder, then the folder and its entire contents are transferred. For example, if the source path is `aspera/tmp/sent_files`, the only folder that will be transferred to the destination is the `sent_files` folder. Neither `/aspera` nor `/tmp` appear at the destination location.

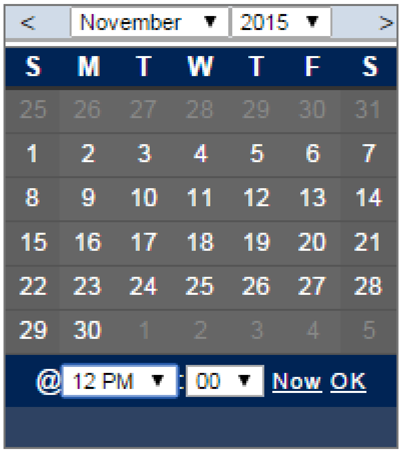
To transfer only the contents of a selected folder, select **Specify base for source path(s)** and enter the filepath to the folder. For example, if the source folder is `aspera/tmp/sent_files` and you specify that same path as the base for source paths, the contents of `/sent_files` is transferred to the destination directory as separate items that are not contained in a `/sent_files` folder.

For more information on specifying a base path, see "Specify Base for Source Path" on page 150.

7. In the **Destination** section, click the **Connect** drop-down menu and select the source node, cluster, or saved endpoint.
    - Node: A node is listed as the node name (by default, its IP address) and IP address. Select the **Endpoint type** from the drop-down menu and enter your credentials or select your SSH key.
    - Cluster: A cluster is listed as the domain name. Select the **Endpoint type** from the drop-down menu and enter your credentials.
    - Endpoint: A saved endpoint is listed as *login@address* and is associated with login credentials for the username or access key. Selecting a saved endpoint does not prompt you for credentials.
    - Any: Choosing **Any** allows the user to create a new endpoint instead of choosing from the list.
8. Click**Browse**, select the destination directory, and click **Add**.
9. Optional: Configure settings in the **More Options** section.

    Click the toggle arrow next to each section to view settings.

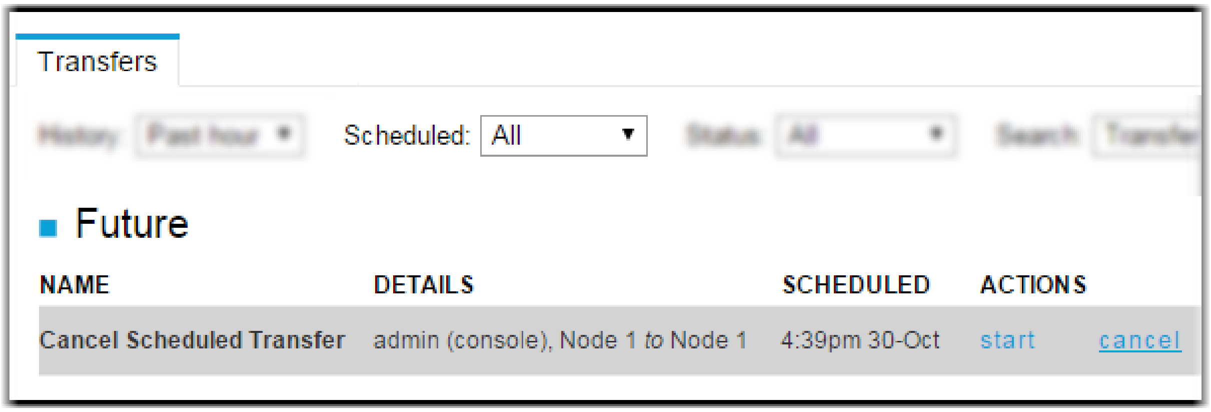
Section	Description
Connection	Configure *fasp* settings.
Transfer	Configure transfer rates and policies.
Security	Encrypt the transfer.
File Handling	Configure source file attributes, archive source files after transfer, and set filters for source files.
Notifications	Configure email notification options. For more information on email notifications, see "Configuring Email Notifications" on page 14.
Advanced	Configure transfer initiator, *fasp* MTU, and read and write block sizes on source and destination nodes.
Transfer Time	Schedule your transfer to run **Now** or **Later**. If you choose **Later**, click the 📅 button and choose the date and time you want the transfer to run.

Section	Description
	 < November ▼ 2015 ▼ > S M T W T F S 25 26 27 28 29 30 31 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 1 2 3 4 5 @ 12 PM ▼ : 00 ▼ Now OK

For information on these options, see "Simple Transfer Options" on page 145.

10. Click **Transfer** to start the transfer (or **Schedule** if you set a transfer time).

    Note: You can cancel scheduled simple transfers by going to **Activity > Transfers**. Click the **Scheduled** drop-down menu and select **All**. In the row for the transfer, click **Cancel**.



Transfers

Scheduled: All ▼

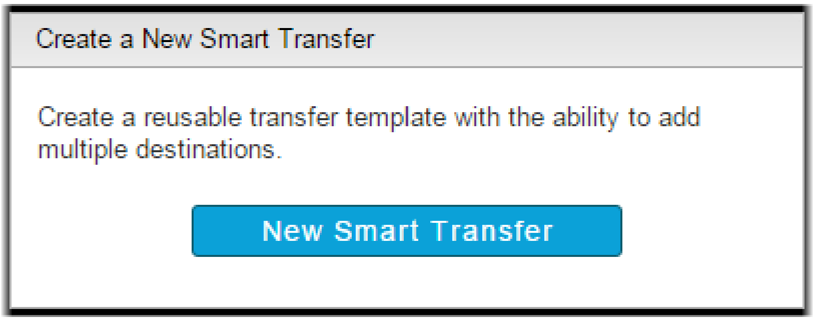
Future

NAME	DETAILS	SCHEDULED	ACTIONS	
Cancel Scheduled Transfer	admin (console), Node 1 *to* Node 1	4:39pm 30-Oct	start	cancel

Creating a Smart Transfer

Console can be used to initiate transfers between nodes when the Console user has the permission to start transfers. Console provides two types of transfer methods: simple transfers and smart transfers. Simple transfers are one-time transfer sessions that require entering all transfer information. Smart transfers are reusable templates with saved transfer settings.

1. To create a smart transfer template, go to **Transfer > New Smart Transfer**.



Create a New Smart Transfer

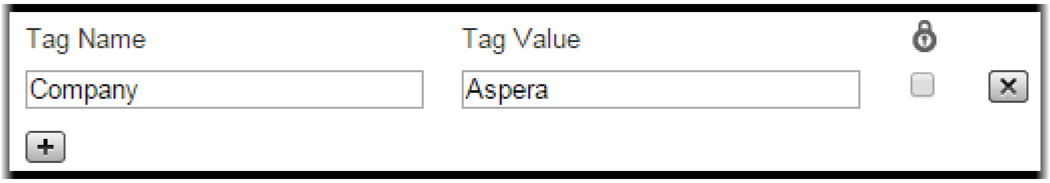
Create a reusable transfer template with the ability to add multiple destinations.

2. Enter a transfer name.
3. Optional: Select **Share this smart transfer** to share this smart transfer with any user who has permissions for the transfer paths.

For more information on sharing smart transfers, see "Sharing a Smart Transfer" on page 57.

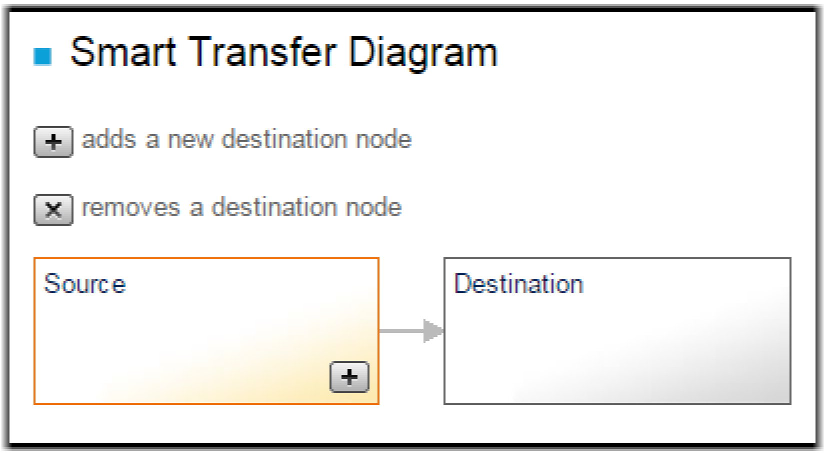
4. Optional: Select **Allow changes to transfer settings at submit time** to allow the user who starts this smart transfer to change settings before submitting the transfer request.
5. Optional: Add new tags or modify existing tags.

   Click the + button to add a new tag. Enter the tag name and the tag value. Click the ✕ button to delete an existing tag. Select the 🔒 button to prevent a user from changing or deleting the locked tag when starting this transfer. For more information about tags, see "Working with Tags" on page 106.



Tag Name	Tag Value	🔒	
Company	Aspera	□	✕

The highlighted box in the Smart Transfer Diagram indicates whether you are configuring the Source or Destination for the smart transfer. Make sure **Source** is selected.



Smart Transfer Diagram

+ adds a new destination node

✕ removes a destination node

6. Select the source node or saved endpoint from the **Connect** drop-down menu.
   - Node: A node is listed as the node name (by default, its IP address) and IP address. Select the **Endpoint type** from the drop-down menu and enter your credentials or select your SSH key.
   - Cluster: A cluster is listed as the domain name. Select the **Endpoint type** from the drop-down menu and enter your credentials.
   - Endpoint: A saved endpoint is listed as *login@address* and is associated with login credentials for the username or access key. Selecting a saved endpoint does not prompt you for credentials.
   - Any: Choosing **Any** allows the user to create a new endpoint instead of choosing from the list.
7. Choose your **Source** directory.

   Click **Choose Source Directory** to browse the node for the directories and files you want to transfer. Console displays the source path you choose once you have chosen your source directory.

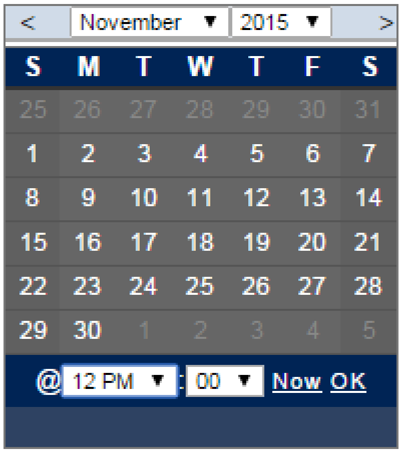
   Note: When browsing the node, you can narrow your search by applying a filter. When specifying a filter, the asterisk (*) is not a wildcard. Any string you enter as a filter is treated as a "search within". In other words, the string "foo" matches "123foo", "foo456", and "123foo456".
8. Select **Specify base for source path(s)** to place the transferred files directly into the destination folder without its hierarchy of directories. The specified base for the source path is *removed* from the source path when transferring directories.
   For example, if the source path is `/shared_files/projects/presentation`, a successful transfer results in the folder *`destination_folder`*`/shared_files/projects/presentation` on the destination node. A successful transfer with "/shared_files/projects" specified as the base path results in the folder *`destination_folder`*`/presentation` on the destination node.

   For more information on specifying a base path, see "Specify Base for Source Path" on page 150.
9. Select one of the following file-transfer rules from the **Items to transfer** drop-down list:

- Always transfer the entire directory: The transfer always transfers all files in the source directory.
- Allow initiator to select items when starting manually: The user starting this smart transfer can choose the items in the directory included in the transfer.

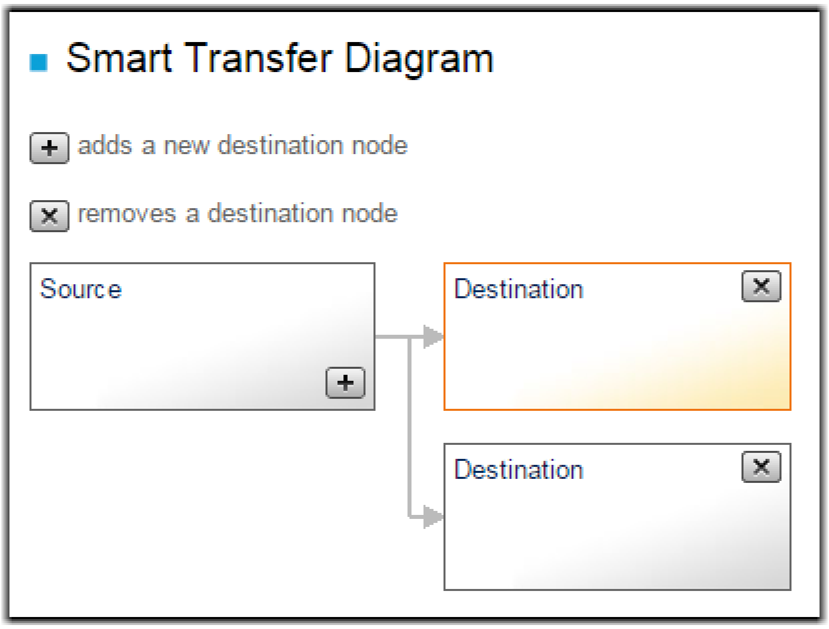
10. Expand the settings under More Options to configure addition settings.

    Click the toggle arrow next to each section to view settings.

Section	**Description**
Connection	Configure *fasp* settings.
Transfer	Configure transfer rates and policies.
Security	Encrypt the transfer.
File Handling	Configure source file attributes, archive source files after transfer, and set filters for source files.
Notifications	Configure email notification options. For more information on email notifications, see "Configuring Email Notifications" on page 14.
Advanced	Configure transfer initiator, *fasp* MTU, and read and write block sizes on source and destination nodes.
Transfer Time	Schedule your transfer to run **Now** or **Later**. If you choose **Later**, click the ▦ button and choose the date and time you want the transfer to run.  < November ▼ 2015 ▼ > S M T W T F S 25 26 27 28 29 30 31 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 1 2 3 4 5 @ 12 PM ▼ : 00 ▼ Now OK

For more information on these options, see "Smart Transfer Options" on page 147.

The highlighted box in the Smart Transfer Diagram indicates whether you are configuring the Source or Destination for the smart transfer. Make sure a Destination is selected. You can create additional destination endpoints by clicking the + button. To remove a destination, click the ✕ button inside the destination box.



11. Select the destination node or saved endpoint from the **Connect** drop-down menu.
    - Node: A node is listed as the node name (by default, its IP address) and IP address. Select the **Endpoint type** from the drop-down menu and enter your credentials or select your SSH key.
    - Cluster: A cluster is listed as the domain name. Select the **Endpoint type** from the drop-down menu and enter your credentials.
    - Endpoint: A saved endpoint is listed as *login@address* and is associated with login credentials for the username or access key. Selecting a saved endpoint does not prompt you for credentials.
    - Any: Choosing **Any** allows the user to create a new endpoint instead of choosing from the list.
12. Select your **Destination** directory.

    Click **Choose Destination Directory** to browse the node for the directories and files you want to transfer. Console displays the source path you choose once you have chosen your source directory.

    Note: When browsing the node, you can narrow your search by applying a filter. When specifying a filter, the asterisk (*) is not a wildcard. Any string you enter as a filter is treated as a "search within". In other words, the string "foo" matches "123foo", "foo456", and "123foo456".
13. Optional: Allow the user starting this smart transfer to change the directory on this destination node. The **Change Destination Path** button appears for a destination with this option enabled.
14. Optional: Allow the user starting this smart transfer to remove this destination node. The ☒ button appears for a destination with this option enabled.
15. Optional: Configure additional settings for this individual destination node.

    Note: This option is only available for a smart transfer with multiple destination nodes.

    Select **Set transfer options individually for this destination**. The **More Options** appears at the bottom of the page.

Section	Description
Connection	Configure *fasp* settings.
Transfer	Configure transfer rates and policies.
Security	Encrypt the transfer.
File Handling	Configure source file attributes, archive source files after transfer, and set filters for source files.

Section	Description
Notifications	Configure email notification options. For more information on email notifications, see “Configuring Email Notifications” on page 14.
Advanced	Configure transfer initiator, *fasp* MTU, and read and write block sizes on source and destination nodes.

For information on these options, see “Smart Transfer Options” on page 147.

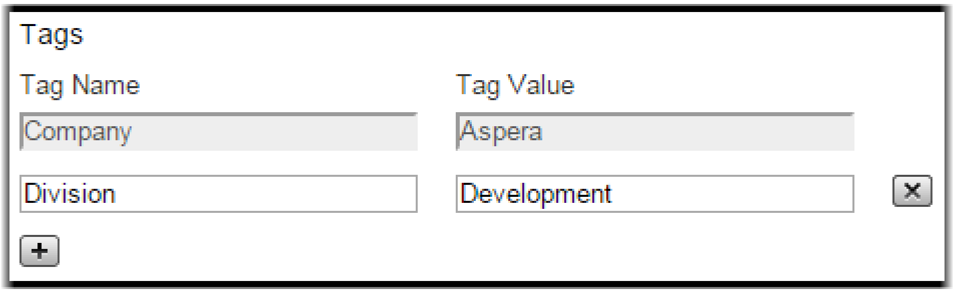
16. Click **Save**.

Once a smart transfer template has been saved, it is accessible from the Transfer page. Go to **Transfer** to **start**, **edit**, **copy**, and **delete** existing smart transfers.

Starting a Smart Transfer

Console can be used to initiate transfers between nodes when the Console user has the permission to start transfers. Console provides two types of transfer methods: simple transfers and smart transfers. Simple transfers are one-time transfer sessions that require entering all transfer information. Smart transfers are reusable templates with saved transfer settings.

1. Go to **Transfer** to see all the smart transfers you have permission to access. For instructions on creating a smart transfer, see “Creating a Smart Transfer” on page 52.
2. Find the smart transfer listed under Saved Smart Transfers and click **Start**.
3. Optional: Modify the **Transfer Name** and leave a comment describing the transfer.
4. Optional: Add new tags or modify existing tags.



Tags

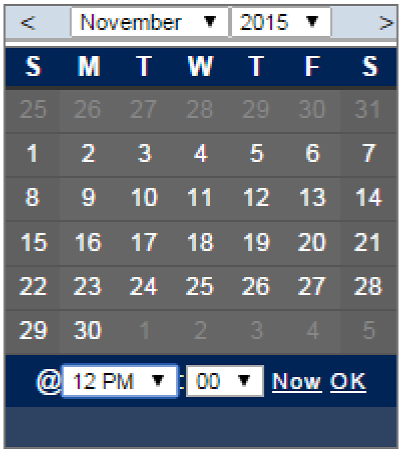
Tag Name	Tag Value
Company	Aspera
Division	Development

Click the + button to add a new tag. Enter the tag name and the tag value. Click the ☒ button to delete an existing tag. Locked tags are greyed out and cannot be modified. For more information, see “Working with Tags” on page 106.

5. Optional: Configure email notification options.

   Expand the Notifications section. Add or delete email addresses and configure notifications for existing email addresses. For more information, see “Configuring Email Notifications” on page 14.
6. Optional: Schedule the transfer to run **Now** or **Later**.

   If you choose **Later**, click the ▦ button and choose the date and time you want the transfer to run.



S	M	T	W	T	F	S
25	26	27	28	29	30	31
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	1	2	3	4	5

7. Click **Start**.

Sharing a Smart Transfer

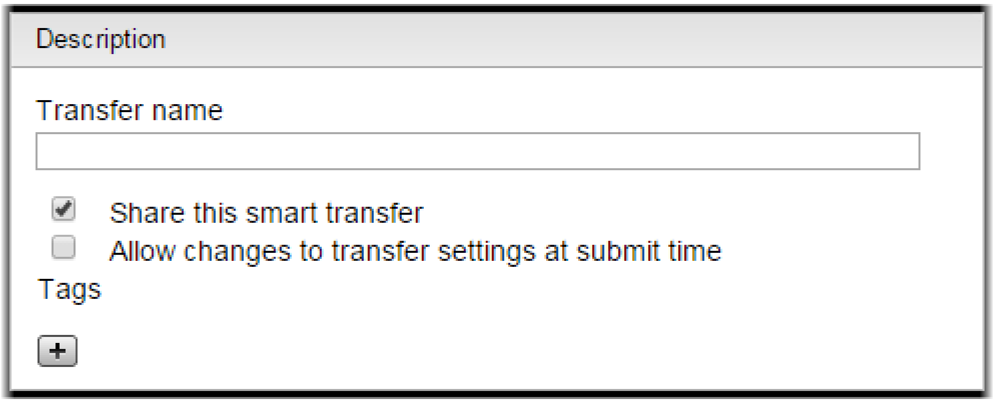
Smart transfers are reusable templates with saved settings. The primary use case for sharing smart transfers is to set up pre-defined transfers for non-admin users to run. You can decide what transfers a user can monitor and start by limiting the user's permissions and access to a smart transfer. By default, shared transfers require you to use endpoints created by an admin using the **Edit Nodes > Endpoints** page. Once configured in Console settings, you can also share smart transfers saved with personal login credentials and domain names.

For more information about sharing smart transfers with personal logins, see "Sharing a Smart Transfer with Personal Login Credentials" on page 58

1. Create endpoints on the nodes you want to use for this smart transfer.

   For detailed instructions, see "Adding Endpoints" on page 30.

   Tip: To use domain names as transfer endpoints, create an unmanaged node using a domain name, then add an Endpoint to this unmanaged node.
2. Go to **Transfer** and click **New Smart Transfer**.
3. Select **Share this smart transfer**.



Description

Transfer name

☑ Share this smart transfer

☐ Allow changes to transfer settings at submit time

Tags

   Note: When creating a smart transfer with Any as an endpoint, you must first save the smart transfer before selecting **Share this smart transfer**.
4. Select endpoints for the Source and Destination.

   Tip: Create new personal saved endpoints by selecting the desired node and entering the SSH user login credentials.
5. Finish configuring the smart transfer. Click **Save** when finished.
6. Enable a user to start this smart transfer.

- Create a group with permissions to start smart transfers for this transfer path (see "Creating Console Groups" on page 37).
- Add the user to this group (see "Creating Console Users" on page 39).

Admin users have permissions to all transfers and do not need to be added to a group to use a shared smart transfer. By default, admins do not have the ability to edit Smart Transfers that are shared with them but owned by another admin. To enable admins to edit each other's smart transfers, go to **Configuration > Defaults** and select **Smart Transfer Editing: Allow administrators to edit each other's Smart Transfers**.

Note: Even with this feature enabled, admins can only edit smart transfers that do not contain personally saved login credentials.

Tip: Editing another admin's smart transfer changes ownership of the smart transfer to the admin who made the last change.

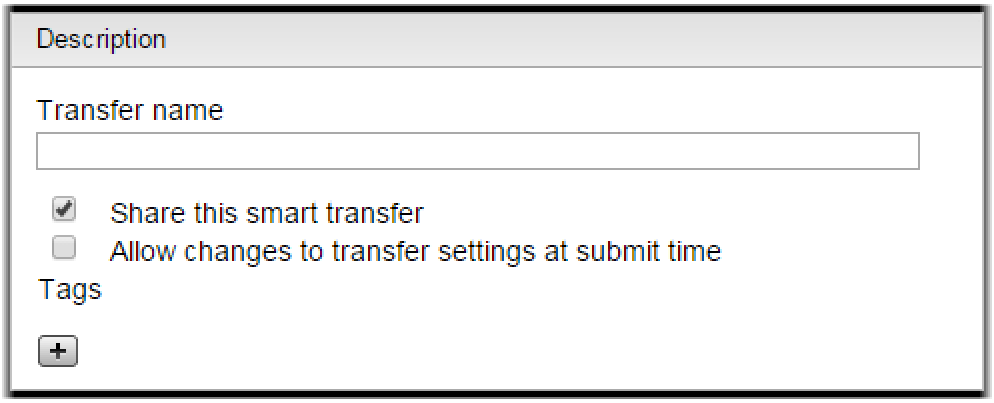
Sharing a Smart Transfer with Personal Login Credentials

Smart transfers are reusable templates with saved settings. The primary use case for sharing smart transfers is to set up pre-defined transfers for non-admin users to run. You can decide what transfers a user can monitor and start by limiting the user's permissions and access to a smart transfer. By default, shared transfers require you to use endpoints created by an admin using the **Edit Nodes > Endpoints** page. Once configured in Console settings, you can also share smart transfers saved with personal login credentials and domain names.

Personal login credentials are automatically created and saved when a user creates a transfer, chooses a node, and enters authentication credentials for an SSH user on that node. The following describes how to share a smart transfer with personal login credentials.

Note: To share a smart transfer, an administrator must enable smart transfer sharing. To enable smart transfer sharing, go to **Configuration > Defaults** and select **Allow users to share Smart Transfers that contain personal Saved Endpoints .**

1. Create a new smart transfer, select **Share this smart transfer**.

   Description

   Transfer name

   ☑ Share this smart transfer

   □ Allow changes to transfer settings at submit time

   Tags

   Note: When creating a smart transfer with Any as an endpoint, you must first save the smart transfer before selecting **Share this smart transfer**.
2. Select personal saved endpoints for the Source and Destination.

   If you have no personal saved endpoints, create a new one by selecting the desired node and entering the SSH user login credentials.
3. Finish configuring the smart transfer. Click **Save** when finished.
4. Enable a user to start this smart transfer.

   - Create a group with permissions to start smart transfers for this transfer path (see "Creating Console Groups" on page 37).
   - Add the user to this group (see "Creating Console Users" on page 39).

Admin users have permissions to all transfers and do not need to be added to a group to use a shared smart transfer. By default, admins do not have the ability to edit Smart Transfers that are shared with them but owned by another admin. To enable admins to edit each other's smart transfers, go to **Configuration > Defaults** and select **Smart Transfer Editing: Allow administrators to edit each other's Smart Transfers**.

Note: Even with this feature enabled, admins can only edit smart transfers that do not contain personally saved login credentials.

Tip: Editing another admin's smart transfer changes ownership of the smart transfer to the admin who made the last change.

Queue Transfers

Overview

The Console queueing feature provides two useful capabilities:

- Admins can limit the number of Console-initiated transfers that can run concurrently for a given destination or from a given source. This can be useful if network connections have limited bandwidth or if particular destination nodes have difficulty handling more than a small number of transfers at a time. For example, if the concurrency limit for a connection is "2", and two transfers are in progress, any new transfers initiated while the first two are still in progress will be queued in the order in which they were initiated.
- All users can change the priority order of queued and in-progress transfers. This can be useful in situations where users need to respond to emergencies or shifting priorities.

Important: Queueing only applies to transfers started from Console or via its API. Transfers started outside Console are not subject to queueing and do not count towards concurrency limits.

Concurrency limits are always assigned on a per-node basis, and per outbound or inbound direction. However, the overall, actual limit on a set of concurrent transfers between two nodes is governed by the node with the lowest limit. That is, if NodeA has an outbound limit of "2" and NodeB has an inbound limit of "1", concurrent transfers from NodeA to NodeB are limited to one transfer at a time, with subsequent transfers queued up in the order in which they were initiated.

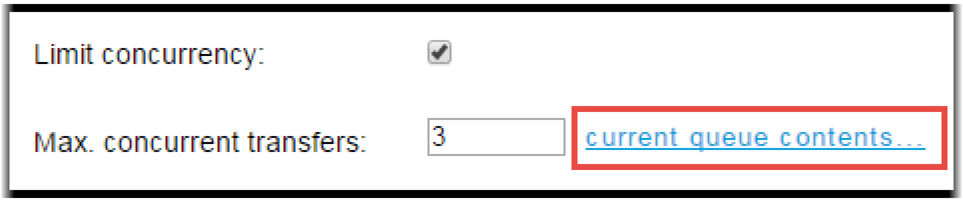
Adjusting the Queueing Properties of In-Progress Transfers

If queuing is enabled on a node (see "Configuring Queues for Nodes" on page 60), the queueing properties of in-progress transfers can be adjusted in several ways:

- Their relative priorities can be raised or lowered.
- They can be paused and resumed.
- The concurrency limit in effect can be raised or lowered.
- Concurrency (and therefore queueing) can be disabled completely.

These adjustments can be made while monitoring the node from the Node Detail page. You can view the Node Detail page by going to **Nodes** and clicking on the node.

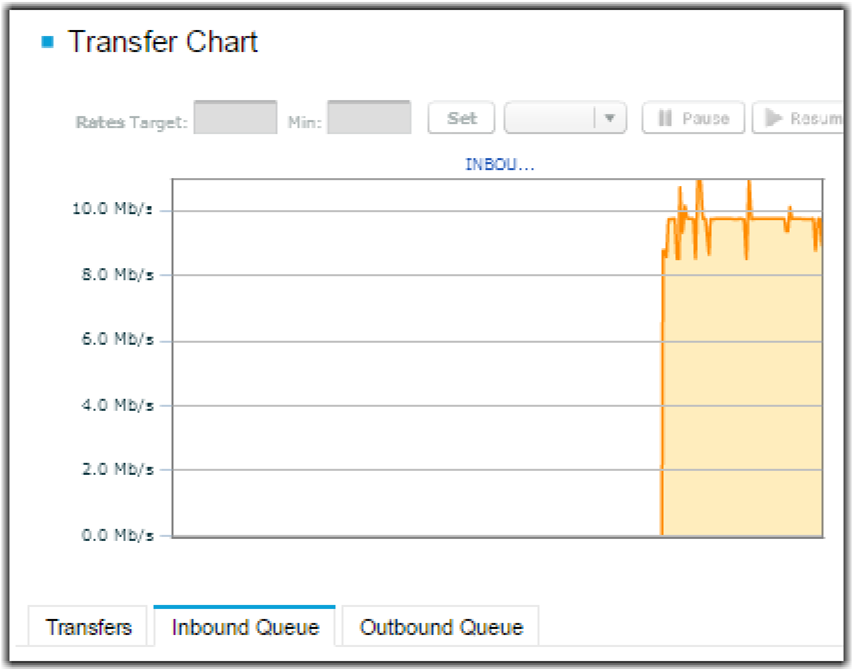
Tip: You can also reach this page from the Queing page on the **current queue contents** link next to an enabled concurrency limit.



Limit concurrency: ☑

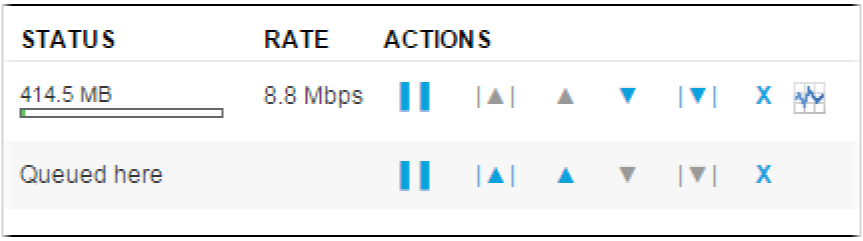
Max. concurrent transfers: 3

On the Node Detail page, below the transfer chart, you may see the **Inbound Queue** tab, the **Outbound Queue** tab, or both. These tabs are visible if the node is configured with inbound or outbound queueing. Clicking the tab displays the node's inbound or outbound transfers - both those currently in progress and those in the queue.



To view past transfers, open the **Transfers** tab. The **Transfers** tab also shows both outbound and inbound transfers, but does not include controls to promote, demote, pause, or resume transfers.

Controlling In-Progress Transfers



STATUS	RATE	ACTIONS
414.5 MB	8.8 Mbps	❚❚ |▲| ▲ ▼ |▼| X 📈
Queued here		❚❚ |▲| ▲ ▼ |▼| X

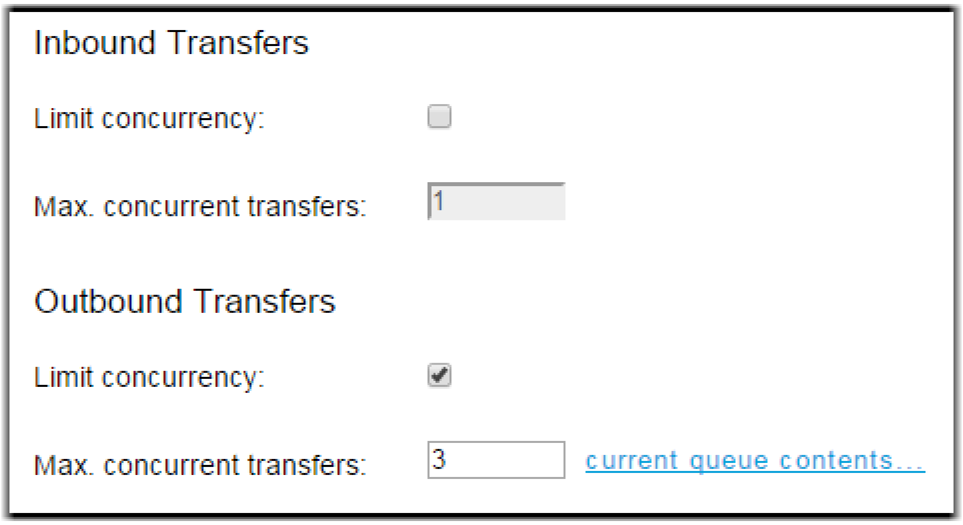
Icon	Action
❚❚	Pause Transfer
▶	Resume Transfer
|▲|	Promote Transfer to Highest Priority
▲	Promote Transfer
▼	Demote Transfer
|▼|	Demote Transfer to Lowest Priority
X	Cancel Transfer
📈	Highlight this session on the graph

Configuring Queues for Nodes

Both managed and unmanaged nodes can be configured for queuing. For more information on queuing, see " Queue Transfers" on page 59.

1. Go to **Nodes**. Find your node in the Managed Nodes or Unmanaged Nodes page . Click **edit** and then click **Queueing**.
2. Enable queueing by selecting **Limit concurrency** (disabled by default) for Inbound Transfers.
3. Choose the maximum number of concurrent transfers allowed for this node. The default setting is "1".
4. Repeat the previous two steps for Outbound Transfers.
5. Click **Update**.

In the example below, queueing is disabled for inbound transfers. For outbound transfers, queueing is enabled for at most three transfers in progress at the same time.



Inbound Transfers

Limit concurrency: □

Max. concurrent transfers: 1

Outbound Transfers

Limit concurrency: ☑

Max. concurrent transfers: 3 current queue contents...

Configure Failover Groups

Overview

A failover group contains a group of different nodes that act as substitutes for the original node in the case that the original node becomes unavailable. When a node goes offline, Console also restarts any transfers in progress on that node, submitting them to a different node in the group.

Note: Transfer failover only activates if the status of a node is set to error. Transfers that are inactive do not failover.

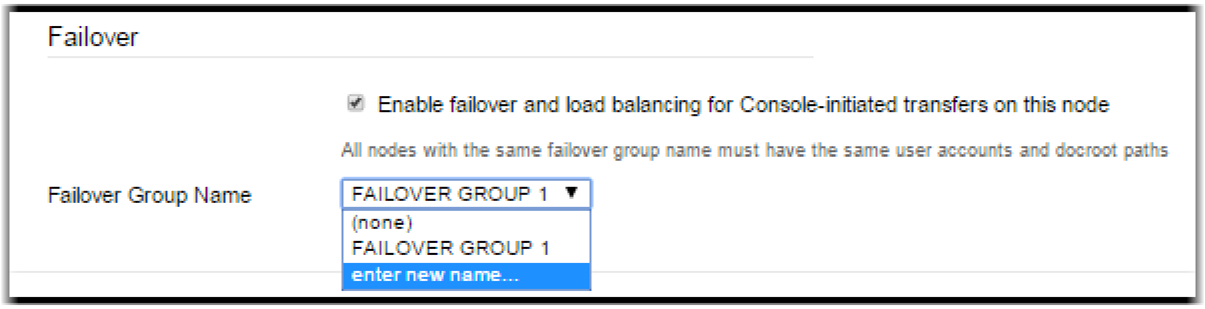
Node Requirements

Nodes must have identical passwords, transfer accounts, and docroots to be grouped together. Make sure each node has identical configurations for each item in the following list before adding them to a failover group:

- System User Accounts
- Transfer User Accounts
- Node API User Accounts
- Docroots
- Endpoints on Console
- Directory Structure

Adding a Node to a Failover Group

When adding or editing a node, select **Enable failover and load balancing for Console-initiated transfers on this node**. Add the node to an existing group or select **enter new name** from the **Failover Group Name** drop-down menu to create a new group.


Failover

☑ Enable failover and load balancing for Console-initiated transfers on this node

All nodes with the same failover group name must have the same user accounts and docroot paths

Failover Group Name: FAILOVER GROUP 1 ▼

If you select **enter new name**,enter a new failover group name in the prompt.

Endpoint Synchronization

Editing an endpoint on a node of a failover group makes those changes to the same endpoint on the other nodes in the failover group.

Note: Only saved and synchronized endpoints should be selected as a destination when starting a transfer.

Configuring Load Balancing

Go to **Configuration > Console Defaults** and configure the Failover / Load balancing Behavior option.

- **Failover + Load Balancing**: The transfer uses the least busy nodes first.
- **Failover only**: The transfer will uses the original endpoints that the user specified.

Creating a Cookie Parsing Rule

Note: Cookie configuration applies only to the use of custom cookies. Console does not apply parsing rules to cookies it recognizes as standard cookies used by Aspera products.

In an **ascp** command-line transfer, you can specify the transfer cookie with an environment variable.

```
set ASPERA_SCP_COOKIE=custom_cookie
```

Using a rule, Console can match the set cookie string and then substitute it for selected transfer information.

1. Go to **Configuration > Cookies**. Click **New Rule**.
2. Name the rule.
3. Configure the cookie.

   Enter the regular expression Console uses to filter transfers. If this string matches a transfer, Console includes the cookie in the transfer and the information in the other fields is used in the transfer session.

   Tip: The regular expression follows Ruby regular expression format currently documented on the Ruby web site.
4. Configure the cookie with the following information:

Field	Description
Started via	Name of the transfer initiator.
Contact description	Description of this transfer initiator.
Transfer name	Name for this transfer.

5. Click **Create**.

When you have multiple cookie parsing rules, Console uses the first rule listed that matches the cookie string. To modify the order of the parsing rules, drag-and-drop the rules in the list. If two rules have identical regular expressions, the rule that is higher in the list is applied.

It is possible to capture parts of the cookie and reuse the value in the three parameters. For instance to enable setting the three transfer fields directly from the initiating application, one can fill in the fields with the following configurations:

Field	Description
Rule name	*MyCustomCookieRule*
Regexp	*^setcustomfields:(.+?):(.+?):(.+?):$*
Started via	\1
Contact description	\2
Transfer name	\3

For example, the following cookie replaces **Started via** with "My App", **Contact description** with "My Contact", and **Transfer name** with "My Transfer".

```
set ASPERA_SCP_COOKIE="setcustomfields:My App:My Contact:My Transfer:"
```

Working with Watchfolders

Watch Folders automate file transfers from a folder on a source node to a destination node. Files placed into a source folder are automatically transferred to the destination.

Creating a Push Watch Folder

A local-to-remote (push) Watch Folder monitors a directory on the source node where the Watch Folder runs. If it detects new files in the source directory, it transfers those files to a directory on the destination node.

To use Watch Folders, you must enable the feature (**Configuration > Defaults** and select **Enable Watchfolder management**) and add Watch Folder-enabled nodes to Console as managed nodes. For instructions on enabling Watch Folders on a node, see *IBM Aspera High-Speed Transfer Server Admin Guide: Watch Folders and the Aspera Watch Service*.

1. Go to **Activity > Watchfolders** and click **Create New Watchfolder**.
2. Enter a name for your watch folder in the Name field.
3. Select **Push** for the Watch Folder type.
4. Configure the source node and directory. Files dropped into the source directory are transferred to the destination node and directory.
   a) Select your source node from the drop-down menu. Only Watch Folder-configured nodes appear in the list.
   b) Select the source directory.

      You can **Browse** and select the source directory, or manually enter the source path.

      Important: The path is relative to the associated system user's docroot. For example, if you enter `project1/final` and the user's docroot is `/aspera/projects`, the source path is `/aspera/projects/project1/final`.
5. Enter authentication credentials for the destination node in the Target pane.
   a) Enter the IP address or hostname of the target node.

b) Choose the type of credentials you want to use for authentication. Enter the credentials to authenticate to the destination node and authorize transfers.

Authentication Type	UI Option	Credentials
SSH with password	**SSH Password**	SSH user and SSH password
SSH with key	**SSH Key**	SSH user and SSH private key **Note:** You must give the full path to the SSH key, not the path relative to the docroot.
HTTPS with Node API	**Node API**	Node API user and Node API password
HTTPS with Shares API	**Shares**	Aspera Shares API user and Aspera Shares API password

Important:

1) To authenticate using SSH with password, the source node must be running IBM Aspera High-Speed Transfer Server 3.7 or later.

2) To authenticate using HTTPS with Node API or HTTPS with Shares API, both the source and destination nodes must be running IBM Aspera High-Speed Transfer Server 3.8 or later.

6. If you want, you can also enter a token string to authenticate the transfers, with the syntax "Basic *token_string*".

   For instructions on creating an access key on the remote node and generating a token from it, see the *IBM Aspera High-Speed Transfer Server Admin Guide: Access Key Authentication*.

7. If you want to specify an Aspera forward proxy for the Watch Folder, enter the proxy address in the following format:

```
dnat://login:password@address:port
```

   For more information on forward proxies, see the IBM Aspera Proxy Admin Guide.

8. Set ports for SSH/TCP or Node API (depending on your authentication type) connections, and for *fasp*™ (UDP) connections.

Target Authentication	Default Ports
SSH Password	SSH Port: 22 UDP: 33001
SSH Key	SSH Port: 22 UDP: 33001
Node API	Node API Port: 9092 UDP: 33001
Shares	Shares API Port: 443 UDP: 33001

9. Select the destination directory.

   You can **Browse** and select the destination directory, or manually enter the destination path.

   Important: Just like for the source path, the destination path is relative to the associated system user's docroot.

If you are using an SSH key for authentication and you want to browse for the destination, you need to first add the key to Console to authenticate Console to browse the node. You do not need to add the key to Console if you know the destination path and enter it manually. For more information on adding SSH keys to Console, see "Storing SSH Keys on Console" on page 83.

10. Configure additional watchfolder settings. See "Watchfolder Options" on page 71.
11. Click **Create**.
12. Test your watch folder.

    Add files to the source directory. Watch Folder detects the new files and transfers the new files to the target directory. You can see the transfer on the **Activity Overview** page.

Creating a Pull Watch Folder

In a remote-to-local (pull) Watch Folder scenario, the destination node (managed node receiving files) runs the Watch Folder service. The destination node remotely monitors the source directory on the source node using the Node API. If the destination node detects new files in the source directory, it transfers those files to the configured, local directory on the destination node.

Important: Pull-type Watch Folders require both source and destination nodes run HSTS version 3.8 and later.

To use Watch Folders, you must enable the feature (**Configuration > Defaults** and select **Enable Watchfolder management**) and add Watch Folder-enabled nodes to Console as managed nodes. For instructions on enabling Watch Folders on a node, see *IBM Aspera High-Speed Transfer Server Admin Guide: Watch Folders and the Aspera Watch Service*.

1. Go to **Activity > Watchfolders** and click **Create New Watchfolder**.
2. Enter a name for your watch folder in the Name field.
3. Select **Pull** for the Watch Folder type.
4. Configure the destination node and directory. Files dropped into the source directory are transferred to this node and this directory.
   a) Select your target node from the drop-down menu. Only Watch Folder-configured nodes appear in the list.
   b) Select the destination directory.

      You can **Browse** and select the destination directory, or enter the source path.

      Important: The destination directory is relative to the associated system user's docroot. For example, if you enter `project1/final` and the user's docroot is `/aspera/projects`, the source path is `/aspera/projects/project1/final`.
5. Enter authentication credentials for the source node in the Source pane.
   a) Enter the IP address or hostname of the source node.
   b) Choose the type of credentials you want to use for authentication. Enter the credentials to authenticate to the destination node and authorize transfers.

      Pull Watch Folders must use the Node API or the Shares API to authenticate to the source node. The source node must be running HSTS 3.8.0 and later.

Authentication Type	UI Option	Credentials
HTTPS with Node API	**Node API**	Node API user and Node API password
HTTPS with Shares API	**Shares**	Aspera Shares API user and Aspera Shares API password

6. If you want, you can also enter a token string to authenticate the transfers, with the syntax "Basic *token_string*".

For instructions on creating an access key on the remote node and generating a token from it, see the *IBM Aspera High-Speed Transfer Server Admin Guide: Access Key Authentication*.

7. If you want to specify an Aspera forward proxy for the Watch Folder, enter the proxy address in the following format:

   ```
   dnat://login:password@address:port
   ```

   For more information on forward proxies, see the IBM Aspera Proxy Admin Guide.
8. Set ports for Node API connections, and for *fasp*™ (UDP) connections.

Target Authentication	Default Ports
Node API	Node API Port: 9092 UDP: 33001
Shares API	Shares API Port: 443 UDP: 33001

9. Select the source directory.

   You can **Browse** and select the source directory, or manually enter the source path.

   Important: Just like for the destination path, the source path is relative to the associated system user's docroot.
10. Configure additional watchfolder settings. See “Watchfolder Options” on page 71.
11. Click **Create**.
12. Test your watch folder.

    Add files to the source directory. Watch Folder detects the new files and transfers the new files to the target directory. You can see the transfer on the **Activity Overview** page.

Creating a Push Watch Folder with ATS Node Destination

Create a local-to-remote (push) Watch Folder with an Aspera Transfer Service (ATS) node as the destination node. If Watch Folder detects new files in the source directory, it transfers those files to a directory on ATS node.

Important: Using Watch Folders with an ATS node requires that both source and destination nodes run IBM Aspera High-Speed Transfer Server version 3.8 and later.

To use Watch Folders, you must enable the feature (**Configuration > Defaults** and select **Enable Watchfolder management**) and add Watch Folder-enabled nodes to Console as managed nodes. For instructions on enabling Watch Folders on a node, see *IBM Aspera High-Speed Transfer Server Admin Guide: Watch Folders and the Aspera Watch Service*.

1. Go to **Activity > Watchfolders** and click **Create New Watchfolder**.
2. Enter a name for your watch folder in the Name field.
3. Select **Push** for the Watch Folder type.
4. Configure the source node and directory. Files dropped into the source directory are transferred to the destination node and directory.
   a) Select your source node from the drop-down menu. Only Watch Folder-configured nodes appear in the list.
   b) Select the source directory.

   You can **Browse** and select the source directory, or manually enter the source path.

   Important: The path is relative to the associated system user's docroot. For example, if you enter `project1/final` and the user's docroot is `/aspera/projects`, the source path is `/aspera/projects/project1/final`.

5. Enter Node API authentication credentials for the ATS node in the Target pane.
   a) Enter the IP address or hostname of the target node.
   b) Enter the Node API username in the **Login** field.
   c) Choose **Node API** for the target authentication credential type.
   d) Enter the Node API username in the **Login** field.
6. If you want, you can also enter a token string to authenticate the transfers, with the syntax "Basic *token_string*".

   For instructions on creating an access key on the remote node and generating a token from it, see the *IBM Aspera High-Speed Transfer Server Admin Guide: Access Key Authentication*.
7. If you want to specify an Aspera forward proxy for the Watch Folder, enter the proxy address in the following format:

   ```
   dnat://login:password@address:port
   ```

   For more information on forward proxies, see the IBM Aspera Proxy Admin Guide.
8. Set ports for the Node API:
   - **Node API**: 443
   - **UDP**: 33001
9. Select the destination directory.

   You can **Browse** and select the destination directory, or manually enter the destination path.

   Important: Just like for the source path, the destination path is relative to the associated system user's docroot.
10. Configure additional watchfolder settings. See “Watchfolder Options” on page 71.
11. Click **Create**.
12. Test your watch folder.

    Add files to the source directory. Watch Folder detects the new files and transfers the new files to the target directory. You can see the transfer on the **Activity Overview** page.

Creating a Pull Watch Folder with ATS Node Destination

In a remote-to-local (pull) Watch Folder scenario, the destination node (managed node receiving files) runs the Watch Folder service. The destination node remotely monitors the source directory on the source node (Aspera Transfer Service (ATS) node). If the destination node detects new files in the source directory, it transfers those files to the configured, local directory on the destination node.

Important: Using Watch Folders with an ATS node requires that both source and destination nodes run IBM Aspera High-Speed Transfer Server version 3.8 and later.

To use Watch Folders, you must enable the feature (**Configuration > Defaults** and select **Enable Watchfolder management**) and add Watch Folder-enabled nodes to Console as managed nodes. For instructions on enabling Watch Folders on a node, see *IBM Aspera High-Speed Transfer Server Admin Guide: Watch Folders and the Aspera Watch Service*.

1. Go to **Activity > Watchfolders** and click **Create New Watchfolder**.
2. Enter a name for your watch folder in the Name field.
3. Select **Pull** for the Watch Folder type.
4. Configure the destination node and directory. Files dropped into the source directory are transferred to this node and this directory.
   a) Select your target node from the drop-down menu. Only Watch Folder-configured nodes appear in the list.
   b) Select the destination directory.

      You can **Browse** and select the destination directory, or enter the source path.

Important: The destination directory is relative to the associated system user's docroot. For example, if you enter `project1/final` and the user's docroot is `/aspera/projects`, the source path is `/aspera/projects/project1/final`.

5. Enter authentication credentials for the source node in the Source pane.

   Pull Watch Folders must use the Node API to authenticate to the source node. The source node must be running HSTS 3.8.0 and later.

   a) Enter the IP address or hostname of the ATS node.

   b) Enter the username of a Node API user on the ATS node.

   c) Enter the Node API password of the Node API user.

6. If you want, you can also enter a token string to authenticate the transfers, with the syntax "Basic *token_string*".

   For instructions on creating an access key on the remote node and generating a token from it, see the *IBM Aspera High-Speed Transfer Server Admin Guide: Access Key Authentication*.

7. If you want to specify an Aspera forward proxy for the Watch Folder, enter the proxy address in the following format:

   ```
   dnat://login:password@address:port
   ```

   For more information on forward proxies, see the IBM Aspera Proxy Admin Guide.

8. Set ports for the Node API:

   - **Node API**: 443
   - **UDP**: 33001

9. Select the source directory.

   You can **Browse** and select the source directory, or manually enter the source path.

   Important: Just like for the destination path, the source path is relative to the associated system user's docroot.

10. Configure additional watchfolder settings. See "Watchfolder Options" on page 71.
11. Click **Create**.
12. Test your watch folder.

    Add files to the source directory. Watch Folder detects the new files and transfers the new files to the target directory. You can see the transfer on the **Activity Overview** page.

Monitoring and Managing Watch Folders

Monitor Watch Folder transfers from the **Activity** page.

Go to **Activity > Watchfolders** to see a list of Watch Folders and their status. You can **edit** and **delete** Watch Folders from this page.

If a Watch Folder is configured for nodes running IBM Aspera High-Speed Transfer Server 3.9.3 and later, you can pause, stop, and resume the Watch Folder.

Working with Growing Files

Growing files are "in-progress" files that continue to grow in size, such as videos generated at live events. Watch Folders can detect and transfer these growing to a target directory.

Requirements

- Both the source host and remote host must be running IBM Aspera High-Speed Transfer Server 3.6.0 and later to support the `faspstream` technology used to transfer growing files.
- The transfer user on the node running the Watch Folder service transfer must not have a docroot.

- Pull Watch Folders do not support growing files.

Configuration

When creating or editing a Watch Folder in Console, you can configure how Console handles growing files in the Growing Files section under More Options:

Option	Description	Default	Example
Maximum parallel transfers	The maximum number of concurrent transfers of growing files watchfolder can initiate.	8	4
Target rate	The target transfer rate.	`10000`	`10000`
Bandwidth policy	The bandwidth policy used for prioritizing growing file transfers.	**fair**	**high**
Transport encryption	Transfer with an encryption method.	**none**	**aes128**
TCP port	TCP port to use for this Watch Folder. If different, Console connects to the target node using this port instead.	22	22
fasp™ port (UDP)	UDP port to use for this Watch Folder. If different, Console connects to the target node using this port instead.	`33001`	`33001`
Completion timeout	Duration in seconds of inactivity after which Console considers the growing file transfer to be complete.	`10`	`15`
Memory	The maximum amount of memory that the faspstream binary is allowed to use.	`128`	`128`
Chunk size	The size of data to pack before sending over the network.	`128`	`128`
Growing file filters	Defines which files are considered growing files. Click the + button to add a new filter to identify the growing file. You can set a filter to include or exclude files by globbing or by regular expression.		`*.m2p`

Note: Sessions with growing files are not affected by settings in the Transfer section.

Controlling Transfer Order with File Lists

Control the transfer order for files added to a Watch Folder by configuring the Watch Folder to transfer files in packages, which are defined by file lists. The file list determines what files goes into a packaged and the order in which they are transferred.

The last file in a file list is transferred last; it will not be transferred until all other files in the file list have been transferred.

Use file lists with workflows that trigger on the presence of a particular file, but require the presence of other files to function correctly. For example, a workflow that builds a project dependent on the existence of other files fails if the trigger file is transferred before those other files have been transferred. Using a file list ensures the trigger file is not transferred until the rest of the files have been transferred.

Creating a File List

The file list is a simple text file with `#aspera-filelist` in the first line and a list of files. Console only transfers the last file in the list after the other files have finished transferring. Here is an example file list:

```
#aspera-filelist
superhero_epic_the_musical-english.mp4
superhero_epic_the_musical-spanish.mp4
superhero_epic_the_musical-chinese.mp4
```

```
superhero_epic_the_musical.png
superhero_epic_the_musical-english-trailer.mp4
ADI.XML
```

Note: The paths listed in a file list are relative to the Watch Folder directory. Paths to other directories are not supported.

Configuration

When creating or editing a Watch Folder in Console, you can configure file list handling in the Packages/Drops section under More Options:

Option	Description	Default	Example
Package timeout	The duration in seconds for which the Watch Folder waits for files defined in the file list. If the required files do not appear within the duration, files with dependencies are not transferred because of unsatisfied dependencies.	`10`	`30`
Final transfer	Defines which file is transferred last. • **Last file in list**: The last file in the package list is transferred last. • **File list**: The file list is transferred last, but the other files are transferred without any specific order.	**Last file in list**	**Last file in list**
File list filters	Click the + button to add a new filter to identify file lists. You can set a filter to include or exclude files by globbing or by regular expression.		**Glob** **Include**`*.package`

Example

One of our workflows monitors a Watch Folder destination directory for new files. When it detects an XML file, it starts a process to move all the files listed in that XML file to another directory and packages those files in a media project according to metadata that the XML also provides. If the XML file is transferred before the other files (which is likely, given its small size), our workflow runs and fails due to missing files. To solve this issue, we include a file list with our media files and the XML file:

```
#aspera-filelist
superhero_epic_the_musical-english.mp4
superhero_epic_the_musical-spanish.mp4
superhero_epic_the_musical-chinese.mp4
superhero_epic_the_musical.png
superhero_epic_the_musical-english-trailer.mp4
ADI.XML
```

We name our file list `superhero_epic.package`.

Next, we configure our Watch Folder to detect the file list by adding a glob filter that looks for files that include the `*.package` pattern. We set our timeout to `30` seconds for buffer, and keep the default **Last file in list**, to make sure `ADI.XML` transfers last.

When we move our files into the monitored source directory on the source node, Watch Folder detects the `superhero_epic.package` file list and waits 30 seconds before checking that all the files are present in the source directory. Finding all the files present, Watch Folder transfers all the `.mp4` files and the `.png` file to the target directory on the destination node first, and then transfers `ADI.XML`.

Our workflow detects the the `ADI.XML` file and successfully performs its task.

Excluding Files from Watch Folder Monitoring

Use filters to determine what Watch Folders detects and transfers.

When creating or editing a Watch Folder in Console, you can configure file filters in the Watchfolder Settings section under More Options.

Click the + button to add a new file filter. You can set a filter to include or exclude files using globbing or regular expression matching.

For example, to exclude text files (`.txt`) and backup files (`.bak`), create two filters, one that uses glob matching to exclude file matching the `*.txt` pattern, and on that uses glob matching to exclude matching the `*.bak` pattern.

Watchfolder Options

The following tables provide information on additional configurable settings that are available when creating watchfolders.

Watchfolder Settings

Note: A watchfolder groups new or updated files it detects in its source folder into "drops". A drop is defined by the duration set by the snapshot creation period. All files in a given drop are transferred in the same transfer session, post-processed together, and reported as a unit.

Option	Description
Drop detection strategy	The strategy this watchfolder uses to detect files dropped into the source folder. • Cool off only: Create a drop that includes new files added within the duration by the snapshot creation period. • Top level files: Create a drop for each file placed in the top level of the source folder. • Top level directories: Create a drop for each directory placed in the top level of the source folder. This drop also includes the sub-directories and files in the top level directory.
Drop detection cool off	The duration allowed for new files to be included in a drop. Aspera recommends choosing a multiple of the specified `snapshot_creation_period` for predictable results.
Snapshot creation period	The duration used to determine what files are included in the current drop.
Connect timeout	The duration the source node waits to connect to the destination node.
Sample period	The frequency of the system estimateing the available bandwidth.

Option	Description
Queue threshold	The duration watchfolder adds files to a session. Use this feature to limit the number of files transferred based on the computed available bandwidth.
Retry duration	The duration in which the source node tries to establish a connection with the destination node.
Wait between retries	The duration the source node waits in between retries.
File detection cool off	The duration watchfolder in which placing a new file in the source folder does not trigger a new drop. **Note:** This setting does not apply to the **Cool off only** detection strategy.
File filters	Click the + button to add a new filter to identify file lists. You can set a filter to include or exclude files by globbing or by regular expression.

Transfer

Option	Description
Target rate	The transfer target rate.
Minimum rate	The transfer minimum rate
Bandwidth policy	Choose a transfer policy among fixed | high | fair | low.
Transport Encryption	Select aes-128 to transfer with this encryption method.
Retry policy	The number of attempts and the duration between each retry.

Security

Option	Description
Content Protection	Select **Encrypt transferred files with a password** to enable content encryption. Enter and confirm the password the recipient must use to decrypt the transferred files. **Note:** When editing a watchfolder with content protection enabled, you must re-The content protection password. A password must be provided in order to save changes to the watchfolder.

File Handling

Option	Description
Resume policy	Specify a resume policy and the overwrite rule when the file exists on the destination.
File attributes	Preserve file UIDs, GIDs, or timestamps.
Source Archiving	The designated directory source files are moved to after completing a transfer. The transfer's session details page display the archive directory's filepath as the **After transfer** path.

Option	Description
	Note: The **After transfer** path will only be visible in the session details of the Console that initiated the transfer. Another Console monitoring the same managed nodes will not have access to the **After transfer** path. **Note:** Re-running the transfer may generate a "No such file or directory" error since the source files were moved to the archive directory. You can use archive directory variables in the filepath to define specific archive paths for each drop. Hover over the **Archive directory variables** link for a list of available variables.
Source deletion	Delete the transferred files from the source computer after transfer.

Growing Files

Option	Description	Default	Example
Maximum parallel transfers	The maximum number of concurrent transfers of growing files watchfolder can initiate.	8	4
Target rate	The target transfer rate.	`10000`	`10000`
Bandwidth policy	The bandwidth policy used for prioritizing growing file transfers.	**fair**	**high**
Transport encryption	Transfer with an encryption method.	**none**	**aes128**
TCP port	TCP port to use for this Watch Folder. If different, Console connects to the target node using this port instead.	22	22
fasp™ port (UDP)	UDP port to use for this Watch Folder. If different, Console connects to the target node using this port instead.	`33001`	`33001`
Completion timeout	Duration in seconds of inactivity after which Console considers the growing file transfer to be complete.	`10`	`15`
Memory	The maximum amount of memory that the faspstream binary is allowed to use.	`128`	`128`
Chunk size	The size of data to pack before sending over the network.	`128`	`128`
Growing file filters	Defines which files are considered growing files. Click the + button to add a new filter to identify the growing file. You can set a filter to include or exclude files by globbing or by regular expression.		`*.m2p`

Packages / Drops

Option	Description	Default	Example
Package timeout	The duration in seconds for which the Watch Folder waits for files defined in the file list. If the required files do not appear within the duration, files with dependencies are not transferred because of unsatisfied dependencies.	`10`	`30`
Final transfer	Defines which file is transferred last. • **Last file in list**: The last file in the package list is transferred last.	**Last file in list**	**Last file in list**

Option	Description	Default	Example
	• **File list**: The file list is transferred last, but the other files are transferred without any specific order.		
File list filters	Click the ➕ button to add a new filter to identify file lists. You can set a filter to include or exclude files by globbing or by regular expression.		**Glob Include**`*.package`

Configuring Proxies

Use a proxy to connect to nodes for transfer initiation and node browsing.

Go to **Configuration > Proxy** to configure global HTTP and FASP proxies.

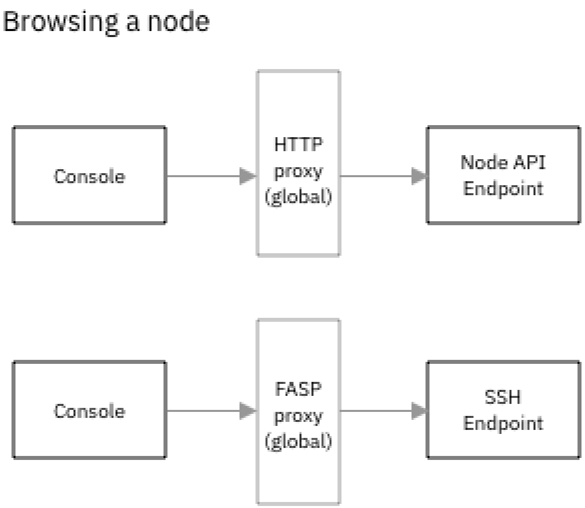
HTTP Proxy Vs. Fasp Proxy

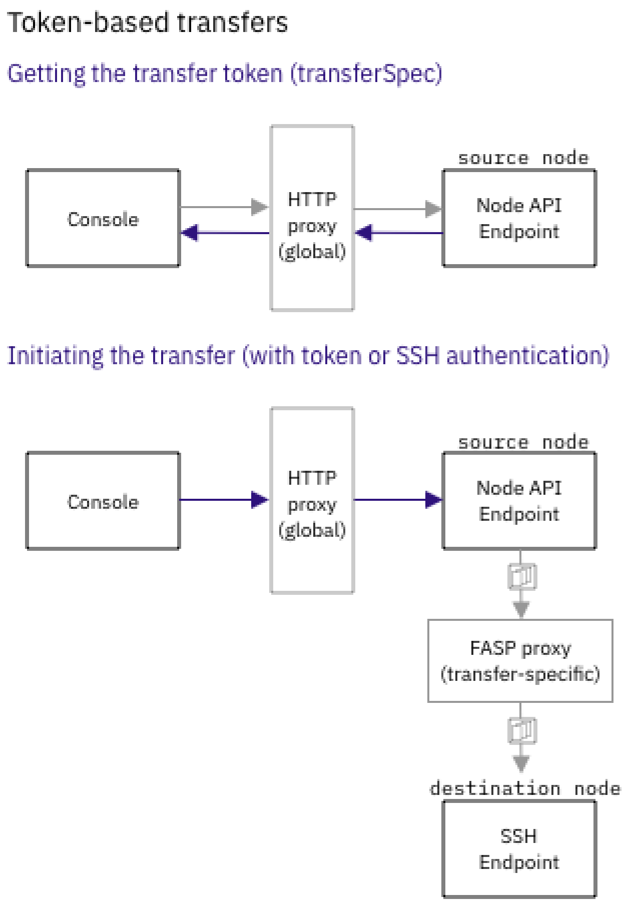
Console supports using two types of proxies: HTTP and FASP proxies.

The FASP proxy is only used when browsing a node through an SSH endpoint. Console uses the HTTP proxy for all other scenarios:

- Browsing a source or destination node using a Node API endpoint
- Retrieving a transfer token from the destination node when starting a token-based transfer
- Starting a transfer at the source node

The following diagrams show different scenarios in which Console goes through a proxy to communicate with a node:


Browsing a node



Note: When the FASP proxy between the source node and destination node is different from the global FASP proxy used by Console. The proxy between the nodes is defined during creation of a simple or smart transfer (**More Options > Connection**).

Configuring Proxies

Go to **Configuration > Proxy** to configure global HTTP and FASP proxies. You can enable one or both proxies. Both HTTP and FASP proxies take similar parameters:

Field	Description
Address	Proxy IP address or domain name
Port	Proxy port number
SSL	(FASP proxy only) Enable or disable use of SSL
Username	Proxy username
Password	Proxy user password

Creating Proxy Exclusion Rules

You can exclude a range of IP addresses and domain names from going through proxies by creating proxy exclusion rules. Define a rule by selecting the rule type and providing a range of addresses.

For example, you can exclude all IP Addresses starting with 192 and all domain names ending in `ibm.com` with the following two rules:

Order	Rule Type	Address
1	Exclude	`192.*`
2	Exclude	`*.ibm.com`

Console applies the first matched rule. The order of your rules is important, especially if you are including some addresses and excluding others. For example, to include `10.0.71.51`, but exclude all other `10.X` addresses:

Order	Rule Type	Address
1	Include	`10.0.71.51`
2	Exclude	`10.*`

Console uses configured proxies to connect to a node at `10.0.71.51`, even though all `10.X` addresses are excluded, because the address matched the inclusion rule first. In the reverse case, however, Console does not use configured proxies for `10.0.71.51`, even though the address is included, the address matched the exclusion rule first:

Order	Rule Type	Address
1	Exclude	`10.*`
2	Include	`10.0.71.51`

If an address doesn't match any of the rules, Console applies the global proxy as usual.

Running Reports

Creating a Basic Report

Console allows you to create and export custom reports, as well as apply filters and scheduling options. The steps below demonstrate how to configure new, **basic** reports. To view an example of a basic report, see the three samples in this topic. To learn about creating **advanced** reports within Console, see "Creating an Advanced Report" on page 77. To create an **advanced report**, click the **New Advanced** button instead. You can also copy and edit Console's built-in, advanced reports, which are listed on the *Manage Report Types* page. For further information on **advanced** reporting, see "Creating an Advanced Report" on page 77.

1. Go to **Reports > Manage Report Types**. Click the **New Basic** button.
2. Enter a name for your report (limited to 75 characters) and a detailed description about the report.
3. Choose the level of detail to show on your report.

   Select a field from the drop-down list to be used as the basis for organizing your report. Console generates a report with a row for each item that matches a chosen field. If you choose more than one field, Console generates a multi-level report. The data in the generated report is grouped in ascending order by the fields selected from the drop-down list. For example, if you select **Client address**, the data in the report is grouped by the transfer initiator IP addresses. For example, Console groups the five transfers initiated by IP Address 1 in the first grouping,the three transfers initiated by IP Address 2 in the second grouping, and so on..

   Note: Once a field is selected, the drop-down list updates automatically to allow for multiple levels of organization. To remove a level of organization, click the **Remove** link that appears next to the selected field.

The drop-down list includes all Console built-in fields and custom fields. For a list of built in fields, see "Reference: Basic Report Organization Options" on page 151. For more information on custom fields, see "Creating Custom Fields" on page 81.

4. Select the data columns to include in your report. These include built-in and custom fields.

   Select whether to use basic fields only or both basic fields and advanced fields from the **Available Columns** drop-down menu. Use the blue arrows to add and remove selected data columns.

   Note: The columns available in the list are determined by the organizational fields chosen in the step before.

5. Configure result sorting.

   Select fields to sort by and whether to sort the data in ascending or descending order

   Grouping and sorting options appear based on the data columns that you chose to include in the report. By default, the report is sorted by the organization field selected in the previous step.

6. Add a filter to show only results matching the entered value.

   For detailed information on Console's filters, please see "Reference: Reporting Filters" on page 153.

7. Create your report. You can also run it at this time.

   - Click **Create**: Save the report without running it. You are redirected to the Manage Report Types page where you can see the new report in the list of reports. Custom reports have **edit** and **delete** links, which differentiate them from Console's built-in reports. Both custom reports and built-in reports include a **copy** link for duplicating the report and a **run** link to view run settings and generate the report.
   - Click **Create and Run**: Save the report and run it. The new report is added to the Manage Report Types page, but first, you are redirected to the New Report page where you must finalize the report run settings and click the **Run Report** button to run the report.

Creating an Advanced Report

The following instructions describe how to create advanced reports. To view an example of an advanced report, see "Advanced Report Example: Transfer Sessions with High Packet Loss" on page 183. For more informationabout creating **basic** reports, see "Creating a Basic Report" on page 76.

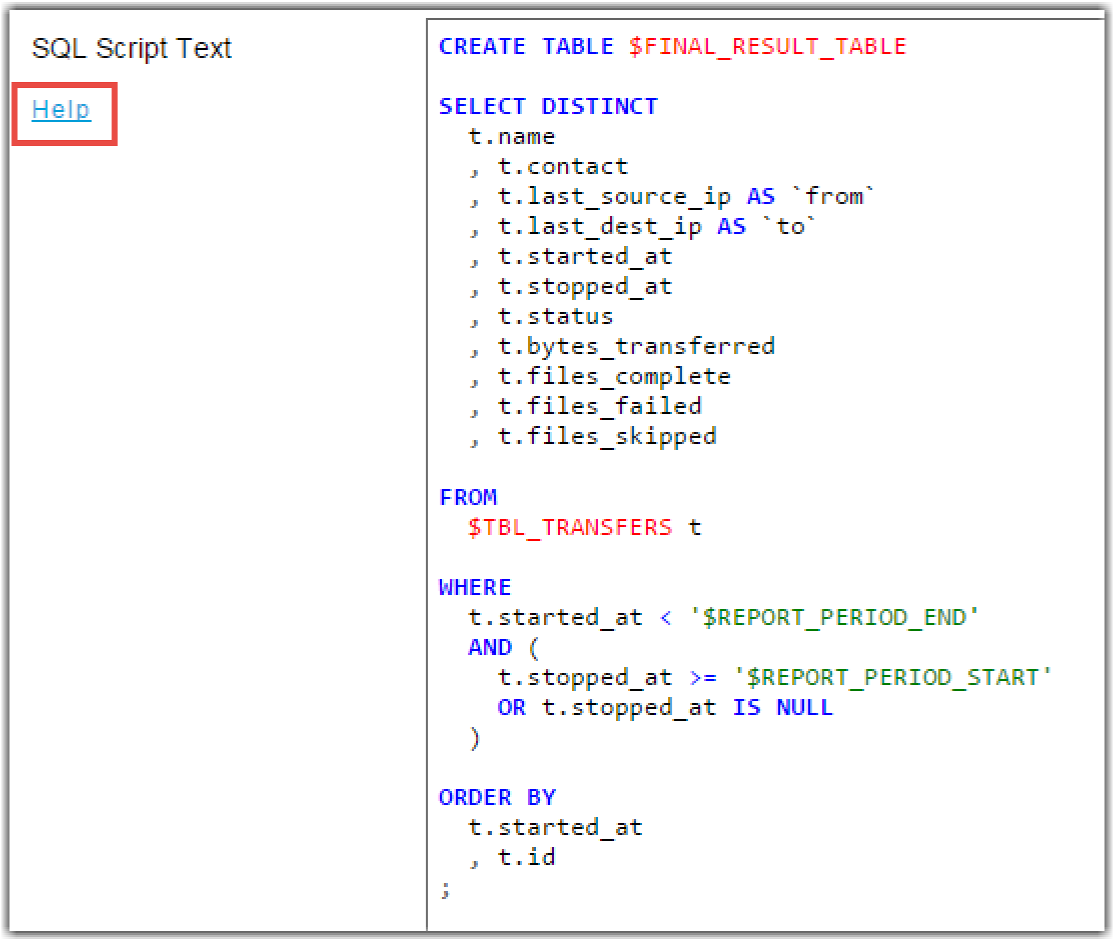
Important: Aspera recommends you read through the "Advanced Report Usage Notes" on page 166 before configuring an advanced report.

1. Go to **Reports > Manage Report Types**. Click the **New Advanced** button.

   Note: You may also modify an advanced report by clicking the **edit** action for an advanced report that is listed on the **Manage Report Types** page.

2. Enter a name for your report (limited to 75 characters) and a detailed description about the report.
3. Configure the SQL script text.

   For information on available SQL variables or database field references, click on the **Help** link.



```
CREATE TABLE $FINAL_RESULT_TABLE

SELECT DISTINCT
  t.name
  , t.contact
  , t.last_source_ip AS `from`
  , t.last_dest_ip AS `to`
  , t.started_at
  , t.stopped_at
  , t.status
  , t.bytes_transferred
  , t.files_complete
  , t.files_failed
  , t.files_skipped

FROM
  $TBL_TRANSFERS t

WHERE
  t.started_at < '$REPORT_PERIOD_END'
  AND (
    t.stopped_at >= '$REPORT_PERIOD_START'
    OR t.stopped_at IS NULL
  )

ORDER BY
  t.started_at
  , t.id
;
```

For a list of available reference variables, see:

- "Reference: SQL Variables for Advanced Reports" on page 155
- "Reference: Database Fields for Advanced Reports" on page 157,
- "Creating Custom Fields" on page 81

When creating advanced reports, you may specify a custom variable within the **WHERE clause** (for example, *$custom_variable*). Once declared within the SQL script text, you can to view and edit the variable by clicking **Edit Parameters** the Edit Advanced Report Template page. You are prompted to enter a value for the variable when you run the report.

4. Optional: Add a filter in the WHERE section of your script.
   For example, this example script filters out transfers that do not have a reported policy and transfers that do not fall within the specified date range.

```
...
WHERE
 ts.reported_policy IS NOT NULL
  AND
   ts.started_at < '$REPORT_PERIOD_END'
   AND (
    ts.stopped_at >= '$REPORT_PERIOD_START'
    OR ts.stopped_at IS NULL
  )
...
```

   For a list of available SQL variables you can use, "Reference: SQL Variables for Advanced Reports" on page 155.

5. Create your report. You can also run it at this time.

   - Click **Create**: Save the report without running it. You are redirected to the Manage Report Types page where you can see the new report in the list of reports. Custom reports have **edit** and **delete** links, which differentiate them from Console's built-in reports. Both custom reports and built-in reports include a **copy** link for duplicating the report and a **run** link to view run settings and generate the report.

- Click **Create and Run**: Save the report and run it. The new report is added to the Manage Report Types page, but first, you are redirected to the New Report page where you must finalize the report run settings and click the **Run Report** button to run the report.

Finalizing and Running a Report

Console requires you to finalize the report's run settings before running a report.

1. You can initiate finalizing and running a report in the following ways:
   - After configuring your basic or advanced report, click **Create and Run**.
   - Go to **Reports** > **Manage Report Types** from the Console menu and clicking the **run** link From the **Actions** column.
   - Go to **Reports > Run a Report**. Select a built-in or custom report from the list.
   - Go to **Reports** and click the **rerun** link from the Actions column for a recently run report.

   You are redirected to the New Report page.
2. Name the report.
3. Run the report now or schedule it to run later.
   - Select **Run Now**: Run this report immediately.
   - Select **Run Later**: Schedule a report by setting the run date. You may also select **Repeat** to schedule a repeating report.
4. Define the report period.

Option	**Description**
Report on	Select a pre-defined time period from the drop-down list. • last hour • last 24 hours • last week • month to date • last month • custom
Start date	Select the start date of this report. You must select **custom** in the drop-down menu to modify this field.
End date	Select the end date of this report. You must select **custom** in the drop-down menu modify this field.
Time zone	Select the time zone for this report.

5. Enter values for your custom SQL variables under **Report Parameters**. If there are no values, no custom variables were specified for this report.

   For more information on custom variables, see "Editing Custom Variables" on page 80.
6. Optional: Enter an email address and click the **Add** button to email a recipient a copy of this report.

   After adding an email address, select whether the report is sent as an XLSX or a CSV file.
7. Optional: Choose additional file formats (XLSX and CSV). These files can be downloaded after the report has been generated.
8. Click **Run Report** after finalizing your settings.

Your generated report is listed on the Scheduled and Recently Run Reports page. When viewing your report, you have the following options:

- For a custom report, click **Edit Report Type** to configure report.
- To run the report again, click **Rerun**.
- If you chose to export your report in CSV or XLSX, click the respective button to download the files.

Editing Custom Variables

When creating advanced reports, you can specify a custom variable within the WHERE clause. For example, to create a search by contact, enter:

```
...
WHERE
   contact = '$CONTACT_MATCH'; # $CONTACT_MATCH is the custom variable.
...
```

Once you declare the variable within the SQL script text, you can view and edit the variable on the Edit Advanced Report Template page.

1. To edit a custom SQL variable used in an advanced report, go to **Reports** and click the **Manage Report Types** button. Find the advanced report and click **edit**. Click **Edit Parameters**.
2. Find the custom variable you want to configure and click **edit**.
3. Select the desired variable type from the **Type** drop-down menu.

Variable Type	Description
string	The value of this variable must be a string.
integer	The value of this variable must be an integer.
date	The value of this variable must be a valid date. Click the calendar icon to select a valid date.
ip	The value of this variable must be a valid IP Address.

4. Optional: Allow the user running the report to leave the variable undefined by clearing **Is field required?**. Custom variables are required by default.

   If **Is field required?** is selected, a user running this report is required to enter a value for the custom variable to run the report.

   Note: If the custom variable is *not* required *and* it is used with the **AND** operator, then write the report query as follows:

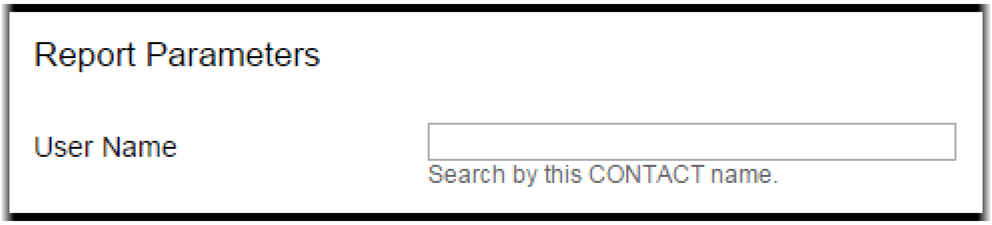
```
...
WHERE
...
   AND (
     t.status = '$FOO'
     OR '$FOO' = ''
   )
...
```

   Failure to include `OR '$FOO' = ''` results in an empty report because the data is filtered by `t.status = ''`, which is always false.

5. Optional: Define the variable name that is displayed when Console asks for the value of this variable. For example, if you want to search by contacts and included a custom variable named *$CONTACT_MATCH* in your SQL script, Console by default prompts the user running the report to enter a value for "Contact Match." If you enter "User Name" in the **Label** field, Console asks for a value called "User Name" instead and matches the result to *$CONTACT_MATCH*.
6. Optional: Add a hint to remind the user the purpose of this variable.

Continuing the previous example, if your custom variable, *$CONTACT_MATCH*, is used to search your database for contacts matching the value of this variable, a possible hint is: "Search by this CONTACT name."

When running the report, the user is prompted with the following:



Report Parameters

User Name

Search by this CONTACT name.

7. When finished, click **Update**.

Creating Custom Fields

Custom fields are used to specify rules for automatically populating fields in basic and advanced reports.

1. Go to **Configuration > Custom Fields**.
2. Click **New Custom Field**.
3. Select **transfer** or **file** from the **Level** drop-down list, depending on whether the new custom field stores transfer- or file-related content.
4. Enter a name for the custom field. The name must be unique and lowercase.

   The resulting SQL name is prefixed with "cf_". For example, the field name "metadata" appears as "cf_metadata".

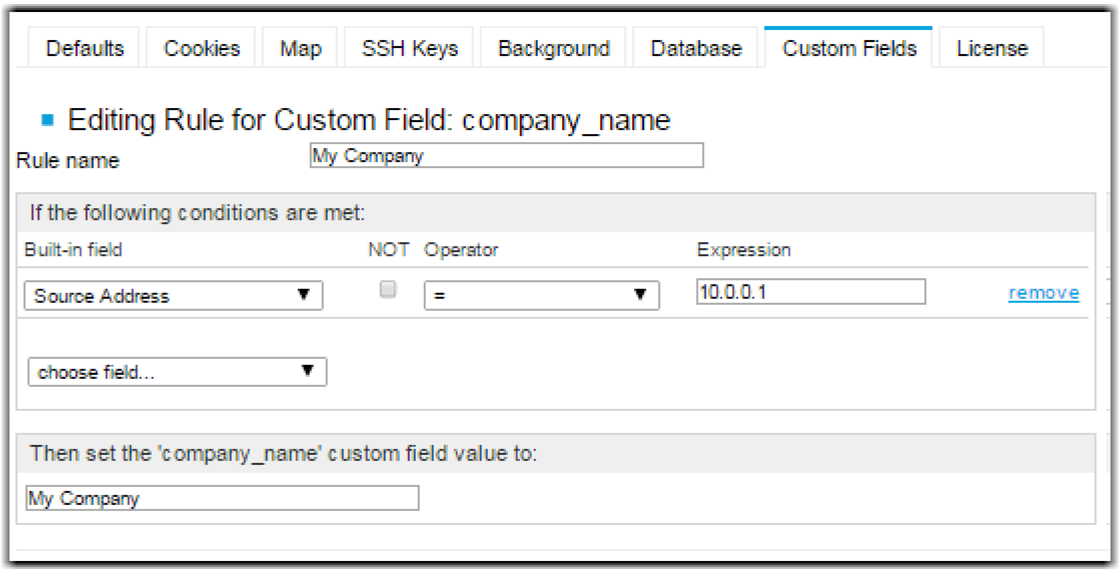
   Note: Custom fields appear in the database with the "cf_" prefix. Custom fields are utilized in the $TBL_FILES and $TBL_TRANSFER tables.
5. Enter the start date (date on which to start custom field calculation).
6. Enter a custom field description.
7. Click **Create**.
8. Create and associate new rules for your custom field.

   Rules are conditions that define when the custom field to comes into effect. To set up the rule's conditions, configure the following settings:

   - Select a built-in field from the drop-down list.
   - Enter an operator.
   - Enter an expression.
   - Enter the value Console uses to populate the custom field if conditions are met.

   For a list of field names and definitions, see “Reference: Built-In Fields for Custom Field Rules” on page 151.

   For example, to create a custom field that is populated with your company name, create a new custom field and associate it with the following rule:



Editing Rule for Custom Field: company_name

Rule name: My Company

If the following conditions are met:

Built-in field	NOT	Operator	Expression
Source Address	□	=	10.0.0.1
choose field...			

Then set the 'company_name' custom field value to:

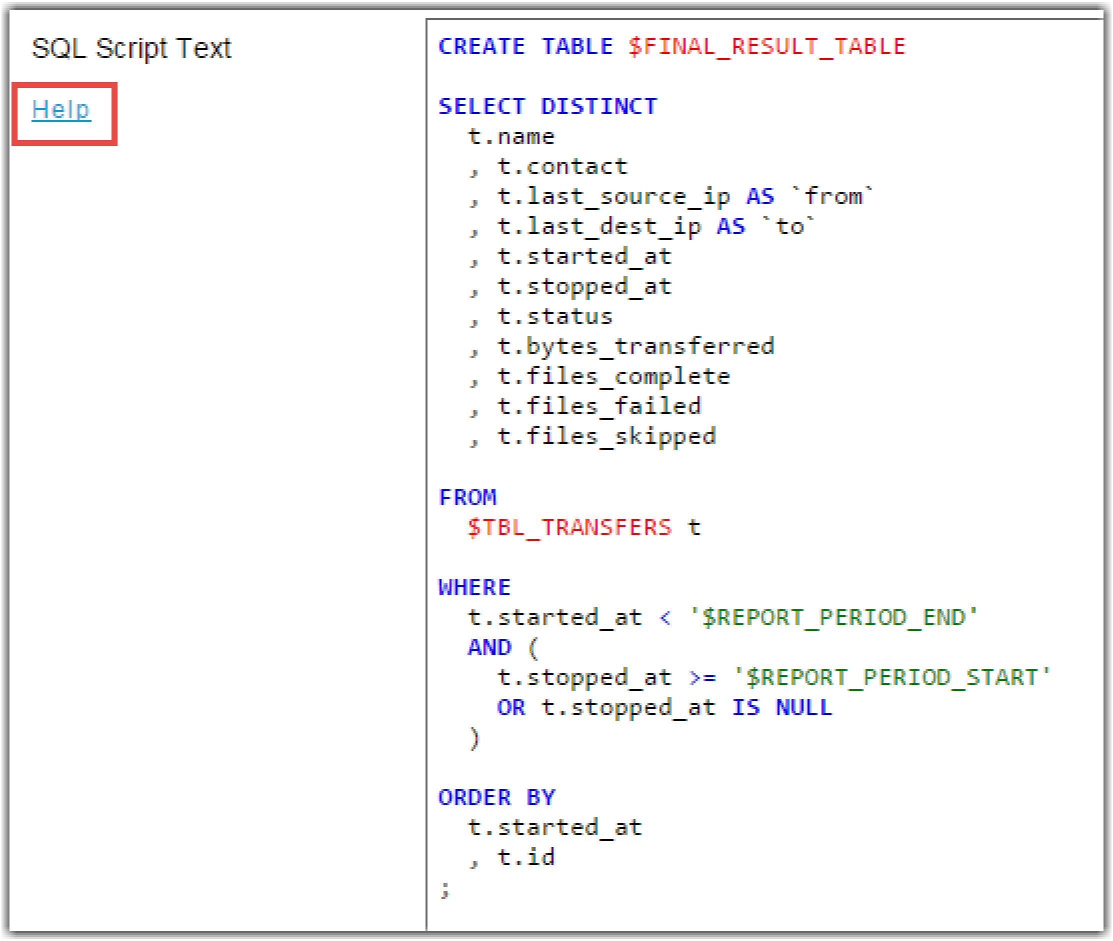
My Company

For examples on using matchers, see "Using Matchers in Custom Rules" on page 152.

9. Click **Create**.

For each custom field, you can create multiple rules that populate with different values based on various conditions. When multiple rules are present, Console uses the first rule listed (as long as it matches the condition). To modify the order of the custom field rules, use the drag-and-drop function to move the rules in the list.

When creating an advanced report, you can find your available custom fields by clicking the **Help** link in the **SQL Script Text** section.



```
CREATE TABLE $FINAL_RESULT_TABLE

SELECT DISTINCT
  t.name
  , t.contact
  , t.last_source_ip AS `from`
  , t.last_dest_ip AS `to`
  , t.started_at
  , t.stopped_at
  , t.status
  , t.bytes_transferred
  , t.files_complete
  , t.files_failed
  , t.files_skipped

FROM
  $TBL_TRANSFERS t

WHERE
  t.started_at < '$REPORT_PERIOD_END'
  AND (
    t.stopped_at >= '$REPORT_PERIOD_START'
    OR t.stopped_at IS NULL
  )

ORDER BY
  t.started_at
  , t.id
;
```

For an example of using a custom field in a report, see "Advanced Report Example: Transfer Sessions with High Packet Loss" on page 183.

Configuring SSH Keys

SSH Keys

SSH keys provide a more secure way to authenticate than using passphrases. Console generally uses SSH keys for two purposes:

- Authentication to administer and configure a node.
- Authentication to make a transfer from one node to another.

You can store keys and find a list of existing keys by navigating to the SSH Private Key page in either of two locations:

- **Personal Preferences:** Select **Preferences** from the drop-down menu next to your username in the upper right-hand corner. Then, select the **SSH Keys** tab.
- **Console Configuration:** Go to **Configuration** > **SSH Keys** from the Console menu.

For more information on storing keys, see "Storing SSH Keys on Console" on page 83.

The steps to using an SSH key differs if you are using an SSH key to make a transfer or using one to make a transfer to nodes with endpoints that use SSH keys.

Using SSH Keys in Transfers

A user must add an SSH key in his personal preferences before he can use that key in a transfer. Even if the SSH key is configured in Console Configuration settings, if the user did not the key in his personal preferences, the key does not appear when he enters the credentials for a node to set up a transfer.

Making Transfers to Nodes With Endpoints that Use SSH keys

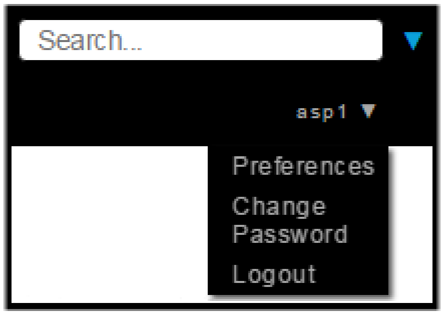
When making transfers to nodes with endpoints using SSH keys, the transfer user on the initiating node also needs to have the private key in the **.ssh** folder. For a walkthrough of this process, see "Transferring Files with an Endpoint Using SSH Keys" on page 84.

Storing SSH Keys on Console

Console uses a node machine's private key to authenticate into the machine using public key authentication. You must first store in Console the private key paired with the public key on the node machine. You can store private keys privately in your user preferences or globally in Console configurations. These SSH keys can then be used to authenticate endpoints or transfers.

1. Go to your private SSH keys or Console's stored SSH Keys.

   - **Personal Preferences:** Select **Preferences** from the drop-down menu next to your username in the upper right-hand corner.

     Then, click the **SSH Private Keys** tab.

- **Console Configuration:** Go to **Configuration > SSH Keys**.

2. Click **New SSH Private Key**.
3. Enter a descriptive name to represent the SSH key in Console.
4. Enter the filename of the key on the initiating node in the **Filename on node** field.

   To use this key in a transfer, Console searches the initiating node for a file that matches the filename given. You can enter the filename with or without the full path. For example, Console accepts both `my-ssh-key` and `path/to/my-ssh-key`.

   Make sure you or the node administrator uploads the SSH key to the node or nodes from which you want to initiate transfers.
5. Click **Choose File** to upload the same private key uploaded to the initiating nodes to Console.

   If you do not have direct access to the node, you must ask the node administrator to give you the private key. This key is used to browse and configure the node.
6. Enter and confirm the passphrase of the key, if any.
7. Click **Save**.
8. Click **Test** to test the new SSH private key.

   Provide the following information:

   - The address of the computer that has the paired public key installed in their **authorized_keys** file.
   - The corresponding user name.

   Then, click **Connect with SSH Key** to test against the computer.

   Tip: If the connection fails, contact the node administrator to make sure the public key is properly installed in the **authorized_keys** file.

Transferring Files with an Endpoint Using SSH Keys

The objective of this example is to set up two nodes in Console to transfer files from one node to the other using public key authentication.

- *User A:* Transfer user found on Node A.
- *User B:* Transfer user found on Node B.
- *Node A:* The node initiating the transfer. This node holds the private key.
- *Node B:* The node receiving the files. This node holds the public key matching the private key in Node A. We set up an endpoint using SSH keys for this node.

Note: For the purpose of this example, both nodes are Linux machines.

1. Generate a private key as User A on Node A with the following command:

   ```
   # ssh-keygen -t rsa
   ```

   Choose the default location to store this new private key (Default is `~/.ssh`).
2. Make sure **User A** has read and write permissions for the private key file.

   ```
   $ chmod 600 ~/.ssh/id_rsa
   $ chmod 644 ~/.ssh/id_rsa.pub
   ```
3. Copy the SSH public key into User B's **authorized_keys** file on Node B.

   ```
   # cat ~/.ssh/id_rsa.pub >> ~./ssh/authorized_keys
   ```
4. In Console, add Node A as a managed node and Node B as an unmanaged node.
5. Go to **Configuration > SSH Keys** and upload the private key to Console. This key should be paired with the public key copied to Node B.
6. Go to **Nodes** and edit Node B. Click **Endpoints** and add a new endpoint. Choose to use the SSH key that was uploaded to Console.

7. Make a transfer from User A on Node A to the saved endpoint on Node B.

Working With SSL

Installing a Signed SSL Certificate Provided by Authorities

In a default IBM Aspera Console installation, Apache generates and uses a self-signed SSL certificate. Install a signed certificate provided by authorities to secure your server.

1. Create a working directory

   Go to **Start menu > All Programs > Accessories > Command Prompt** and create a new working directory:

   ```
   > mkdir c:\ssl
   > cd c:\ssl
   ```

2. Copy **openssl.cnf** to your working directory.

   Enter the following commands in your Command Prompt window:

OS Version	Commands

32-bit Windows	`> copy "c:\Program Files\Common Files\Aspera\common\apache\conf\openssl.cnf" "c:\ssl\"`
	`> cd c:\ssl`

64-bit Windows	`> copy "c:\Program Files (x86)\Common Files\Aspera\common\apache\conf\openssl.cnf" "c:\ssl\"`
	`> cd c:\ssl`

3. Generate your Private Key (`.key`) and Certificate Signing Request (CSR) (`.csr`):

   a) Run the following **`openssl`** command, where *key_name* is the name of the unique key that you are creating and *csr_name* is the name of your CSR:

   ```
   $ openssl req -new -config "c:\ssl\openssl.cnf" -nodes -newkey rsa:2048 -keyout
   key_name.key -out csr_name.csr
   ```

   Note: Windows does not, by default, have a `C:\ssl\` directory. If the directory does not exist on your server, create the directory:

   ```
   > mkdir c:\ssl
   ```

   b) Configure the certificate's X.509 attributes.

   Important: The `Common Name` field must be filled in with the fully qualified domain name of the server to be protected by SSL. If you are generating a certificate for an organization *outside of the US*, see https://www.iso.org/obp/ui/#search/code/ for a list of 2-letter, ISO country codes.

   For example:

   ```
   Generating a 1024 bit RSA private key
   ....................++++++
   ................++++++
   writing new private key to 'my_key_name.key'
   -----
   You are about to be asked to enter information that will be incorporated
   into your certificate request.
   What you are about to enter is what is called a Distinguished Name or a DN.
   There are quite a few fields but you can leave some blank
   For some fields there will be a default value,
   If you enter '.', the field will be left blank.
   -----
   Country Name (2 letter code) [US]: US
   ```

```
State or Province Name (full name) [Some-State]: CA
Locality Name (eg, city) []: Emeryville
Organization Name (eg, company) [Internet Widgits Pty Ltd]: IBM Aspera
Organizational Unit Name (eg, section) []: ASP
Common Name (i.e., your server's hostname) []: faspex.asperasoft.com
Email Address []: faspex@asperasoft.com
```

c) When prompted, you can enter extra attributes, including an optional challenge password.

Manually entering a challenge password when starting the server can be problematic in some situations (for example, when starting the server from system boot scripts). You can skip entering values for any extra attribute by hitting the `Enter` button.

```
...
Enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
```

After finalizing the attributes, the private key and CSR are be saved to your root directory.

Important:

- If you make a mistake when running the OpenSSL command, discard the generated files and run the command again.
- After successfully generating your key and Certificate Signing Request, secure your private key, as it cannot be re-generated.

4. Send CSR to your signing authority.

   You now need to send your newly generated, unsigned CSR to a Certifying Authority (CA). Once the CSR has been signed, you have a real certificate. Follow the key provider's instructions to generate and submit both your private key and the Certificate Signing Request (CSR) to acquire the certificate.

   Important: Some Certificate Authorities provide a Certificate Signing Request generation tool on their Website. Check with your CA for additional information.

5. If your CA returns the SSL certificates to you in PFX (`.pfx`) format, use the **`openssl`** command convert the certificates to PEM (`.pem`) format:

   ```
   > openssl pkcs12 -in path/to/pfx_cert_name.pfx -nocerts -out path/to/key_name.key -nodes
   > openssl pkcs12 -in path/to/pfx_cert_name.pfx -nokeys -out path/to/cert_name.crt -nodes
   ```

6. Store your certificates on your server.

   For example:

   - `my_server.crt`
   - `my_server.key`

   Your certificate provider may require you to also install an Intermediate CA Certificate file. Copy the file to `C:\Program Files (x86)\Common Files\Aspera\Common\apache\conf\server-ca.crt`

7. Install the SSL certificate with the following command:

   ```
   > asctl apache:install_ssl_cert cert_file_path key_file_path [chain_file_path]
   ```

   For example:

   You can find the installed certificate and key at the following locations:

   - `C:\Program Files (x86)\Common Files\Aspera\Common\apache\conf\server.crt`
   - `C:\Program Files (x86)\Common Files\Aspera\Common\apache\conf\server.key`

Generating and Installing a New Self-Signed SSL Certificate

Generate a self-signed certificate if you don't plan on sending your certificate to be signed by a Certified Authority (CA), or if you want to test your SSL implementation while waiting for the CA to sign your certificate.

A self-signed certificate is a temporary certificate that is valid for 365 days. Self-signed certificates are not meant to be used in your production environment. Users accessing your server are warned by their browser warn them that your server is not secure.

By default, IBM Aspera Console uses a generated, self-signed certificate as a placeholder until you can install a certificate signed by authorities.

You can find the installed certificates at:

- C:\Program Files (x86)\Common Files\Aspera\Common\conf\server.crt
- C:\Program Files (x86)\Common Files\Aspera\Common\conf\server.key

Generate a self-signed certificate using **openssl** command, where *key_name* is the name of the unique key that you are creating and *cert_name* is the name of your certificate file:

```
> openssl x509 req -days 365 -in csr_name.csr -signkey key_name.key -out cert_name.crt
```

Regenerating Self-Signed SSL Certificate (Apache)

When you initially set up Console on your system a pregenerated, self-signed SSL certificate is also installed. If you have changed your Apache hostname, regenerate the self-signed certificate by following the instructions below.

1. Open a Command prompt window and run the *asctl* command.

   In a command prompt window (**Start menu** > **All Programs** > **Accessories** > **Command Prompt**), run the following command to generate a new, self-signed SSL certificate for your installation of Console (where you will replace the HOSTNAME with your Apache server's IP address or host name):

   ```
   > asctl apache:make_ssl_cert HOSTNAME
   ```

   Answer **yes** when prompted to overwrite the existing certificate.
2. Confirm that your certificates are updated.

   Check the following location to confirm your self-signed SSL certificates have been updated:

OS Version	File
32-bit Windows	• C:\Program Files\Common Files\Aspera\Common\apache\conf\server.crt • C:\Program Files\Common Files\Aspera\Common\apache\conf\server.key
64-bit Windows	• C:\Program Files (x86)\Common Files\Aspera\Common\apache\conf\server.crt • C:\Program Files (x86)\Common Files\Aspera\Common\apache\conf\server.key

Working with Shares and Directory Services

Console and Shares on Same Machine

Important: This topic assumes that you have already installed Shares. If you have not installed Shares yet, please see the IBM Aspera Shares Administrator's Guide.

If you installed Console on the same machine as Shares, you must update the host and port settings in Shares' `database.yml` file. Your **database.yml** file can be found in the following directory:

```
C:\shares\www\config\database.yml
```

Open **database.yml** with a text editor and perform the following modifications:

- Comment out the socket location.
- Change the host to 127.0.0.1.
- Change the TCP port to 4406.

After performing these modifications, your `database.yml` file should look similar to the example below.

```
production:
  adapter: mysql2
  encoding: utf8
  reconnect: false
  database: web_production
  pool: 5
  username: admin
  password: v00d00
  # socket: /tmp/mysql.sock
  # socket: /var/lib/mysql/mysql.sock
  host: 127.0.0.1
  port: 4406
```

Configuring the Directory Service

Important: You must install Shares locally or on a remote host before you can configure a directory service. For information on installing the latest version of Shares, please review the Administrator's Guide.

Before continuing, please ensure that the following prerequisites have been satisfied:

- Shares is installed locally or on a remote host: For instructions on how to install Shares on the same machine as Console, see "Console and Shares on Same Machine" on page 87.
- Console is installed on the same machine as Shares: Configure your Shares web server to use a non-standard HTTPS port (for example, 8443, rather than 443). See the Shares Administrator's Guide (the "Setting up Shares" topic).
- Console is installed on the same machine as Shares: Configure your Shares database with the correct host and port settings. For more information, see "Working with Shares and Directory Services" on page 87.

1. Go to **Accounts > Directories** from the Console menu.
2. Select **Remote Authentication** to enable remote authentication so that Console can access the groups and users on your Shares server.
3. Enter the base URL.

   Shares users and groups are authenticated through this Node API base URL. The standard base URL is `https://`*`shares_IP_address`*`/api/v1`.

   Note: Because you must use HTTPS to connect to your Shares directory service, ensure that your Node API base URL uses HTTPS, rather than HTTP.
4. Enter the admin Shares user with API login capabilities. You can set API login abilities for a user from the Shares Admin page (**Admin > Users > *username* > Security > API login**).
5. Click **Save and test settings**.

Once the directory service is successfully connected, you can add remote users and remote groups by boing to **Users > Groups**. For more information, see "Adding Remote Users" on page 88 and "Adding Remote Groups" on page 89.

Adding Remote Users

1. Go to the **Users** menu. Click **Add Remote User**.

Note: The **Add Remote User** button does not appear if you have not configured the directory service.

2. Enter the full or partial name of an existing remote user. Click **Search**.
3. Once your search results appear, select the remote user by clicking **Add**.
4. Configure the remote user's Console permissions and assign the user to a group.

Adding Remote Groups

1. Go to **Groups**. Click **Add Remote Group**.

   Note: The **Add Remote User** button does not appear if you have not configured the directory service.
2. Enter the full or partial name of an existing remote user. Click **Search**.

   Note: Console does not find remote groups that start with a backslash (\) or an asterisk (*****). Avoid naming groups that start with these characters.
3. Once your search results appear, select the remote group by clicking **Add**.
4. Configure the remote group's transfer paths and members.

Working with SAML

SAML and Console

IBM Aspera Console supports Security Assertion Markup Language (SAML) 2.0, an XML-based standard that allows secure web domains to exchange user authentication and authorization data. With the SAML model, you can configure Console as a SAML *online service provider (SP)* that contacts a separate online *identity provider (IdP)* to authenticate users. Authenticated users can then use Console to access secure content.

With SAML enabled, Console redirects a user to the IdP sign-on URL. The user signs in with the IdP and the IdP sends a SAML assertion back to Console. When a SAML user logs in to Console for the first time, Console automatically creates a new user account based on the information provided by the SAML response. Any changes subsequently made to the account on the DS server are not automatically picked up by Console. For more information about user provisioning for SAML users, see "User Accounts Provisioned by Just-In-Time (JIT) Provisioning" on page 90.

IdP Requirements

To use SAML with Console, you must already have an identity provider (IdP) that meets the following requirements:

- Supports SAML 2.0
- Able to use an HTTP POST Binding.
- Able to connect to the same directory service that Console uses.
- Not configured to use pseudonyms.
- Can return assertions to Console that include the entire contents of the signing certificate.
- If prompted, set to sign the SAML response. (Signing the SAML assertion is optional.)

Configure the SAML IdP

Before configuring SAML in Console, make sure you configure your IdP to send a correct SAML response to Console. For more information, see "Configuring Your Identity Provider (IdP)" on page 91.

For instructions on configuring SAML, see "Configuring SAML" on page 92.

SAML and Directory Services

Console supports the use of both SAML and directory services. If you configure both services to Console, ensure the services use different Active Directory domains. Aspera advises against configuring LDAP directly to Console if the SAML IdP acts as a frontend for the same Active Directory domain.

Bypassing the Default SAML IdP

Console provides a mechanism for users to bypass the SAML redirect and log in using a local username and password. This feature allows admins to correct server settings, including a mis-configured SAML setup, without logging in through SAML.

To bypass the SAML login, add `login?local=true` to the end of the login URL. For example:

```
https://198.51.100.48/login?local=true
```

User Accounts Provisioned by Just-In-Time (JIT) Provisioning

When a SAML user logs in to IBM Aspera Console for the first time, Console automatically creates a new user account based on the information provided by the SAML response. If the SAML response also contains group information, and that group does not yet exist in Console automatically creates a new SAML group for each group of which the user is a member. For more information about SAML groups, see "Creating SAML Groups" on page 92.

Existing Local Users and Local Groups

When a SAML user first logs in, Console creates a new Console SAML user account for the SAML user. Console cannot create that account if a local user account with the same name already exists. You must first delete the local user (so that Console can create the user account) before the SAML user can log in.

Whenever a SAML user logs in, Console updates the user's SAML group memberships:

SAML directory (IdP)	Console groups	Result	User allowed to log in?
The SAML user is a member of a SAML group	A SAML group with the same name does not exist in Console	Console creates a SAML group with the same name and adds the associated SAML user account to the group	Yes
The SAML user is a member of a SAML group	A SAML group with the same name already exists in Console	Console adds the associated SAML user account to the Console SAML group	Yes
The SAML user is not a member of a SAML group	The SAML user account in Console is part of a SAML group	Console removes the associated SAML user account from the SAML group	Yes
The SAML user is a member of a SAML group	A local group with the same name already exists in Console	Console cannot create the SAML group. An admin must delete the local group or convert the local group to a SAML group before the SAML user can log in	No

Configuring Your Identity Provider (IdP)

IdP Requirements

To use SAML with Console, you must already have an identity provider (IdP) that meets the following requirements:

- Supports SAML 2.0
- Able to use an HTTP POST Binding.
- Able to connect to the same directory service that Console uses.
- Not configured to use pseudonyms.
- Can return assertions to Console that include the entire contents of the signing certificate.
- If prompted, set to sign the SAML response. (Signing the SAML assertion is optional.)

IdP Metadata Formats

You must configure formats to set up your IdP to work with Console:

Tag	Format
NameID Format	`urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified`
Entity ID	https://*_ip*/auth/saml/*provider_id*/metadata/
Binding	`urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST`
Callback URL	https://*_ip*/auth/saml/callback

Tag	Format
Entity ID	https://*server_name_or_ip*/aspera/console/auth/saml/metadata
ACS	https://*server_name_or_ip*/aspera/console/auth/saml/callback
Base URL	https://*server_name_or_ip*/aspera/console

If the IdP is capable of reading SAML XML metadata for a service provider, you can upload a saved XML metadata file to configure the IdP. You can retrieve the XML metadata for an existing Console by going to an existing SAML configurations on the application (**System Administration > Authentication > SAML Configurations**), selecting a configuration, clicking the **Metadata** button,`https://`*`server_ip`*`/auth/saml/metadataaspera/console/auth/saml/metadata` and saving the XML as an XML file.

SAML Assertion Requirements

Console: expects assertion from an IdP to contain the following elements:

Default Attribute	Console User Field	Required
`NameID` / `SAML_SUBJECT`	Username	Yes, with the format: `urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified`
`email`	Email address	Yes
`given_name`	First name	YesOptional
`surname`	Last name	Optional
`member_of`	SAML group	Necessary for SAML groups

Tip: All attributes other than `NameID` or `SAML_SUBJECT` can also use the `urn:oasis:names:tc:SAML:2.0:attrname-format:basic` format.

Tip: You can configure the Console user fields to map to different attributes in the Console SAML configuration settings.

Configuring SAML

Before following the instructions below, have the following information on hand:

- IdP Single Sign-On URL (SSO)
- IdP Certificate Fingerprint

1. Go to **Accounts > SAML** to open the SAML Configuration page.
2. Select **SAML Authentication**.
3. Enter the **IdP SSO Target (Redirect) URL**.
4. Allow access to Console through the redirect URL by adding the URL to the `AcceptedHosts` list in `(C:\Program Files (x86)\Aspera\Management Console\config\console.yml)`.

   For more information, see "Allowing Access to Console at Defined Hostnames" on page 14.
5. Enter the certificate fingerprint.
6. Choose the certificate fingerprint algorithm that matches the certificate fingerprint.
7. Click **Save**.

Creating SAML Groups

You can create SAML groups in Console without waiting for a SAML user in a SAML group to log in. Console verifies membership in the SAML group when the members log in with their SAML credentials.

SAML groups are created when:

- An admin creates a SAML group in the web application.
- An admin converts a local group into a SAML group in the web application.
- A SAML user that is a member of a SAML group logs in using SAML credentials.

After creation, you can add pre-existing SAML users to the group. For more information, see "User Accounts Provisioned by Just-In-Time (JIT) Provisioning" on page 90

Create a New SAML Group

1. Go to **Accounts > Groups**.
2. Click **New Group** to create a SAML group.
3. Select **SAML** from the **Directory** drop-down menu.
4. Click **Create** to create the SAML group.

Convert an Existing Local Group to a SAML Group

1. Go to **Accounts > Groups**.
2. Click **edit** for an existing group and select **SAML** from the **Directory** drop-down menu.
3. Click **Update**.

Convert a SAML Group to a Local Group

1. Go to **Accounts > Groups**.
2. Click **edit** for an existing group and select **Local** from the **Directory** drop-down menu.
3. Click **Update**.

Note: If you convert a SAML group to a local group, and a SAML user is part of a SAML group with the same name, that SAML user cannot log into Console. For more information, see "User Accounts Provisioned by Just-In-Time (JIT) Provisioning" on page 90.

Set Up Active Directory Federation Services (ADFS) for Console SAML

Register a new relying party trust using Console SAML metadata.

Before registering a new relying party trust, configure a Console SAML configuration through the Web UI (). For more information, see "Configuring SAML" on page 92.

1. Save the SAML metadata file:
   a) Go to `*server_url*/auth/saml/metadata`.

      For example, if your server is `console.aspera.us`, go to `console.aspera.us/auth/saml/metadata`.

   b) Save the page as an XML (`.xml`) file.
2. On the server hosting AD FS, launch the **ADFS Management Console**.
3. Add a new relationship (click the plus next to Trust Relationships).
4. Right click on **Relying Party Trust** and select **Add Relying Party Trust**
5. On the Add Relying Party Trust Wizard window, click **Start**.
6. Choose **Import data about the relying party from a file**.
7. Browse to the location of your metadata file, select it, and click, **Open**.
8. Click **Next** and choose a unique display name.
9. Choose **Permit all users to access this relying party**.
10. Click **Next** until you see the **Open the Edit Claim Rules dialog for this relying party trust when the wizard closes**. Clear that option.
11. Close the window.
12. Right click on the newly created relying party and select **Properties > Advanced**.
13. Change the **Secure hash algorithm** to SHA-256 and click **OK**.
14. Test the configuration by logging in to Console through the ADFS SAML configuration.

Backing Up Console Database

Back Up Console with asctl

There are two different ways to back up the Console database:

1. Through the `asctl` command, which backs up only the MySQL database. Use this method before a Console upgrade procedure, or to guard against possible database corruption.
2. Through the Console web UI, which backs up the MySQL database in addition to all the files required to fully restore the Console application. Use this method for disaster recovery purposes, in order to restore Console when the entire server is lost.

Back up the Console database using an **`asctl`** command. Open a Command Prompt (**Start > All Programs > Accessories >Command Prompt**) and execute the following command:

```
> asctl console:backup_database
```

This command uses **mysqldump** to create Console's MySQL database backup. The backup file, **aspera_console.sql**, is saved in the following directory:

OS Version	Path
32-bit Windows	`C:\Program Files\Aspera\Management Console\backup\<year-month-day_time>`
64-bit Windows	`C:\Program Files (x86)\Aspera\Management Console\backup\<year-month-day_time>`

For instructions on restoring your Console database, see "Restoring the Console Database" on page 94.

Backing Up Console with the Web UI

There are two different ways to back up the Console database:

1. Through the `asctl` command, which backs up only the MySQL database. Use this method before a Console upgrade procedure, or to guard against possible database corruption.
2. Through the Console web UI, which backs up the MySQL database in addition to all the files required to fully restore the Console application. Use this method for disaster recovery purposes, in order to restore Console when the entire server is lost.

1. Select **Configuration > Database** from the Console menu. Click **Back Up**.
2. Enter the desired path of the Console machine into the **Save to** field. This path is the destination folder for the **console_full_backup_YYYY-MM-DD_hhmmss** backup file.
3. Schedule the backup to **Run Now** or to **Run Later**.
   - Click **Run now**: Back up the database immediately.
   - Click **Run later**: Specify a time in the future or configure a repeating backup operation.
4. Click **Back Up Now**.

Once Console has been backed up, the backup file appears on the Database Backups page, where scheduled, current, and recent backups are listed. To view details on a particular backup, click anywhere in the backup's row.

Restoring the Console Database

You can restore any back up of a Console database as long as you have access to the backup file.

1. Stop Console.

   ```
   > asctl console:stop
   ```
2. Restore the Console database.
   - If you made a back up of the Console database with the asctl command, you can restore it with the following command:

     ```
     > asctl -v console:restore_database C:\path\to\dir
     ```

     For example:

     ```
     > asctl -v console:restore_database C:\Program Files (x86)\Aspera\Management
     Console\backup\2013-1-16_164305
     ```
   - If you made a back up of the Console database with the web UI, you can restore it with the following command:

     ```
     > asctl -v console:restore C:\path\to\dir
     ```

     For example:

     ```
     > asctl -v console:restore C:\temp\console_full_backup_2013-1-16_00.57.28_UTC
     ```

Important: The restore command does not support relative paths to the backup directory. The path must be an absolute path in order for the restore command to work.

3. Start Console.

```
> asctl console:start
```

Managing the MySQL Database

Configure MySQL Settings

You may want to modify the MySQL settings for security or management purposes.

To begin, open a Command Prompt (**Start menu > All Programs > Accessories > Command Prompt**) and execute the following commands:

Change the Database root Login Password

MySQL database's root account's password is set during the setup process. For security reason, it is recommended to update the password. Use the following command to change the password. Enter the new and old password when prompted:

```
> asctl mysql:set_root_password
```

Change the MySQL Port

By default, Console's MySQL uses TCP port 4406. Use the following command to change it.

```
> asctl mysql:port 1234
```

If the MySQL's port number is changed, you will need to provide the updated Console settings to all the nodes, and reflect the new settings in all the nodes' **aspera.conf** files.

Connecting Console to MySQL Running on a Separate Server

1. Set up the remote database on another server.
2. On the Console server, stop Console services and back up the local database:

```
> asctl console:stop
> asctl console:backup_database
```

3. Copy the generated backup to the remote server.
4. On the remote server, grant remote access privileges to Console:

```
mysql > CREATE USER 'mysql_user'@console_server_ip_address' IDENTIFIED BY 'mysql_password';
GRANT ALL PRIVILEGES ON *.* TO 'root'@'console_server_ip_address' WITH GRANT OPTION; FLUSH
PRIVILEGES;
```

For example:

```
mysql > CREATE USER 'remote_console_user@'10.0.174.47' IDENTIFIED
BY 'XRs9sJFF5ja1BGlKHYLwzQ=='; GRANT ALL PRIVILEGES ON *.* TO
'remote_console_user@'10.0.174.47' WITH GRANT OPTION; FLUSH PRIVILEGES;
```

5. On the remote server, migrate the Console database to the remote database using the Console backup:

```
> /opt/aspera/common/mysql/bin/mysql -h remote_server_ip_address -P port -umysql_username
-pmysql_password < path/to/console/db/backup
```

Note: The default MySQL port is 4406.

For example:

```
> /opt/aspera/common/mysql/bin/mysql -h 54.182.111.111 -P 4406 -uroot -ptopsecret < /opt/
aspera/console/backup/2015-07-01_23458/console.sql
```

6. On the remote server, verify that the migration was successful.

   Log in to the MySQL database and view the contents of the new database:

   ```
   > /opt/aspera/common/mysql/bin/mysql -h remote_server_ip_address -P port -umysql_username
   -pmysql_password
   mysql > use aspera_console;
   mysql > show tables;
   mysql > select * from fasp_nodes;
   mysql > use aspera_console_reports;
   mysql > show tables;
   ```

7. On the Console server, configure Console to use the remote database.

   a) Back up the `C:\Program Files [(x86)]\Aspera\Common\mysql\database.rb.yml` files.

   b) Edit `C:\Program Files [(x86)]\Aspera\Common\mysql\database.rb.yml`.

      Change:

      - `host` to the IP address of the remote database.
      - `port` to the MySQL port (4406, by default).
      - `password` to the remote MySQL database password.
      - `user` to the remote MySQL database user.

        Note: By default, there is no `user` field. Console defaults to the `root` user. Add a new line to configure a different, non-root user. For example, `:user: remote_console_user`.

      For example:

      ```
      ---
      ...
      :hostname: 54.182.111.111
      :port: 4406
      :task status:
         ...
         ...
      :user: remote_console_user
      :password: XRs9sJFF5ja1BGlKHYLwzQ==
      :setup_complete: true
      ```

      Save your changes.

   c) Edit `C:\Program Files [(x86)]\Aspera\Console\config\database.yml`.

      Locate the `production` and `production_reports` sections and change:

      - `host` to the IP address of the remote database.
      - `port` to the MySQL port (4406, by default).
      - `username` to the remote MySQL database user.
      - `password` to the remote MySQL database password.

      Note: By default, Console uses different users and passwords for the `production` and `production_reports` environments. To follow the same design, repeat the steps to grant access for a separate user that is dedicated to the `production_reports` environment (for example, `aspera_console_reports` in the following example).

      For example,

      ```
      ...
      production:
        reconnect: true
        encoding: utf8
        port: 4406
        adapter: mysql
        username: remote_console_user
      ```

```
  charset: utf8
  database: aspera_console
  host: 127.0.0.1
  collation: utf8_general_ci
  password: XRs9sJFF5ja1BGlKHYLwzQ==
production_reports:
  reconnect: true
  encoding: utf8
  port: 4406
  adapter: mysql
  username: remote_console_reports
  charset: utf8
  database: aspera_console_reports
  host: 127.0.0.1
  collation: utf8_general_ci
  password: DDfUMH+f3FAdHbwvRt+BQR==
```

Save your changes.

8. Shut down the local MySQL database and restart all other Console services.

```
> asctl mysql:disable
> asctl all:restart
```

If you need to restore the original configuration to use the local MySQL database, revert the `.yml` files and then run:

```
> asctl mysql:setup
```

Purging Data from Console UI

You can archive or purge data from Console (for example, purge all sessions before January 1, 2000) by clicking the **Purge** button from the Database Backups page and completing the fields.

1. Schedule Console to purge the data now or at a later date.
   - **Run now**: Purge the database immediately.
   - **Run later**: Specify a time in the future or configure a repeating purge operation.
2. Select time frame of data to purge.

   Choose the date by entering a number and selecting **day**, **week**, or **month** from the drop-down menu. Make sure the automatically updated date displayed next to the drop-down menu is the desired day before proceeding.
3. Choose the type of transfers to include.

   Select **All closed transfers** or choose from the following list:
   - All successful transfers
   - All cancelled transfers
   - All error transfers
   - All inactive transfers
   - All zero-byte transfers
4. Save the data being purged for archiving purposes.

   Select **Save data being purged?** and enter the desired **absolute** path into the **Save to** field (for example, /tmp/data or D:\data\). The purged data will then be stored in the file **purged_data.sql** in the directory: `[absolute path]\console_purge_YYYY-MM-DD_hhmmss\`.

   Tip: Saved purged data can be restored by following the instructions in "Restoring Purged Data" on page 98.
5. When finished, click the **Purge Now** or **Schedule Purge** button (depending on whether you selected **Run now** or **Run later** above).

Purging Data from the Command Line

Use the **aspera:db:archive** command to purge data from the Console database.

Run the rake task using the **asctl** command:

```
> asctl --trace console:rake aspera:db:archive
```

You can also use the BEFORE option to purge data from before a specific date.
For example:

```
> asctl --trace console:rake aspera:db:archive BEFORE=2019-07-01
```

Restoring Purged Data

If you selected **Save data being purged?** when you purged Console data using the UI, you can restore purged data with a MySQL data import.

1. Stop Console services.

   ```
   > asctl console:stop
   ```

2. Import saved data into MySQL:

   ```
   > cd "C:\Program Files (x86)\Common Files\Aspera\Common\mysql\bin"
   > mysql -uUSERNAME -pPASSWORD aspera_console < C:\path\to\purged_data.sql
   ```

3. Start Console services.

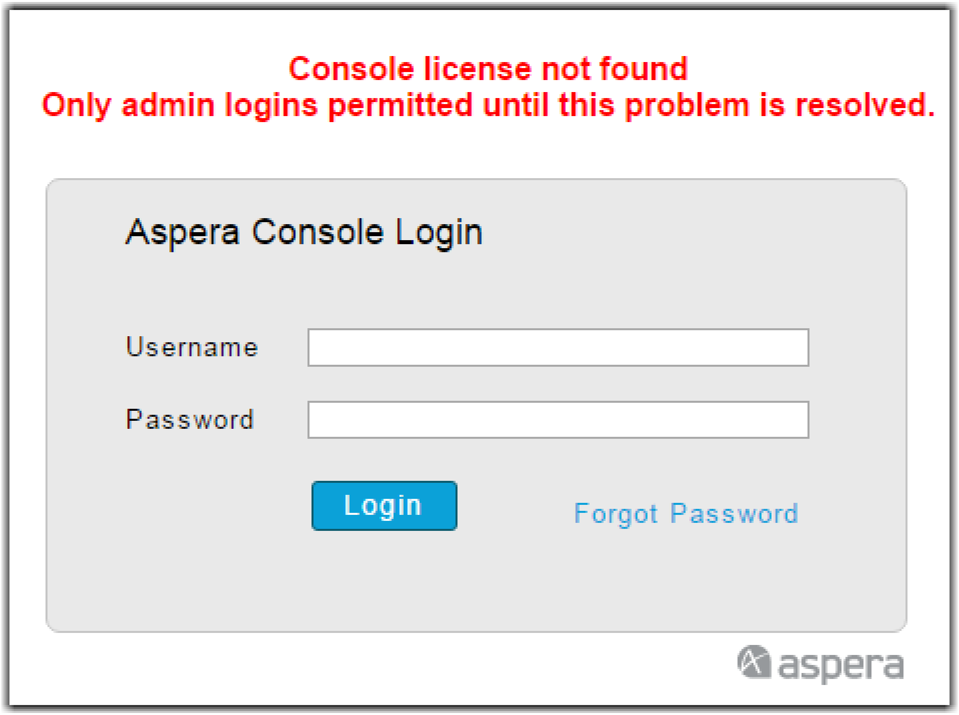
   ```
   > asctl console:start
   ```

Troubleshooting Console

Updating your Console License

Important: For purchasers of Aspera Enterprise, a license enabling Console as part of Enterprise can be downloaded from IBM Fix Central.

IBM Aspera Console requires a valid license key before you can configure users and send or receive packages. If your Console license has expired or cannot be found, the Console login screen displays the following message:


Console license not found
Only admin logins permitted until this problem is resolved.

An administrator must update the license before any other usage of Console is allowed for any user, including the administrator.

The license can be updated in the Console web UI or by running a rake task on the computer where Console is installed.

From the GUI:

1. Login with an administrator account and go to **Configuration > License**.
2. Click **Upload a license file** or paste the license text into the text window.
3. Click **Save**.

From the Command Line (Rake task):

1. Set the license text as an environment variable.

   ```
   > set LICENSE_TEXT='<ASPERA_LICENSE> <DETAILS expiration_date=... </KEY> </ASPERA_LICENSE>'
   ```

   In this example, only part of the license text is shown. You must paste the entire license text for the license to be valid.
2. Update the Aspera license:

   ```
   > asctl console:rake aspera:update_license
   ```

Restart Console Services

If Console is not working properly, it is recommended you restart the Console service utilizing the **asctl** command, so that Apache and MySQL continue to run uninterrupted. If the Console server's MySQL service is stopped, then Aspera Central will need to be restarted on all nodes to re-establish a connection.

To view the services that Console has installed on your Windows system, go to **Control Panel > Administrative Tools > Services**.

Service	Description
Apache HTTPD Server (Aspera)	Apache Server for Aspera Console.

Service	Description
Aspera Console	Aspera Console main application.
MySQL Server (Aspera)	MySQL Database for Aspera Console.

Right-click any of these services select **Restart** from the menu.

Execute the following **asctl** command to restart Console (while keeping Apache and MySQL running):

```
> asctl console:restart
```

For more asctl commands, see "asctl Command Reference" on page 109

Resetting Console Admin Password

To reset Console's administrator password, execute the following *asctl* command in a Command Prompt (**Start** > **All Programs** > **Accessories** > **Command Prompt**), replacing **name** with your existing admin login, **email** with the current admin password, and **password** with the new admin password.:

```
> asctl console:admin_user name email password
```

Encrypting and Decrypting Database.yml

The database.yml file contains passwords to Console components including its MySQL database. Console automatically encrypts these passwords so that they are not stored in plain text. Console also supports two rake tasks to encrypt or decrypt all passwords in the file.

If you made manual changes to the database.yml file, you can run the following command to encrypt all unencrypted passwords:

```
> asctl console:rake aspera:db:encrypt_database_passwords
```

If you want to see the passwords in plain text, you can run the following command to decrypt all encrypted passwords:

```
> asctl console:rake aspera:db:decrypt_database_passwords
```

Log Files

Console's log files are located in the following directories:

OS Version	Path
32-bit Windows	• **Console**: C:\Program Files\Aspera\Management Console\log\ • **asctl**: C:\Program Files\Common Files\Aspera\Common\asctl\ • **MySQL**: C:\Program Files\Common Files\Aspera\Common\mysql\data\mysqld.log • **Apache**: C:\Program Files\Common Files\Aspera\Common\apache\logs\
64-bit Windows	• **Console**: C:\Program Files (x86)\Aspera\Management Console\log\ • **asctl**: C:\Program Files (x86)\Common Files\Aspera\Common\asctl\ • **MySQL**: C:\Program Files (x86)\Common Files\Aspera\Common\mysql\data\mysqld.log • **Apache**: C:\Program Files (x86)\Common Files\Aspera\Common\apache\logs\

In Console's Apache HTTP server logs directory, you will find the following files:

- access_log
- error_log
- ssl_access_log
- ssl_error_log
- ssl_request_log

Important: All Apache logs are, by default, rotated by size (defaulting to 10MB files and only retaining the last 10 rotated logs).

httpd_template_windows.conf

```
/opt/aspera/common/apache/conf/httpd_template_windows.conf
```

- ErrorLog "|${log_path}bin/asrotatelogs ${log_path}logs/error_log 10M 10"
- CustomLog "|${log_path}bin/asrotatelogs ${log_path}logs/access_log 10M 10" common

httpd-ssl_template.conf

```
/opt/aspera/common/apache/conf/extra/httpd-ssl_template.conf
```

- ErrorLog "|${log_path}bin/asrotatelogs ${log_path}logs/ssl_error_log 10M 10"
- TransferLog "|${log_path}bin/asrotatelogs ${log_path}logs/ssl_access_log 10M 10"
- CustomLog "|${log_path}bin/asrotatelogs ${log_path}logs/ssl_request_log 10M 10" "%t %h %{SSL_PROTOCOL}x %{SSL_CIPHER}x \"%r\" %b"

You can further configure Console's Apache log settings by running the following commands in a Command Prompt (**Start** > **All Programs** > **Accessories** > **Command Prompt**):

Setting	Command
Specify an Apache log level (for example, error level)	`> asctl apache:log_level error`
Enable Apache log (set to notice)	`> asctl apache:enable_logs`
Disable Apache log (set to emerg level)	`> asctl apache:disable_logs`

Locate Configuration Files

Important: Aspera recommends that you DO NOT modify Console's configuration files manually. Instead, use the **`asctl`** command. For additional information on utilizing **`asctl`** commands, see the topic "asctl Command Reference" on page 109.

Console's configuration files are listed below. If you plan to modify these files, Aspera encourages backing up Console through the GUI or by using the *asctl* command. The *asctl* command is limited to backing up the Console database, while the GUI backs up the database, as well as all files required to fully restore the system. For instructions on backing up Console through the GUI, please see "Backing Up Console with the Web UI" on page 94. To back up Console's database using the *asctl* command, please see "Back Up Console with asctl" on page 93.

Component	Configuration File Path
Apache	• **(32-bit)** C:\Program Files\Common Files\Aspera\Common\apache\conf\apache.conf • **(64-bit)** C:\Program Files (x86)\Common Files\Aspera\Common\apache\conf\apache.conf

Component	Configuration File Path
MySQL	• **(32-bit)** C:\Program Files\Common Files\Aspera\Common\mysql\my.conf • **(64-bit)** C:\Program Files (x86)\Common Files\Aspera\Common\mysql\my.conf
Console	• **(32-bit)** C:\Program Files\Aspera\Management Console\config\*.yml • **(64-bit)** C:\Program Files (x86)\Aspera\Management Console\config\*.yml

Appendix

Configuring Console Defaults

The Console Defaults configuration page lets you to set up system defaults for Console, such as IP address and SSH timeout), as well as defaults for transfers (target rate, minimum rate, bandwidth policy, and so on) and login security. To access the Console and Transfer Defaults configuration page, select **Configuration** > **Defaults** from the Console menu.

Console Defaults

Item	Description
Console database IP address	Enter the Console database IP address.
Warn when database free space less than	The space watcher background jobs warns you when available space drops below the set number of gigabytes. Set to zero to disable space watcher warnings.
Skip non-error transfers older than	If a submitted transfer doesn't start after the specified number of minutes, then flag it as having an error.
Mongrel Timeout	Enter the number of seconds to wait for a response when testing mongrels.
Node Polling Timeout	Enter the number of seconds the SOAP Poller background process waits for a response when testing a node.
Mark Inactive Timeout	Enter the number of seconds Console waits before marking a session as inactive.
Submitted Job Timeout	Enter the number of seconds Console waits before marking a submitted Console job as inactive. **Note:** If record processing is slow, transfers can inaccurately be reported with the error: "DB never reported session end." To work around this issue, increase the submitted job timeout.
File Browsing Timeout	Enter the number of seconds to wait for a response from a node when browsing file lists (over and above the SSH timeout to connect).
File Browsing Max Items	Enter the maximum number of items to retrieve from a node when browsing file lists.
Default SSH Encryption	Select the default SSH encryption algorithm for non-Console nodes. **Note:** Console presents this algorithm as the standard, but you can change the algorithm when adding a new node.

Item	Description
Remote Login Connection Timeout	Enter the number of seconds Console waits before timing out when establishing a connection to a remote server.
Remote Login Response Timeout	Enter the number of seconds Console waits before timing out when waiting for the remote server's response.
SSH Timeout	Enter the timeout value in seconds for the SSH connection.
SSH Tunnel Start Port	Start assigning SSH tunnel ports at the specified port number.
Advanced Search Timeout	Enter the timeout value in seconds before advanced search returns current results.
Email Notification Delay	Enter the number of seconds to wait after initiating a transfer before producing notification emails.
Total Bandwidth Graph	Select this option to track total bandwidth usage across all notes on the Dashboard graph.
Advanced File Search	Select this option to allow users to search the entire database for filenames when using advanced search. **Note:** This may slow down Console if your database contains a large number of files.
Email Recipients	Select this option to allow email recipients to see each other's addresses.
Session notifications	Select this option to allow non-admins to access the session notifications page.
Smart Transfer Start Permissions	Select this option to allow users whose transfer path includes "Any" or addresses without a username to start any matching smart transfer that is shared and uses non-personal endpoints. For example, userA is authorized to use a transfer path that has one endpoint set to 10.0.123.45 and the other set to "Any". If userB's shared smart transfer is set up with non-personal endpoints on 10.0.123.45 (source) and 10.0.111.11 (destination), it will appear in userA's smart transfers list and can be started by userA.
Smart Transfer Sharing	Select this option to allow users to share smart transfers with personal logins.
Smart Transfer Editing	Select this option to allow administrators to edit each other's smart transfers.
Failover / Load balancing Behavior	Select **Failover + Load balancing** for Console to use the least busy node(s) first. For more information, see "Configure Failover Groups" on page 61.
Watchfolders	Enable the watchfolder feature in Console.
Watchfolders per page	Configure the number of watch folders to display per page when browsing configured watch folders.

Transfer Defaults

Item	Description
Target Rate	Set the default target rate.
Minimum Rate	Set the minimum rate.
Bandwidth Policy	Set the default transfer policy (choose among low, high, fair, and fixed).
Max. Retry Attempts	Set the maximum retry attempts.
Retry Interval	Set the retry interval in seconds.

Item	Description
Transport Encryption	Select between not-encrypted or aes-128 encryption.
File Compare Type	Select a file comparison type to verify transferred files.
File Overwrite Policy	Select an overwrite policy.

Report Generation

Item	Description
Retention Period	The number of days to keep generated reports before deleting them automatically.
Maximum Email Attachment Size	The maximum size in megabytes of CSV/XLSX files that may be sent by email. (Generated files can still be downloaded from the Reports page.)
File Maximum Data Length	The maximum size in megabytes of the result table for which CSV/XLSX files may be generated. (CSV/XLSX files are not generated if the result table is larger than this.) This setting is useful for preventing Console from trying to convert a giant data set into a file and running out of disk space.
Maximum XLS file rows	The maximum number of rows allowed for generated XLS files.

Security

Item	Description
Session Timeout	Sessions will timeout after the specified number of minutes of inactivity.
Deactivate Users	Deactivate a Console user if there has been "X" failed login attempts within "X" minutes.
Prevent concurrent login	If this checkbox is enabled, users can only be logged in from one client at a time.
Suppress logging of transfer tokens	Select this option to suppress tokens from being written to the database. Existing tokens already in the database are unaffected. **Note:** After enabling this feature, you may experience some lag before the setting takes effect if a request is already in progress and the node is taking a long time to reply.

Console Password Options

Item	Description
Password Expiration	Select this option to expire number of days
Password Duration	Enter the number of days before passwords expire. Setting the value to 0 will disable this feature.
Password Reuse Limit	Enter the number of passwords users need to go through before they can reuse an old password. Setting the value to 0 disables this feature.
Password Requirement Regular Expression	Enter a regular expression to specify password requirements. Leave blank to set no requirements.

Item	Description
	Note: You can select the **Restore Default** link to reset the password requirement to the following: "Passwords must be at least six characters long, with at least one letter, one number, and one symbol."
Password Requirement Message	Set a message describing the password requirements for users setting a new password.

Empty sessions (successfully completed with 0 bytes transferred)

A zero-byte transfer occurs when an Aspera product initiates a transfer but finds that the file already exists at the destination and that there are no changes between the source file and the file at the destination. If Console is monitoring the transfer, it marks the transfer as a success and records that zero bytes were transferred.

Nodes using hot folders regularly make these successful, zero-byte transfer sessions, and Console logs each one in its database. One workaround is to use Aspera Watch Folders instead of hot folders. Watch Folders only attempts to transfer a file when there is a change between the source and destination. For requirements and information on Watch Folders, see "Creating a Push Watch Folder" on page 63.

You can also configure Console to mark zero-byte transfers for deletion by the `DatabaseIngest` background job.

Note: Console might still display the most recent zero-byte sessions on heavy loaded systems if `DatabaseIngest` job backlog is high. Once the backlog is processed, Console will not display zero-byte transfers on the dashboard.

Item	Description
Leave in database	Leave all zero-byte transfers in the database.
Delete if hot folder	Mark for deletion zero-byte transfers initiated from hot folder sessions, leaving all other zero-byte transfers in the database.
Delete all	Mark for deletion all zero-byte sessions regardless of origin.

Understanding Space Watcher

Space watcher is a background process that checks the amount of free space in the database and gives warning when space is running low.

Space Watcher Functionality

Once a minute, space watcher runs a **`ls`** or **`dir`** command, then writes the free space in bytes to a table named **aspera_db_disk_space_free**. The exact command it executes is:

```
dir /-C "aspera_console_db_directory_path"
```

It only writes one record, always with "id=1". The **aspera_db_disk_space_free** table will never have more than one record in it. This table only has three fields:

Field	Value
id	Always equal to **1**.
bytes_free	BIGINT, max value = 9223372036854775807, which is approximately 8191 petabytes
last_reported_at	The time space watcher last stored an entry in the table.

If the process fails to figure out free space for any reason or fails to connect to MySQL, it does nothing and logs nothing. Successful or not, it then closes its connection and then sleeps for a minute before repeating the process.

Space Watcher Messages in Console

Unless warnings have been disabled (by setting the warning threshold to zero), Console checks the **aspera_db_disk_space_free** table when rendering a page. If it sees that there are no records in the table, or that it has been longer than 10 minutes since space watcher last reported, Console displays the following message at the top of the page: **"WARNING: No recent data from database free space watcher"**. If the last entry is recent (within 10 minutes) but the number of free bytes is less than the configured warning level (default: 10 gigabytes), it shows a message such as the following: **"WARNING: Database free space low (7.5 GB remaining)"**.

Working with Tags

Tags in Aspera products are JSON (JavaScript Object Notation) strings. Console uses tags to identify transfers and to label Console-initiated transfers. You can find a specific transfer's tags by navigating to a transfer's Session Details page and selecting the Session ID link under the Session State column.

Tags are used in the following tasks:

- Creating simple transfers.
- Creating and starting smart transfers.
- Creating advanced rulesets to filter by tags.
- Creating custom fields with rules involving tags.
- Searching using the Advanced Search.

The JSON Match Comparison Operator

Console includes a JSON match operator in the Custom Fields and Advanced Rulesets features, which provide a simple syntax for matching JSON formatted tags included in Aspera transfers. Below are examples of transfer tags in Console and Faspex transfers and instructions for matching them using the JSON match operator.

Console Transfers

A Console transfer is defined as any transfer initiated by Console using simple or smart transfers. Tags can be specified in both simple and smart transfers. A Console transfer tag is formatted in the following way:

```
{"aspera":
  {"console":
    {"user_specified"
       {"key1":"val1", … , "key3":"val3"}
    }
  }
}
```

An example of a corresponding JSON match value is shown below:

```
[aspera][console][user_specified][key1]val1
```

Faspex Transfer

A Faspex Transfer is any transfer initiated by Faspex. A Faspex transfer tag is formatted in the following way:

```
{"aspera":
  {"faspex":
    { "key1":"val1", … , "key3":"val3"}
    }
  }
```

The corresponding JSON match value is shown below:

```
[aspera][faspex][key1]val1
```

Note: It is recommended to use the Faspex Metadata filter for Faspex transfers instead. See "Basic Report Example: Faspex Metadata" on page 179 for more information on Faspex Metadata.

Regular Expressions in JSON Matches

You can also use regular expressions in a JSON match. Define the regular expression using forward slashes (/) like in the example below:

```
[aspera][console][user_specified]/+./
```

Important: Aspera advises against using regular expressions in keys, because the result will be the first value that matches the regular expression. In the example below, Console will return the first Faspex transfer it hits without backtracking to check for other transfers that meet the requirements.

```
[aspera][faspex][/+./]/.+/
```

Configure Background Processes

The Background Processes configuration page displays all Console processes and allows you to perform the following tasks:

- View a process log
- Edit a process
- Stop a process (although this is not available for all processes)
- Restart a process (although this is not available for all processes)

To access the Background Processes page, select **Configuration** > **Background** from the Console menu.

The following background processes can be accessed from the table:

- Controller
- Mongrel Manager
- Database Ingest
- Session Data Collector
- Node Info Collector
- File Data Collector
- Data Canonicalizer
- Custom Field
- Database Utility
- Transfer Initiator
- Email
- Report

To modify the settings for a given process, click the **edit** link in the corresponding table row. After clicking **edit**, the *Editing Background Process* page appears, along with the following options:

Options	Description
Startup type	Select the way that the background process starts (that is, manually or automatically) or disable the process from starting altogether.

Options	Description
Log level	Select the preferred level of logging for the log file output to control the verbosity of the log file output). Choose debug, info, warn, error or fatal.
Batch Size	*(Not available for all processes)* Input the number of rows to process each work interval.
Daily restart time (HH:MM)	*(Not available for all processes)* Input the time of day to restart the process in 24-hour time, UTC. Leave blank for no auto-restart.
Sleep Interval	Input the sleep interval time in seconds.
Maximum Startup Interval	Input a time (in seconds) that must elapse before the given process is assumed to be hanging.
Maximum Heartbeat Interval	Input a time (in seconds) that must elapse between heartbeats before the given process is assumed to be hanging.

Configure the Apache HTTP Server

You may configure Console's Apache HTTP Server to use a different host name, communication port, and namespace using asctl commands.

Change the Number of Mongrel servers

By default, Console opens four mongrel servers. To change it, for example, from the default (4) to 10, use the following command:

```
> asctl console:mongrel_count 10
```

Update the Hostname

During the installation, you should have configured the Console's hostname. Use this command to print the current hostname:

```
> asctl apache:hostname
```

To change the hostname, use the following command. Replace **HOSTNAME** with the new hostname:

```
> asctl apache:hostname HOSTNAME
```

Important: When changing the hostname, the server's SSL certificate should be regenerated. Select (y) when prompted to generate a new SSL certificate.

When the hostname is updated, advise your clients of the new URL. In this example, use the following address:

`http://HOSTNAME/aspera/console`

Change HTTP and HTTPS ports

By default, Console's web servers are running on TCP/80 (HTTP) and TCP/443 (HTTPS). Use the following commands to update these ports (where, in this example, we TCP/7080 for HTTP and TCP/7443 for HTTPS):

Item	Command
HTTP	`> asctl apache:http_port 7080`
HTTPS	`> asctl apache:https_port 7443`

Change Console namespace

Console uses the namespace */aspera/console* by default. Use this command to print the current namespace:

```
> asctl console:uri_namespace
```

To set the namespace to, for example, **/console**, use the following command:

```
> asctl console:uri_namespace /console
```

When the namespace is updated, advise your client of the new URL. For example, if your Console server's address is *10.0.0.10*, use this URL:

`https://10.0.0.10/console`

Note: Refer to "asctl Command Reference" on page 109 for a complete asctl command reference.

asctl Command Reference

You can use **asctl** commands in a Command window to display or modify IBM Aspera Console's component settings. Console configuration options that can be modified using **asctl** are listed below. If there are modifications that cannot be accomplished with **asctl**, notify Aspera Support.

Important: You must be an admin to run **asctl**. Right click the Command window and select **Run as administrator**.

Component	Description
Apache	Apache web server.
Console	Console main application.
MySQL	MySQL database.

All components commands

Important: The commands in this section control all Console components.

Task	Command	Description
Show config info	asctl all:info	Print info about all components.
Restart all components	asctl all:restart	Restart all components.
Setup status	asctl all:setup_status	Information about configuring all components.
Start	asctl all:start	Start all components.
Show status	asctl all:status	Display the status of each component.
Stop	asctl all:stop	Stop all components.
Show version	asctl all:version	Display the current version of each component.

Apache

Task	Command	Additional Information
Create a setup file	asctl apache:create_setup_file *file*	Create a reusable file that contains answers to the setup

Task	Command	Additional Information
		questions. Replace *file* with a file name.
Disable Apache	asctl apache:disable	Disable the Aspera Apache server. When disabled, the service will not start when rebooting computer, does not print reminders or update its configurations.
Disable Apache logs	asctl apache:disable_logs	Set the Apache's log level to 'emerg'.
Enable Apache logs	asctl apache:enable_logs	Set the Apache's log level to 'notice'.
Re-generate conf	asctl apache:generate_config	Generate the component's configuration file using the current settings.
Display hostname	asctl apache:hostname	Display the hostname or IP address of the server.
Change hostname	asctl apache:hostname *host*	Change the hostname or IP address of the server. Replace *host* with a new hostname or IP address.
Display HTTP port	asctl apache:http_port	Display the HTTP port the web server listens to.
Change HTTP port	asctl apache:http_port *port*	Change the HTTP port the web server listens to. Replace *port* with a new port number.
Display HTTPS port	asctl apache:https_port	Display the HTTPS port the web server listens to.
Change HTTPS port	asctl apache:https_port *port*	Change the HTTPS port the web server listens to. Replace *port* with a new port number.
Show config info	asctl apache:info	Print configuration info about Apache.
Copy your SSL files into the Aspera default location (under default names)	asctl apache:install_ssl_cert *cert_file key_file [chain_file]*	After upgrading Faspex and Common, use this command to copy your original SSL certificate, key and optional chain file to `/opt/aspera/common/apache/conf` and give them Aspera-standard names. The `httpd-ssl.conf` file is also re-rendered and permissions/ownership is set for the cert files.
Set Apache log level	asctl apache:log_level *option*	Specify the Apache's log level. Replace *option* with **crit**, **error**, **warn**, **notice**, **info** or **debug**.

Task	Command	Additional Information
Create SSL certificate	asctl apache:make_ssl_cert *hostname*	Create a self-signed SSL certificate for the specified hostname. Replace *hostname* with your hostname.
Restart Apache	asctl apache:restart	
Configure Apache	asctl apache:setup	
Configure Apache using saved file	asctl apache:setup_from_file *filename*	Run setup using the answers from a file created using the "create_setup_file" command.
Start Apache	asctl apache:start	
Show Apache status	asctl apache:status	
Stop Apache	asctl apache:stop	
Upgrade Apache	asctl apache:upgrade	
Show Apache's version	asctl apache:version	

Console

Task	Command	Description
Create or update admin	asctl console:admin_user *login email [password]*	Create a new admin, or update an existing admin account. Replace *login* with a login, *email* with its email. You can add the account's password in the command (*[password]*), or enter it when prompted. If the login you have entered exists, the account is updated with new email and password.
Backup database	asctl console:backup_database *dir*	Backup Console database and associate files to the specified directory. Replace *dir* with a path to store the backup.
Display base port	asctl console:base_port	Display the base port of the mongrels.
Change base port	asctl console:base_port *[arg]*	Change the base port of the mongrels. Replace *[arg]* with the new base port number.
Create setup file	asctl console:create_setup_file *file*	Create a reusable file that contains answers to the setup questions. Replace *file* with a file name.
Disable Console	asctl console:disable	Disable Console. When disabled, the service will not start when rebooting computer, does not print reminders or update its configurations.

Task	**Command**	**Description**
Re-generate conf	asctl console:generate_config	Generate Console component's configuration file using the current settings.
Config info	asctl console:info	Print Console configuration info.
Update database	asctl console:migrate_database	Update database to the latest schema.
Display mongrel count	asctl console:mongrel_count	Display the number of mongrels to spawn.
Change mongrel count	asctl console:mongrel_count *arg*	Change the number of mongrels to spawn. Replace *arg* with a number.
Rake command	asctl console:rake *arg*	Evoke a rake command.
Restart Console	asctl console:restart	Restart mongrel web servers and all background processes.
Restore config and data	asctl console:restore *dir*	Restore Console database and configuration from a backup directory.
Restore database	asctl console:restore_database *dir*	Restore Console database from a backup directory.
Configure Console component	asctl console:setup	Configure this component.
Configure Console using saved file	asctl console:setup_from_file *file*	Run setup using the answers from a file created using the "create_setup_file" command.
Start Console	asctl console:start	Starts mongrel web servers and all background processes.
Show Console status	asctl console:status	Display Console status.
Stop Console	asctl console:stop	Stops mongrel web servers and all background processes.
Upgrade	asctl console:upgrade	Upgrade Console from a previous version.
Display namespace	asctl console:uri_namespace	Display Console's URL namespace.
Change namespace	asctl console:uri_namespace *arg*	Change Console's URL namespace. Replace *arg* with the new namespace.
Show Console's version	asctl console:version	Display the currently set up version.
Generate email templates	asctl console:generate_email_templates	Recreate email template files.

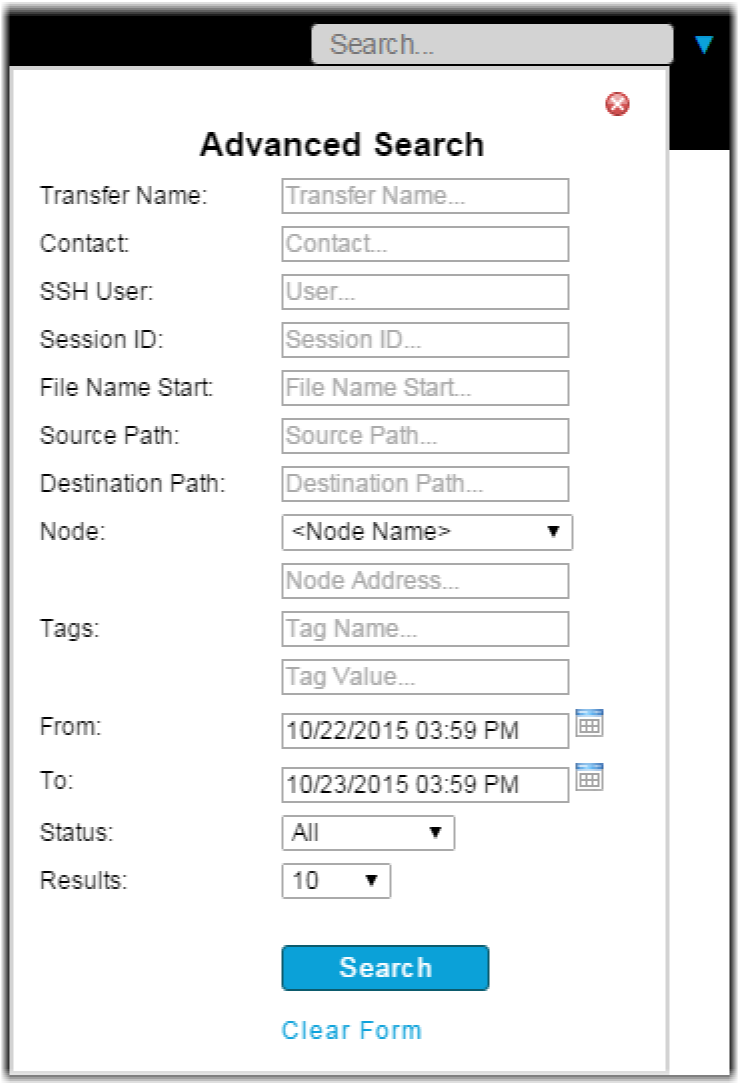
MySQL

Task	Command	Description
Create setup file	asctl mysql:create_setup_file *file*	Create a reusable file that contains answers to the setup questions. Replace *file* with a file name.
Display database directory	asctl mysql:data_dir	Display the directory that the databases are kept in.
Disable MySQL	asctl mysql:disable	Disable the Aspera MySQL. When disabled, the service will not start when rebooting computer, does not print reminders or update its configurations.
Grant access on MySQL-only server	asctl mysql:grant_remote_access *host mysql_user password*	If MySQL server is running on a different computer, use this command on the MySQL machine to allow access from the specified machine. Replace *host*, *mysql_user* and *mysql_password* with the server's hostname, MySQL's user name, and the user's password, respectively.
Show config info	asctl mysql:info	Print configuration info about MySQL.
Show port	asctl mysql:port	Display the port the MySQL server listens to.
Change port	asctl mysql:port *port*	Change the port the MySQL server listens to. Replace *port* with a new port number.
Restart MySQL	asctl mysql:restart	Restart the Aspera MySQL.
Set root password	asctl mysql:set_root_password	Set the password for 'root' in MySQL.
Configure MySQL-only server	asctl mysql:setup	If MySQL server is running on a different computer, use this command on the MySQL machine to configure it.
Configure MySQL using saved file	asctl mysql:setup_from_file *file*	Run setup using the answers from a file created using the "create_setup_file" command.
Start MySQL	asctl mysql:start	Start the Aspera MySQL.
Show MySQL status	asctl mysql:status	Display the Aspera MySQL status.
Stop MySQL	asctl mysql:stop	Stop the Aspera MySQL.
Upgrade MySQL-only server	asctl mysql:upgrade	If MySQL server is running on a different computer, use this command on the MySQL machine to upgrade the database.

Task	Command	Description
Show MySQL's version	asctl mysql:version	Display the currently set up version.

Advanced Search

You can search for a transfer from any page in Console by using the search bar in the top right corner of the page. If you want to refine your search, you can access the Advanced Search dialog by selecting the blue drop-down arrow next to the search bar.


Advanced Search

Transfer Name:	Transfer Name...
Contact:	Contact...
SSH User:	User...
Session ID:	Session ID...
File Name Start:	File Name Start...
Source Path:	Source Path...
Destination Path:	Destination Path...
Node:	<Node Name> ▼
	Node Address...
Tags:	Tag Name...
	Tag Value...
From:	10/22/2015 03:59 PM
To:	10/23/2015 03:59 PM
Status:	All ▼
Results:	10 ▼

Filter	Description
Transfer Name	Include transfers with this name
Contact	Include transfers initiated by this user.
SSH User	Include transfers involving this SSH user.
Session ID	Include transfers with this unique session ID

Filter	Description
File Name Start	Include transfers with files that start with this string.
Source Path	Include transfers with files that originated from this location.
Destination Path	Include transfers with files transferred to this location.
Node	Include transfers involving this selected node or this node IP address.
From	Include transfers started from this date and onwards.
To	Include transfers from this date and onwards.
Status	Include transfers with the current state designated: • Active • Completed • Cancelled • Error
Results	The number of results you want Console to display.

Setting Up the Console Environment

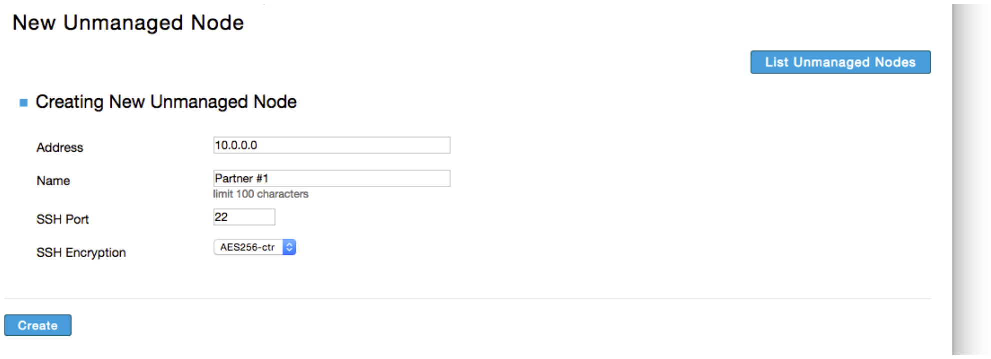
Setup Example #1: Monitoring Transfers with Another Organization

This example shows how to create a user group that can monitor all transfers between your organization and a partner organization. The configuration in this example uses the following values:

- Partner node address: 10.0.0.0
- Console user: partner-1-staff (non-admin)

1. Add the partner's node as an unmanaged node, and add an endpoint.

   To do so, go to **Nodes** from the Console menu, click **List Unmanaged Nodes** > **New Unmanaged Node**. Enter the partner node's information. You can add the partner's node as an unmanaged node without further configuration.



New Unmanaged Node

Creating New Unmanaged Node

Address: 10.0.0.0

Name: Partner #1

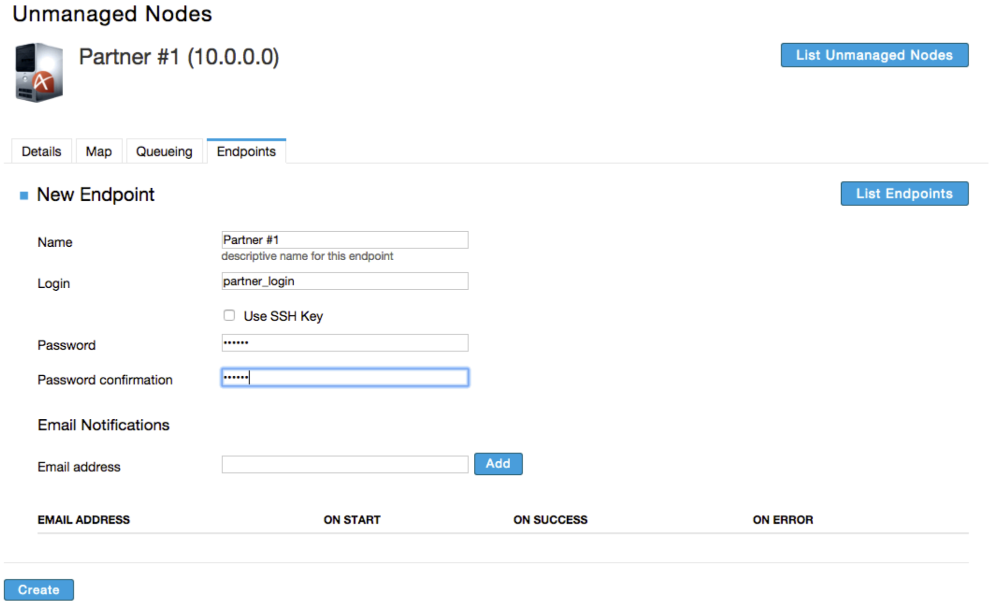
limit 100 characters

SSH Port: 22

SSH Encryption: AES256-ctr

2. Create a new endpoint with saved login information.

   The addition of a new node creates an endpoint in Console with a wildcard (***@10.0.0.0**). Wildcard endpoints require a user to enter login credentials every time the user uses it for a transfer. To create an endpoint with saved login credentials, go to the **Endpoints** tab and click **Add Endpoint**. Enter the login credentials for a user on the node and click **Create**.


Unmanaged Nodes

Partner #1 (10.0.0.0)

New Endpoint

Name	Partner #1 descriptive name for this endpoint
Login	partner_login
	□ Use SSH Key
Password	••••••
Password confirmation	••••••

Email Notifications

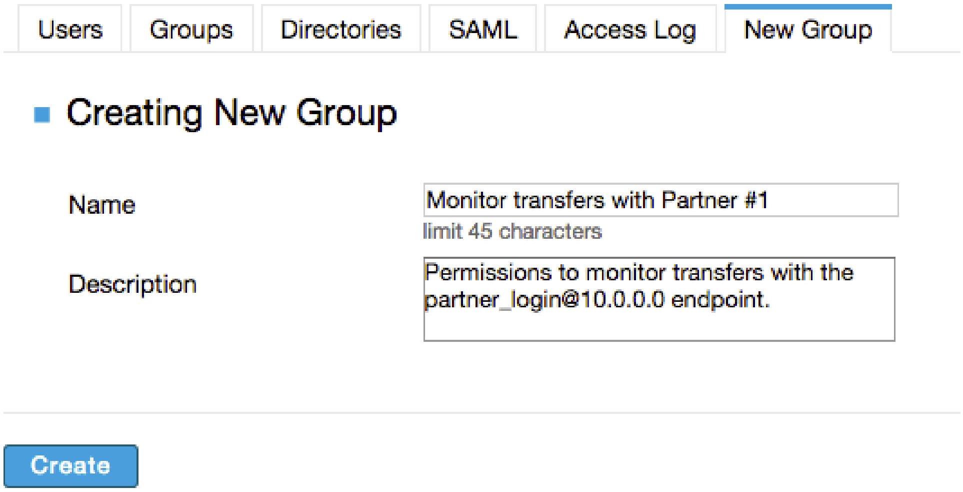
Email address	

EMAIL ADDRESS	ON START	ON SUCCESS	ON ERROR

For more information on the actions listed in this step, see “Adding Unmanaged Nodes” on page 28.

3. Create a group with permission to monitors transfers with the partner's node.

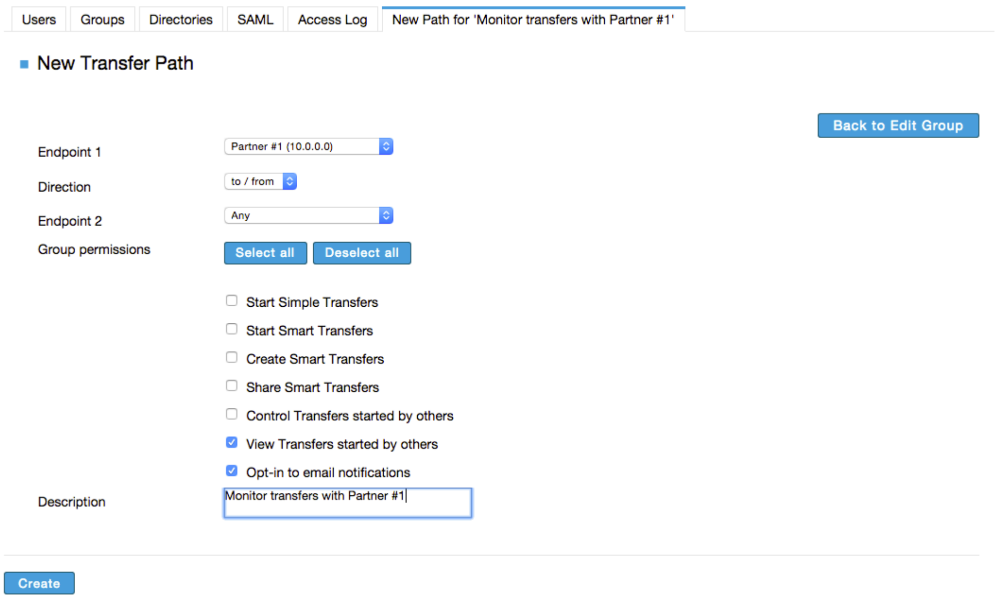
   Go to **Groups** from the Console menu and click **New Group**. In the Create New Group page, enter the group's name and description. Click **Create** when finished.



Creating New Group

Name	Monitor transfers with Partner #1 limit 45 characters
Description	Permissions to monitor transfers with the partner_login@10.0.0.0 endpoint.

When the group is created, select **Transfer Path** > **Add Path**. The following is an example of the Transfer Path settings:



New Transfer Path

Endpoint 1: Partner #1 (10.0.0.0)

Direction: to / from

Endpoint 2: Any

Group permissions:

- ☐ Start Simple Transfers
- ☐ Start Smart Transfers
- ☐ Create Smart Transfers
- ☐ Share Smart Transfers
- ☐ Control Transfers started by others
- ☑ View Transfers started by others
- ☑ Opt-in to email notifications

Description: Monitor transfers with Partner #1

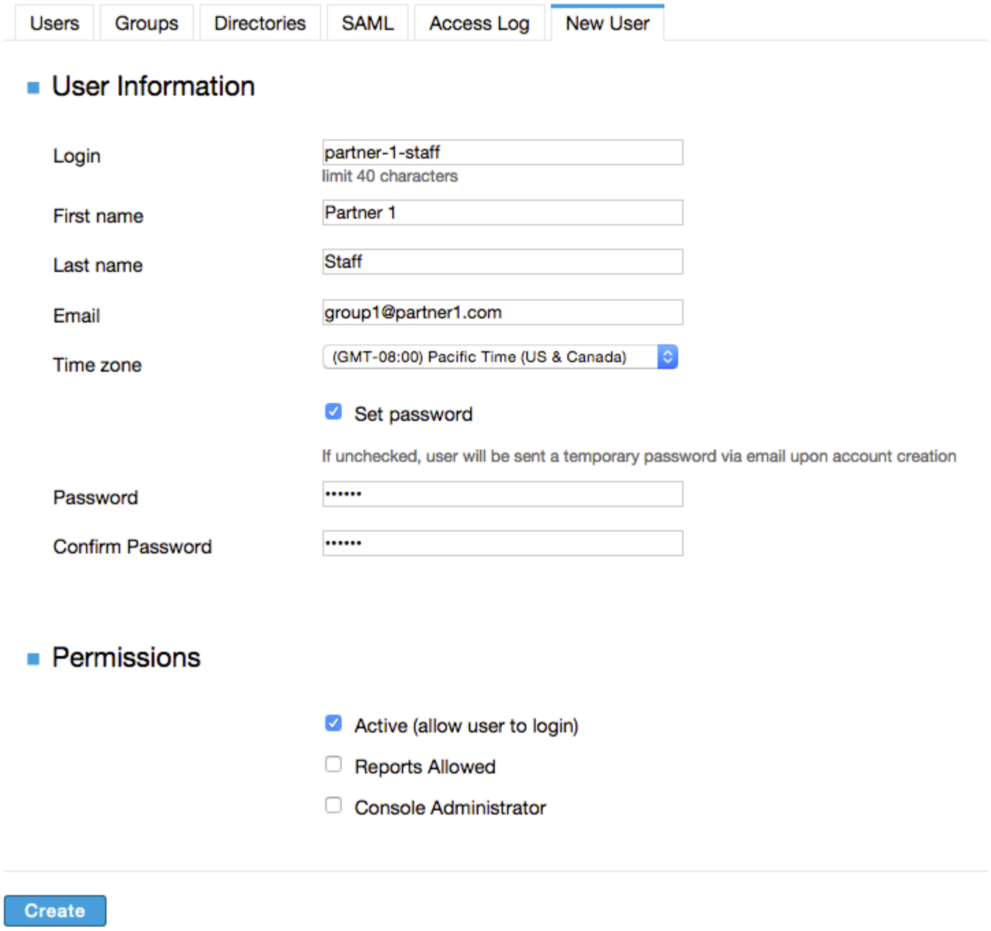
Field	Description
Endpoint1	Choose the Partner's node or endpoint.
Direction	Set **to / from** for inbound and outbound transfers.
Endpoint2	Choose **Any** so that users can make transfers with any node.
Group permissions	Check these options: • View transfers started by others • Opt-in to email notifications

Click **Create**.

For more information on the tasks listed in this step, see “Creating Console Groups” on page 37.

4. Create a Console user and add it to the group.

   To create a user account to monitor the transfers, go to **Users** from the Console menu and click **New User**. Fill out the form and click **Create**.



User Information

Login: partner-1-staff
limit 40 characters

First name: Partner 1

Last name: Staff

Email: group1@partner1.com

Time zone: (GMT-08:00) Pacific Time (US & Canada)

☑ Set password

If unchecked, user will be sent a temporary password via email upon account creation

Password: ••••••

Confirm Password: ••••••

Permissions

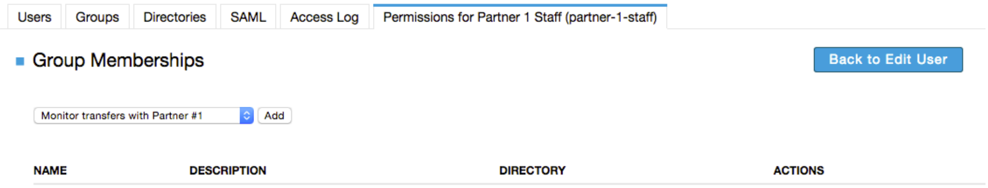
☑ Active (allow user to login)

☐ Reports Allowed

☐ Console Administrator

5. Assign the user to the group.

   In the User Maintenance page's **Groups** tab, select the group from the drop menu and click **Add**.



Group Memberships

Monitor transfers with Partner #1

NAME	DESCRIPTION	DIRECTORY	ACTIONS

   When logging into Console with this user account, you can monitor all the transfers with the partner's node.

   For more information on the tasks listed in this step, see "Creating Console Users" on page 39.

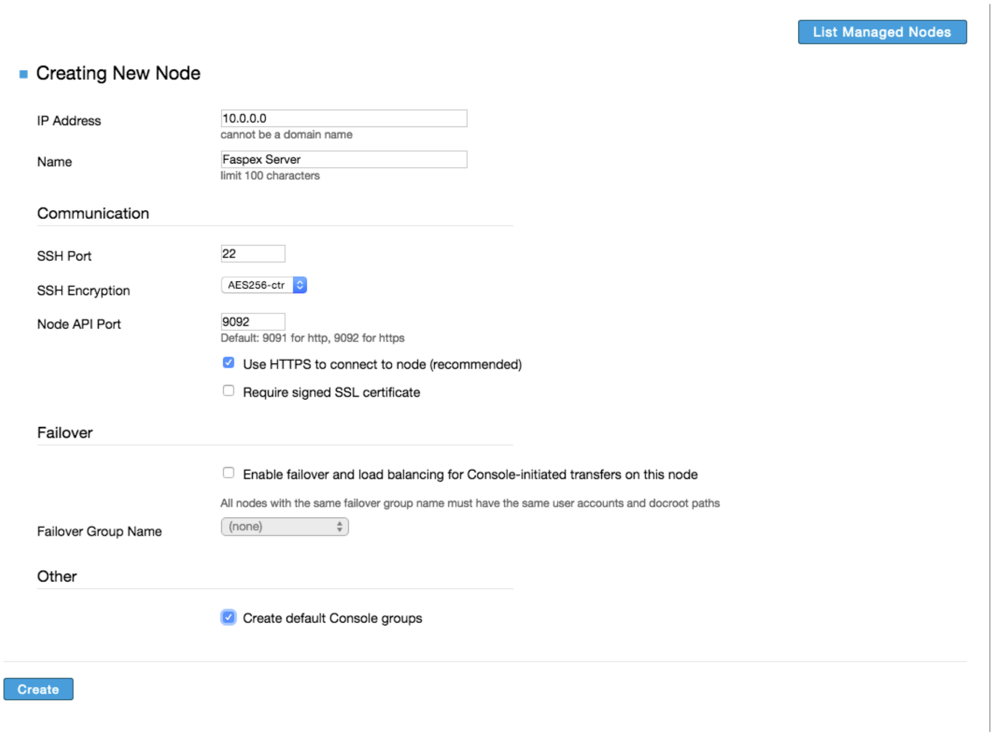
Setup Example #2: Managing Aspera Faspex Transfers

This example shows how to monitor and control transfers on a node running Faspex, a file exchange application built upon IBM Aspera High-Speed Transfer Server for a centralized transfer solution. The configuration in this example is as follow:

- **Faspex node address:** 10.0.0.0
- **Console user:** faspex-monitor-1

1. Add Faspex as a managed node, and create an endpoint.

   Go to **Nodes** from the Console menu and click **New Managed Node**. Enter the Faspex node information and click **Create**.



Creating New Node

IP Address: 10.0.0.0
cannot be a domain name

Name: Faspex Server
limit 100 characters

Communication

SSH Port: 22

SSH Encryption: AES256-ctr

Node API Port: 9092
Default: 9091 for http, 9092 for https

☑ Use HTTPS to connect to node (recommended)

□ Require signed SSL certificate

Failover

□ Enable failover and load balancing for Console-initiated transfers on this node

All nodes with the same failover group name must have the same user accounts and docroot paths

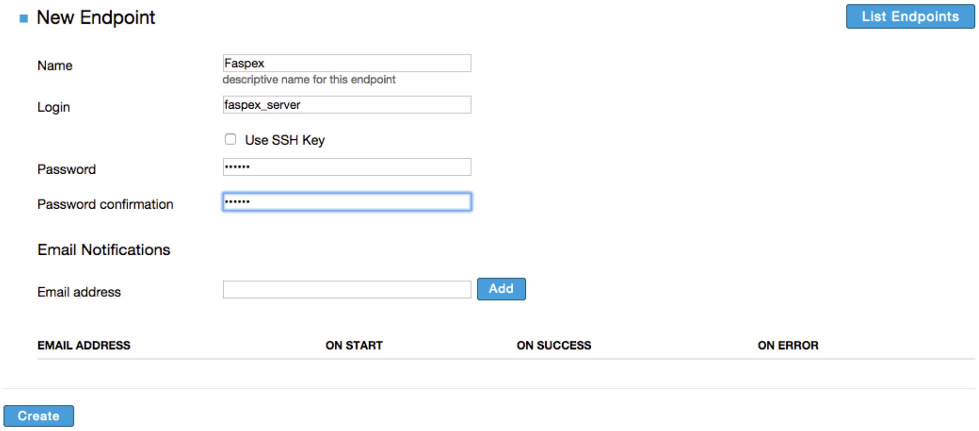
Failover Group Name: (none)

Other

☑ Create default Console groups

Important: If you wish to configure the Faspex node transfer settings using Console, go to the Node Maintenance page, select the **Credentials** tab, and enter the SSH login.

When the node is added, go to the **Endpoints** tab and click **Add Endpoint**. Enter **Faspex** in the Login field, leave password fields blank. When finished, click **Create**.



New Endpoint

Name: Faspex
descriptive name for this endpoint

Login: faspex_server

□ Use SSH Key

Password: ••••••

Password confirmation: ••••••

Email Notifications

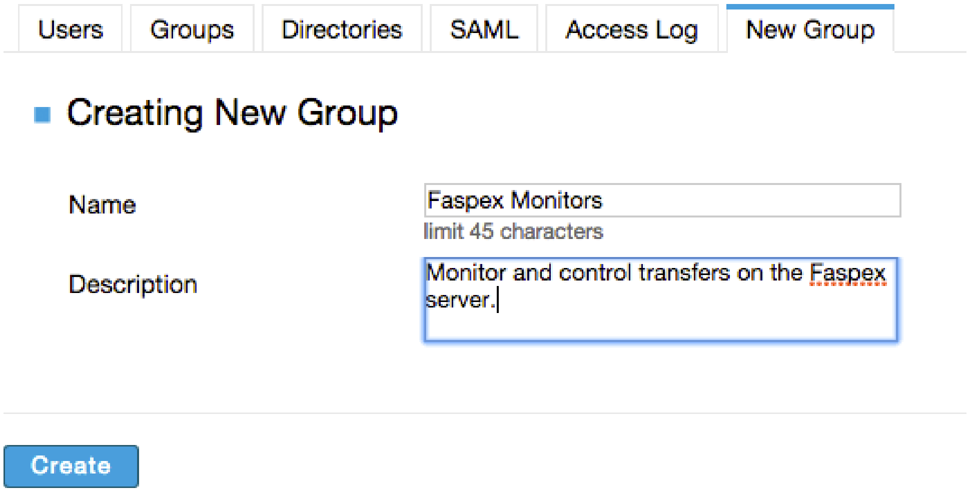
Email address:

EMAIL ADDRESS	ON START	ON SUCCESS	ON ERROR

For more information on the tasks listed in this step, see xfer_user/project1svcAspera.

2. Create a group with the proper permissions.

   Go to **Groups** from the Console menu and click **New Group**. Enter the group's information and click **Create**.



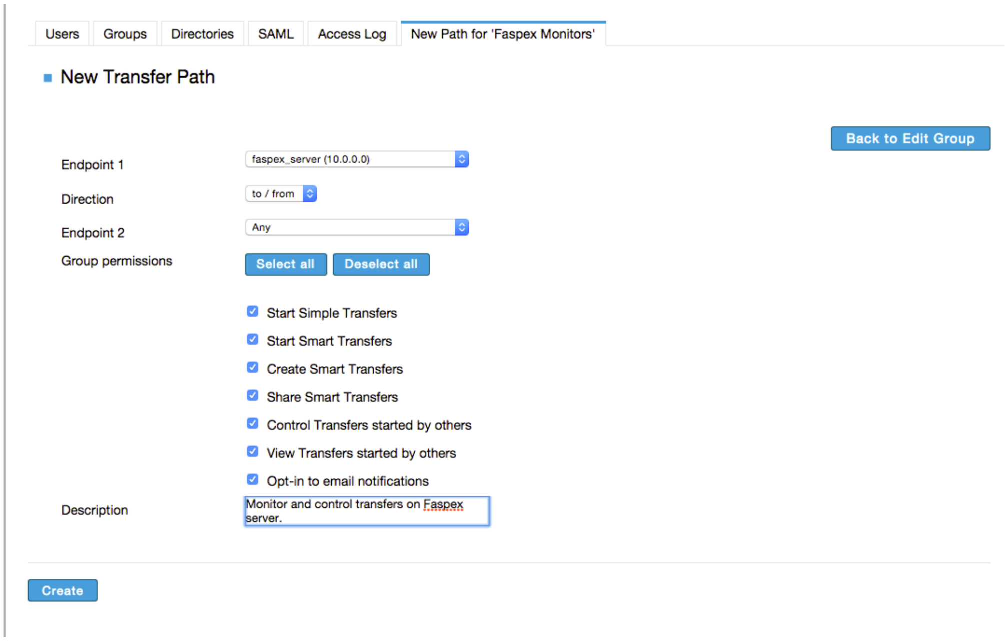
Creating New Group

Name: Faspex Monitors

limit 45 characters

Description: Monitor and control transfers on the Faspex server.

When the group is created in the *Group Maintenance* page, go to the **Transfer Paths** tab and click **Add Path**. Use the following settings for this Transfer Path (change the Faspex node address to match yours):



New Transfer Path

Endpoint 1: faspex_server (10.0.0.0)

Direction: to / from

Endpoint 2: Any

Group permissions:

- ☑ Start Simple Transfers
- ☑ Start Smart Transfers
- ☑ Create Smart Transfers
- ☑ Share Smart Transfers
- ☑ Control Transfers started by others
- ☑ View Transfers started by others
- ☑ Opt-in to email notifications

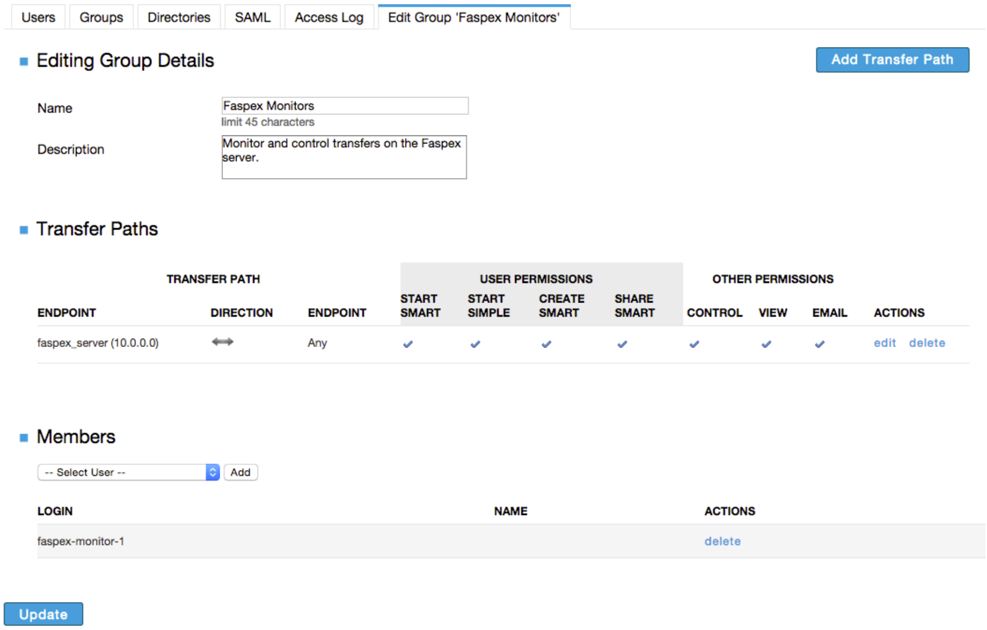
Description: Monitor and control transfers on Faspex server.

Item	Description
Endpoint1	faspex@22.33.44.55
Direction	to / from
Endpoint2	Any
Options	Check these options: • View Transfers started by others • Opt-in to email notifications

For more information on the tasks listed in this step, see "Creating Console Groups" on page 37.

3. Add users to the group.

   In the *Group Maintenance* page, go to the Members tab, select the Console user from the menu and click **Add**. When added, the Console user can monitor and control transfers on the Faspex node through Console.



Editing Group Details

Name: Faspex Monitors
limit 45 characters

Description: Monitor and control transfers on the Faspex server.

Transfer Paths

TRANSFER PATH			USER PERMISSIONS				OTHER PERMISSIONS			
ENDPOINT	DIRECTION	ENDPOINT	START SMART	START SIMPLE	CREATE SMART	SHARE SMART	CONTROL	VIEW	EMAIL	ACTIONS
faspex_server (10.0.0.0)	⟷	Any	✓	✓	✓	✓	✓	✓	✓	edit delete

Members

LOGIN	NAME	ACTIONS
faspex-monitor-1		delete

   For more information on the tasks listed in this step, see "Creating Console Users" on page 39.

Setup Example #3: Create Groups of Different Permissions

This example shows how to assign different permissions to different project members. The configuration in this example is as follows:

- **Endpoint1**: project@1.2.3.4
- **Endpoint2**: project@5.6.7.8
- **User1 (Project admin)**: admin1
- **User2 (Project member)**: member1
- **Group - Project admin**: All permissions between **Endpoint1** and **Endpoint2**
- **Group - Project member**: Limited permissions between **Endpoint1** and **Endpoint2**

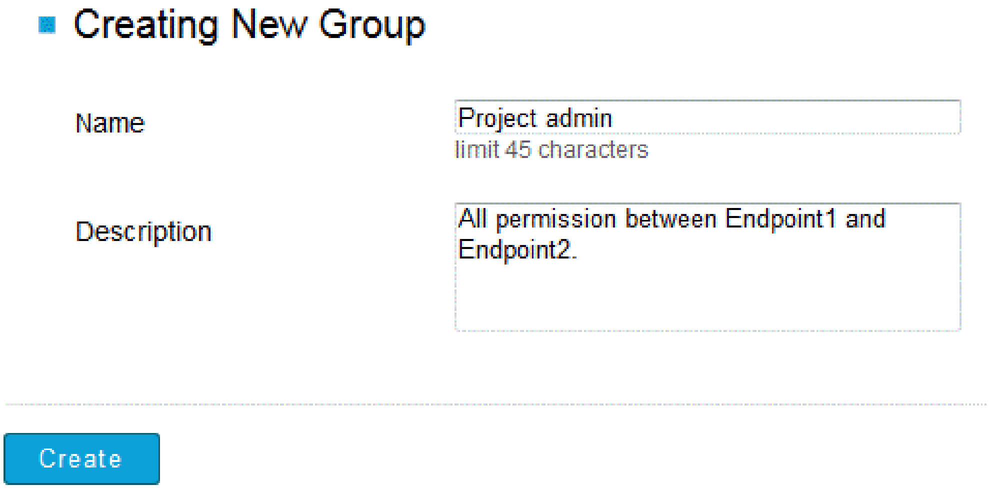
1. Prepare the nodes, endpoints and the Console user accounts.

   - **Add a managed or unmanaged node with an endpoint for transfer**: project@1.2.3.4
   - **Add a managed or unmanaged node with an endpoint for transfer**: project@5.6.7.8
   - **Add a Console user**: admin1
   - **Add a Console user**: member1

   For more information on the tasks listed in this step, see xfer_user/project1svcAspera, "Adding Unmanaged Nodes" on page 28, and "Creating Console Users" on page 39.

2. Create a group for the project administrator.

   Go to **Groups** from the Console menu and click **New Group**. Enter the group's information and click **Create**.



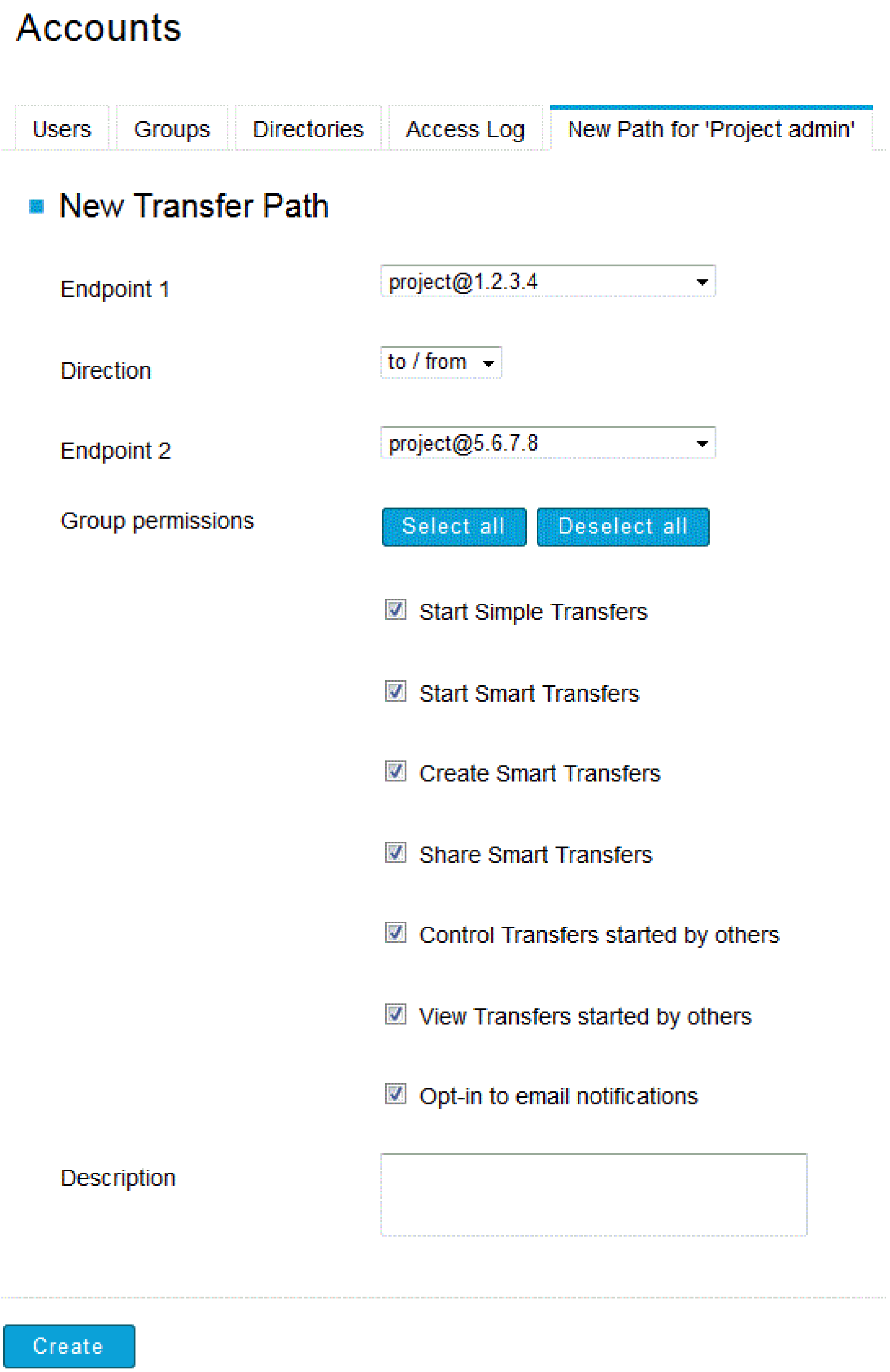
Creating New Group

Name: Project admin

limit 45 characters

Description: All permission between Endpoint1 and Endpoint2.

In the Group Maintenance page, click the **Transfer Paths** tab. Enter the following information:



Accounts

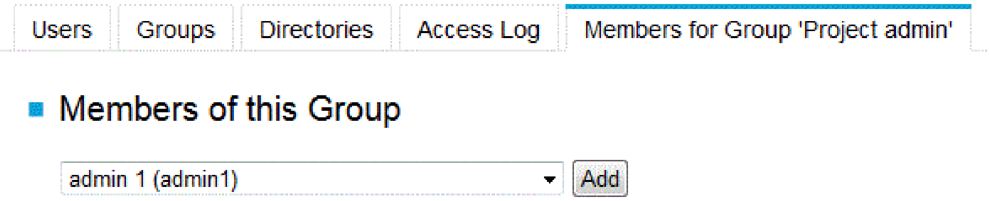
New Transfer Path

Endpoint 1	project@1.2.3.4
Direction	to / from
Endpoint 2	project@5.6.7.8
Group permissions	☑ Start Simple Transfers ☑ Start Smart Transfers ☑ Create Smart Transfers ☑ Share Smart Transfers ☑ Control Transfers started by others ☑ View Transfers started by others ☑ Opt-in to email notifications
Description	

Item	Description
Endpoint1	project@1.2.3.4
Direction	to / from
Endpoint2	project@5.6.7.8
Options	**Select All**.

Click **Create**.

In the Group Maintenance page, go to the **Members** tab and add the user **admin1** into this group.



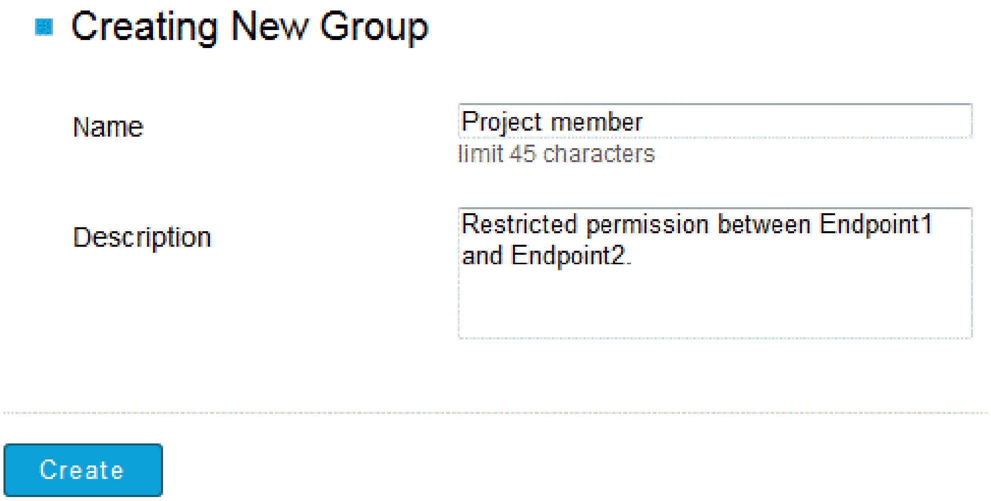
Members of this Group

admin 1 (admin1)

For more information on the tasks listed in this step, see "Creating Console Groups" on page 37.

3. Create a group for the project members.

Go to **Groups** from the Console menu and click **New Group**. Enter the group's information and click **Create**.



Creating New Group

Name: Project member
limit 45 characters

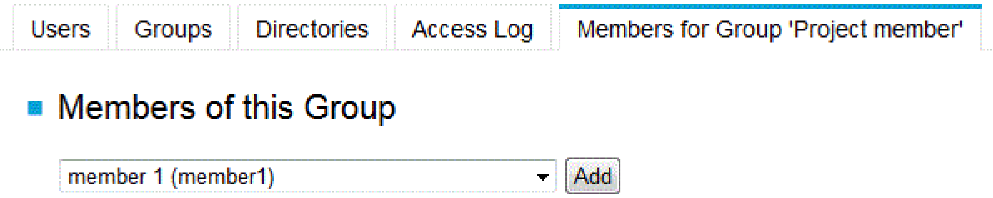
Description: Restricted permission between Endpoint1 and Endpoint2.

In the *Group Maintenance* page, click the **Transfer Paths** tab. Enter the following information:

Item	**Description**
Endpoint1	project@1.2.3.4
Direction	to / from
Endpoint2	project@5.6.7.8
Options	Check these options: • Start Smart Transfers • View Transfers started by others • Opt-in to email notifications

When finished, click **Create**.

In the Group Maintenance page, go to the **Members** tab and add the user **member1** into this group.



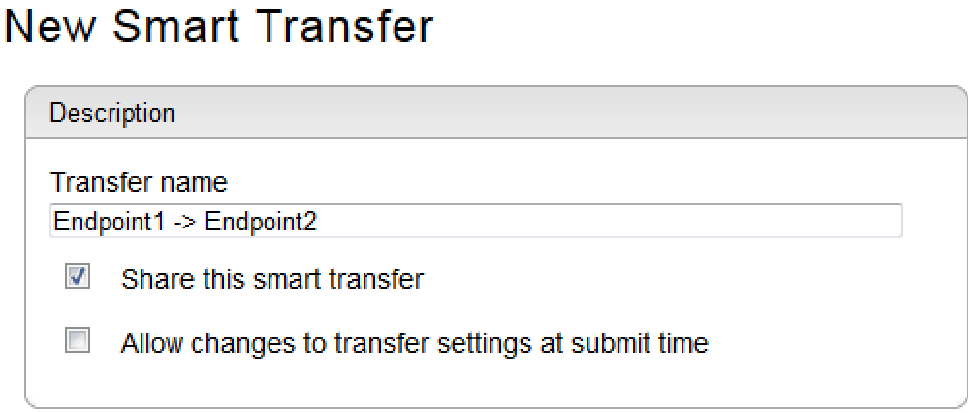
Members of this Group

member 1 (member1)

For more information on the tasks listed in this step, see "Creating Console Groups" on page 37.

4. Create a Smart Transfer template with the project admin account.

   At this point, both Console users should have the proper permissions. Use the project admin account (**admin1**) to create a Smart Transfer template.



New Smart Transfer

Description

Transfer name: Endpoint1 -> Endpoint2

☑ Share this smart transfer

□ Allow changes to transfer settings at submit time

   When the user **admin1** creates and share a Smart Transfer template, **member1** will be able to execute the Smart Transfer template.

   For more information on the tasks listed in this step, see "Sharing a Smart Transfer" on page 57

5. Execute the Smart Transfer with the project member account.

   The **Project member** group, which has the same Transfer Path as the **Project admin** group, has access to the shared Smart Transfer templates. Go to **Transfer** from the Console menu, the Project member will see the Smart Transfer template listed in the Saved Smart Transfers table.

Email Template Examples

Email Template Example: Creating a Simple Notification for a Successful Transfer

The following example shows how to create an email template that notifies a user of a successful transfer with minimal information.

1. Select **Create new transfer success email template** and then **edit**.
2. Name your template "Client Success Email".
3. Enter a **From Name** and **Reply-to Address** if you don't want the notification to come from the default email address.
4. Enter a new email subject: "Client Transfer Notification - Success".
5. Click **Edit Plain Template** and make remove variables to limit information provided to the recipient. For example:

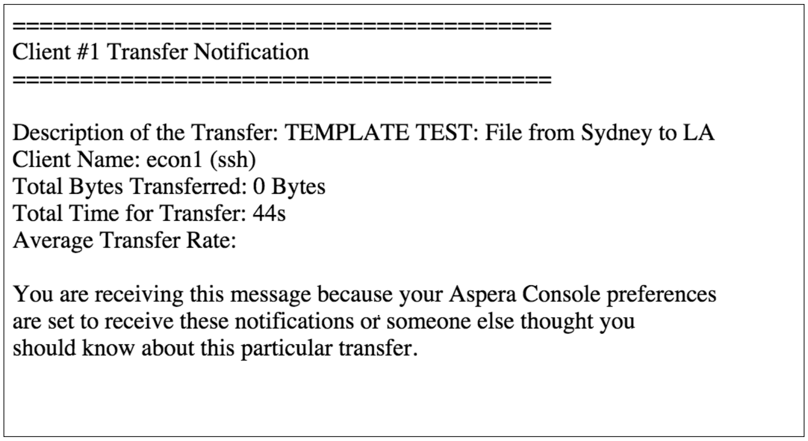
```
========================================
Client Transfer Notification
========================================

Description of the Transfer:        DESCRIPTION
```

```
Client Name:                        CONTACT
Total Bytes Transferred:            BYTES_TRANSFERRED
Total Time for Transfer:            ELAPSED_TIME
Average Transfer Rate:              AVERAGE_RATE

You are receiving this message because your Aspera Console preferences
are set to receive these notifications or someone else thought you
should know about this particular transfer.
```

The end result should look like the following:



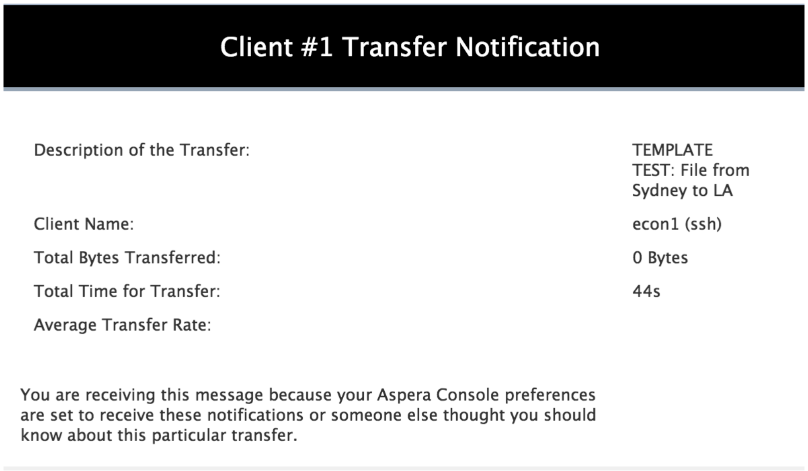
==
Client #1 Transfer Notification
==

Description of the Transfer: TEMPLATE TEST: File from Sydney to LA
Client Name: econ1 (ssh)
Total Bytes Transferred: 0 Bytes
Total Time for Transfer: 44s
Average Transfer Rate:

You are receiving this message because your Aspera Console preferences
are set to receive these notifications or someone else thought you
should know about this particular transfer.

6. **Edit HTML Template** to match the information in the basic template.

   The end result should look like the following:



Client #1 Transfer Notification

Description of the Transfer:	TEMPLATE TEST: File from Sydney to LA
Client Name:	econ1 (ssh)
Total Bytes Transferred:	0 Bytes
Total Time for Transfer:	44s
Average Transfer Rate:	

You are receiving this message because your Aspera Console preferences are set to receive these notifications or someone else thought you should know about this particular transfer.

7. Click the **Send Test Email** button to test the new email template.

Email Template Example: Adding Company Branding to Your Template

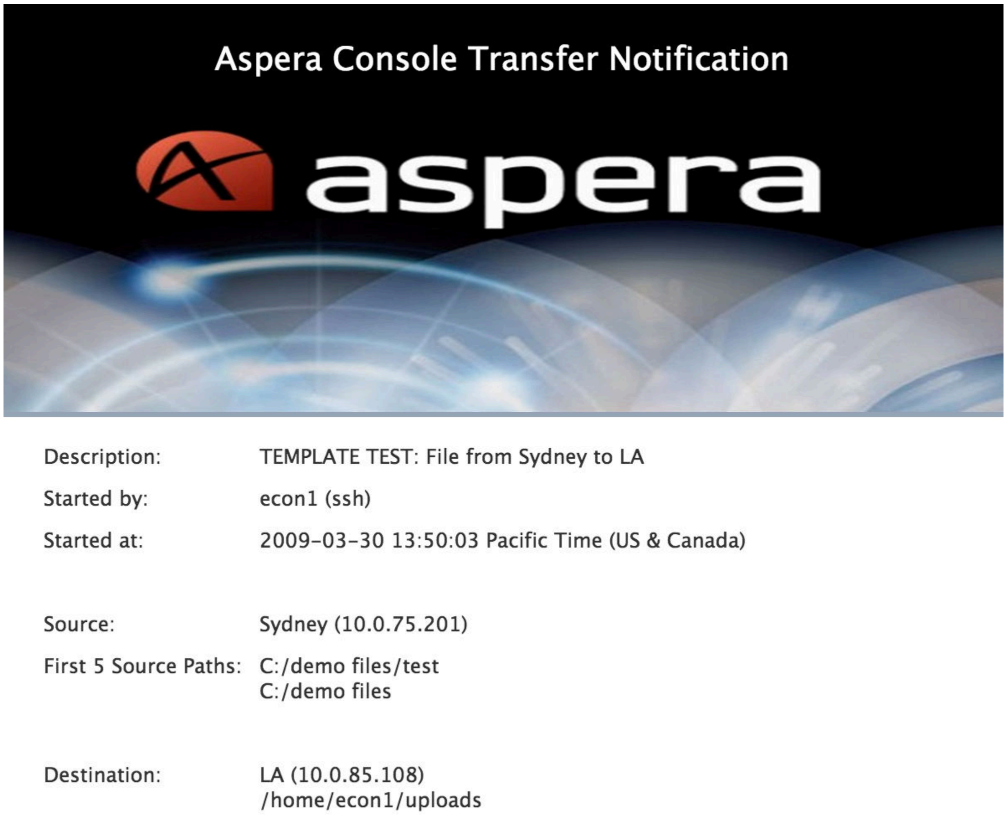
The following example shows how to create an email template that shows company branding when opened in HTML format.

1. On the Template preview screen, click the **Edit HTML Template** button to modify the template's HTML code.
2. Locate the URL of your company logo. Your image must be hosted on a server that is accessible to the recipient.
3. Open the HTML Template and insert the following code in the desired location.

   ```
   <IMG SRC="http://<IMAGE_URL>">
   ```

   In this example, we've inserted the logo into the header.

The result may look like the following:



Aspera Console Transfer Notification

Description:	TEMPLATE TEST: File from Sydney to LA
Started by:	econ1 (ssh)
Started at:	2009–03–30 13:50:03 Pacific Time (US & Canada)
Source:	Sydney (10.0.75.201)
First 5 Source Paths:	C:/demo files/test C:/demo files
Destination:	LA (10.0.85.108) /home/econ1/uploads

Node References

Node-Level Configuration Options

To start node configuration, go to **Nodes** in the Console menu. Click **edit** for an existing node that you wish to configure. The server's admin credentials are required for the configuration. See “Updating a Node's Admin Credentials” on page 28.

The node configuration options can be found in the Configuration tab. The following is a summarized chart for navigating and changing values when you click on an individual section. Click **Save changes** when finished:

Note: Configuration at the node level will affect all user accounts and group accounts on that node performing Aspera transfers.

Section	Configuration Details
Database	Configuring policy and logging level settings.
Transfer Server	Setting transfer server IP address and port.
HTTP Fallback Server	Enable and configure HTTP / HTTPS fallback server.
Docroot	Setting document root and its access permissions.
Authorization	Connection permissions, token key, and encryption requirements.
Bandwidth	Incoming and outgoing transfer bandwidth and policy settings.
Advanced File Handling	File handling settings, such as file block size, overwrite rules, and exclude pattern.
Advanced Network Options	Network IP address, port, and socket buffer settings.

Database

#	Field	Description	Values	Default
1	Host IP	Enter the Aspera Console server's IP address, default 127.0.0.1	valid IPv4 address	127.0.0.1
2	Port	The default value for an Aspera Console installation is 4406. Valid port numbers range between 1 and 65535.	Integer between 1 and 65535	4406
3	User	User login for the database server.	text string	blank
4	Database Name	Name of the database used to store Aspera transfer data.	text string	blank
5	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 time frame.	Integer between 1 and 40	10
6	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
7	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
8	Session Progress Interval	The frequency at which an Aspera node logs transfer session information, up to 65535 seconds.	Positive integer	1
9	File Events	Setting this value to **true** enables the logging of complete file paths and file names. Performance may 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.	• true • false	true
10	File Progress	Setting this value to **true** will log file status such as bytes transferred at a given interval.	• true • false	true
11	File Progress Interval	The frequency at which an Aspera node logs file transfer information, up to 65535 seconds. The default setting of 1 logging sessions every second.	Integer between 1 and 65535	1
12	Files Per Session	The value is the cut-off point for file names logged in a given session. For example, 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 will log all file names for a given session.	Positive integer or zero	0
13	Ignore Empty Files	Setting this to **true** will block the logging of zero-byte files.	• true	false

#	Field	Description	Values	Default
			• false	
14	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
15	Rate Events	Setting this to **true** will log changes made to the Target Rate, Minimum Rate, and Transfer Policy of a transfer by any user or Aspera node administrator during a transfer.	• true • false	true

Transfer Server

#	Field	Description	Values	Default
1	Bind 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. Setting the value to **0.0.0.0** allows the Aspera transfer server to accept transfer requests on all network interfaces for this node. Alternatively, a specific network interface address may be specified.	Valid IPv4 address	127.0.0.1
2	Bind Port	The port at which the transfer server will accept transfer requests.	Integer between 1 and 65535	40001

HTTP Fallback Server

Note: While Console can change a node's settings for HTTP fallback, Console does not support HTTP fallback for transfers it initiates.

#	Field	Description	Values	Default
1	Cert File	The absolute path to an SSL certificate file. If left blank, the default certificate file that came with your HST Server will be used.	file path	blank
2	Key File	The absolute path to an SSL key file. If left blank, the default certificate file that came with your HST Server will be used.	file path	blank
3	Bind Address	This is the network interface address on which the HTTP Fallback Server listens. The default value 0.0.0.0 allows the Aspera HTTP Fallback Server to accept transfer requests on all network interfaces for this node. Alternatively, a specific network interface address may be specified.	valid IPv4 address	0.0.0.0

#	Field	Description	Values	Default
4	Restartable Transfers	Setting this to *true* allows interrupted transfers to resume at the point of interruption.	• true • false	true
5	Session Activity Timeout	Any value greater than 0 sets the amount of time, in seconds, that the HTTP Fallback Server will wait without any transfer activity before canceling the transfer. Notice that this option cannot be left at 0, otherwise interrupted HTTP Fallback sessions will get stuck until server or asperacentral is restarted.	Positive integer	0
6	Enable HTTP	Enables the HTTP Fallback Server that allows failed UDP transfers to continue over HTTP.	• true • false	false
7	HTTP Port	The port on which the HTTP server listens. Valid port numbers range between 1 and 65535.	positive integer	8080
8	Enable HTTPS	Enables the HTTPS Fallback Server that allows failed UDP transfers to continue over HTTPS.	• true • false	false
9	HTTPS Port	The port on which the HTTPS server listens. Valid port numbers range between 1 and 65535.	positive integer	8443

Docroot

#	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	N/A
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	N/A
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	N/A
4	Browse Allowed	Setting this to true allows users to browse the directory.	• true • false	N/A

Authorization

#	Field	Description	Values	Default
1	Incoming Transfers	The default setting of **allow** allows users to transfer to this computer. Setting this to **deny** will prevent transfers to this computer. When set to **require token**, only transfers initiated	• allow • deny	allow

#	Field	Description	Values	Default
		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.	• require token	
2	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 to organizations requiring custom authorization rules.	HTTP URL	blank
3	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
4	Outgoing Transfers	The default setting of **allow** allows users to transfer from this computer. Setting this to **deny** will prevent transfers from this computer. When set to **require 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 • require token	allow
5	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, default blank	
6	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
7	Token Encryption Cipher	The cipher used to generate encrypted authorization tokens.	• aes-128 • aes-192 • aes-256	aes-128
8	Token Encryption Key	This is the secret token 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 (Token-based Authorization Topic) for more information.	Text string	blank
9	Token Life (seconds)	Sets token expiration for users of web-based transfer applications.	Positive integer	1200
10	Encryption Allowed	Describes the type of transfer encryption accepted by this computer. When set to **any** the computer allows both encrypted	• any • none	any

#	Field	Description	Values	Default
		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.	• aes-128	

Bandwidth

#	Field	Description	Values	Default
1	Incoming Vlink ID	The value sets the 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. See "Configuring Virtual Links" on page 34	Pre-defined value	0
2	Incoming Target Rate Cap (Kbps)	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
3	Incoming Target Rate Default (Kbps)	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 **Fixed** policy.	Positive integer	10000
4	Incoming Target Rate Lock	After an incoming transfer is started, its target rate may be modified in real time. The default setting **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
5	Incoming Minimum Rate Cap (Kbps)	The value sets the Minimum Rate Cap for incoming 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
6	Incoming Minimum Rate Default (Kbps)	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	Positive integer	0

#	Field	Description	Values	Default
		setting is not relevant to transfers with a **Fixed** policy.		
7	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 **Fixed** policy.	• true • false	false
8	Incoming Bandwidth 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
9	Incoming Bandwidth 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 (such as, 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.	• fixed • high • fair • low	fair
10	Incoming Bandwidth 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
11	Outgoing Vlink ID	The value sets the 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. The Vlink ID is a unique numeric identifier. See “Configuring Virtual Links” on page 34	Pre-defined value	0
12	Outgoing Target Rate Cap (Kbps)	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
13	Outgoing Target Rate Default (Kbps)	This value represents the initial rate for outgoing transfers, in kilobits per second.	Positive integer	10000

#	Field	Description	Values	Default
		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 Fixed policy.		
14	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
15	Outgoing Minimum Rate Cap (Kbps)	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
16	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 **Fixed** policy.	Positive integer	0
17	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 **Fixed** policy.	• true • false	false
18	Outgoing Bandwidth 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
19	Outgoing Bandwidth Policy Allowed	The value chosen sets the allowed Bandwidth Policy for outgoing 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 (for example, 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
20	Outgoing Bandwidth 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

Advanced File Handling

#	Field	Description	Values	Default
1	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
2	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)	0644
3	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
4	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 is a non-Windows receiver and when Directory Create Mode is not specified.	Positive integer (octal)	0755
5	Read Block Size (bytes)	This is a performance tuning parameter for an Aspera sender. It represents the number of bytes an Aspera 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 size, which may be different for different operating systems.	Positive integer	0
6	Write Block Size (bytes)	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 systems with very	• true • false	true

#	Field	Description	Values	Default
		high concurrency (because memory utilization will grow with the number of concurrent transfers).		
8	Max File Cache Buffer (bytes)	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	Extension name of a class of special files holding metadata information of regular data files. Useful in the context of resuming partially completed transfers. During resume mode (-k1/2/3), each data file has a corresponding metadata file with the same name and the pre-specified resume suffix.	text string	aspx
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	undefined
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 to any, follow client configurations (-o PreCalculateJobSize={yes|no}). If set to no, disable calculating job	• any • yes • no	any

#	Field	Description	Values	Default
		size before transferring. If set to yes, enable calculating job size before transferring.		
1 5	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	false
1 6	File checksum method	Specify the type of checksum to calculate for transferred files. The content of transfers can be verified by comparing the checksum value at the destination with the value read at the source.	• any • md5 • sha1	any
1 6	Partial Suffix	Set the file suffix for partially downloaded files.		**.aspx**

Advanced Network 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	Specify a port number for server-side ascp to bind its UDP connection. This also prevents client ascp processes from binding to same UDP port. Valid port numbers range between 1 and 65535.	Positive integer	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	Maximum Socket Buffer (bytes)	Upper bound the UDP socket buffer of an ascp session below the input value. 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
5	Minimum socket buffer (bytes)	Set the minimum UDP socket buffer size for an ascp session.	Positive integer	0

Node Account-Level Configuration Options

When configuring users and groups on a node from Console, both group-level and user-level settings share the same configuration options. This topic covers the following configuration sections:

Section	Configuration Details
Docroot	Setting document root and its access permissions.
Authorization	Connection permissions, token key, and encryption requirements.
Bandwidth	Incoming and outgoing transfer bandwidth and policy settings.

Section	Configuration Details
Advanced File Handling	File handling settings, such as file block size, overwrite rules, and exclude pattern.
Advanced Network Options	Network IP, port, and socket buffer settings.

Docroot

#	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	N/A
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	N/A
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	N/A
4	Browse Allowed	Setting this to true allows users to browse the directory.	• true • false	N/A

Authorization

#	Field	Description	Values	Default
1	Incoming Transfers	The default setting of **allow** allows users to transfer to this computer. Setting this to **deny** will prevent transfers to this computer. When set to **require 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 • require token	allow
2	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 to organizations requiring custom authorization rules.	HTTP URL	blank
3	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
4	Outgoing Transfers	The default setting of **allow** allows users to transfer from this computer. Setting this to **deny** will prevent transfers from this computer. When set to **require token**, only transfers initiated with valid tokens will be	• allow • deny • require token	allow

#	Field	Description	Values	Default
		allowed to transfer from this computer. Token-based transfers are typically employed by web applications such as Faspex and require a Token Encryption Key.		
5	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, default blank	
6	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
7	Token Encryption Cipher	The cipher used to generate encrypted authorization tokens.	• aes-128 • aes-192 • aes-256	aes-128
8	Token Encryption Key	This is the secret token 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 (Token-based Authorization Topic) for more information.	Text string	blank
9	Token Life (seconds)	Sets token expiration for users of web-based transfer applications.	Positive integer	1200
1 0	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

Bandwidth

#	Field	Description	Values	Default
1	Incoming Vlink ID	The value sets the 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. See "Configuring Virtual Links" on page 34	Pre-defined value	0

#	Field	Description	Values	Default
2	Incoming Target Rate Cap (Kbps)	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
3	Incoming Target Rate Default (Kbps)	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 **Fixed** policy.	Positive integer	10000
4	Incoming Target Rate Lock	After an incoming transfer is started, its target rate may be modified in real time. The default setting **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
5	Incoming Minimum Rate Cap (Kbps)	The value sets the Minimum Rate Cap for incoming 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
6	Incoming Minimum Rate Default (Kbps)	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 **Fixed** policy.	Positive integer	0
7	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 **Fixed** policy.	• true • false	false
8	Incoming Bandwidth 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
9	Incoming Bandwidth 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	• fixed • high • fair	fair

#	Field	Description	Values	Default
		transfer based on policy setting. When set to **high**, transfers with a Policy of high and less aggressive transfer policies (such as, 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.	• low	
10	Incoming Bandwidth 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
11	Outgoing Vlink ID	The value sets the 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. The Vlink ID is a unique numeric identifier. See "Configuring Virtual Links" on page 34	Pre-defined value	0
12	Outgoing Target Rate Cap (Kbps)	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
13	Outgoing Target Rate Default (Kbps)	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 Fixed policy.	Positive integer	10000
14	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
15	Outgoing Minimum Rate Cap (Kbps)	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
16	Outgoing Minimum Rate Default	This value represents the initial minimum rate for outgoing transfers, in kilobits per second.	Positive integer	0

#	Field	Description	Values	Default
		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 **Fixed** policy.		
17	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 **Fixed** policy.	• true • false	false
18	Outgoing Bandwidth 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
19	Outgoing Bandwidth Policy Allowed	The value chosen sets the allowed Bandwidth Policy for outgoing 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 (for example, 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
20	Outgoing Bandwidth 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

Advanced File Handling

#	Field	Description	Values	Default
1	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
2	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)	0644

#	Field	Description	Values	Default
3	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
4	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 is a non-Windows receiver and when Directory Create Mode is not specified.	Positive integer (octal)	0755
5	Read Block Size (bytes)	This is a performance tuning parameter for an Aspera sender. It represents the number of bytes an Aspera 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 size, which may be different for different operating systems.	Positive integer	0
6	Write Block Size (bytes)	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 systems with very high concurrency (because memory utilization will grow with the number of concurrent transfers).	• true • false	true
8	Max File Cache Buffer (bytes)	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	Extension name of a class of special files holding metadata information of regular data files. Useful in the context of resuming partially completed transfers. During resume mode (-k1/2/3), each data file has a corresponding	text string	aspx

#	Field	Description	Values	Default
		metadata file with the same name and the pre-specified 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	undefined
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 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.	• any • yes • no	any
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	false
16	File checksum method	Specify the type of checksum to calculate for transferred files. The content of transfers can be verified by comparing the checksum value at the destination with the value read at the source.	• any • md5 • sha1	any
16	Partial Suffix	Set the file suffix for partially downloaded files.		**.aspx**

Advanced Network 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	Specify a port number for server-side ascp to bind its UDP connection. This also prevents client ascp processes from binding to same UDP port. Valid port numbers range between 1 and 65535.	Positive integer	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	Maximum Socket Buffer (bytes)	Upper bound the UDP socket buffer of an ascp session below the input value. 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
5	Minimum socket buffer (bytes)	Set the minimum UDP socket buffer size for an ascp session.	Positive integer	0

Transfer References

Simple Transfer Options

The following tables provide information on additional configurable settings that are available when creating simple transfers.

Connection

Fasp Port (UDP)	Specify the UDP port for FASP file transfers.
Fasp proxy	Enable transferring through a FASP proxy server, and specify the proxy host address, port, username, and password. This feature enables the source node to bypass restrictions to the destination node for this specific transfer by using a proxy.

Security

Content protection	Check the option to enable the content protection that encrypts the files on destination, using the entered password.
Transport encryption	Select aes-128 to transfer with this encryption method.

Transfer

Target rate	Specify the transfer target rate.

Minimum rate	Set the transfer minimum rate
Bandwidth policy	Choose a transfer policy among fixed/high/fair/low.
Retry policy	Check the option to enable the retry policy, as well as specify the number of attempts and the duration.

Notifications

Email address	To send status notifications for transfer events (start, success, or error), enter an email address and click Add. When the email address appears in the table, specify which email template to use for each transfer event.

File Handling

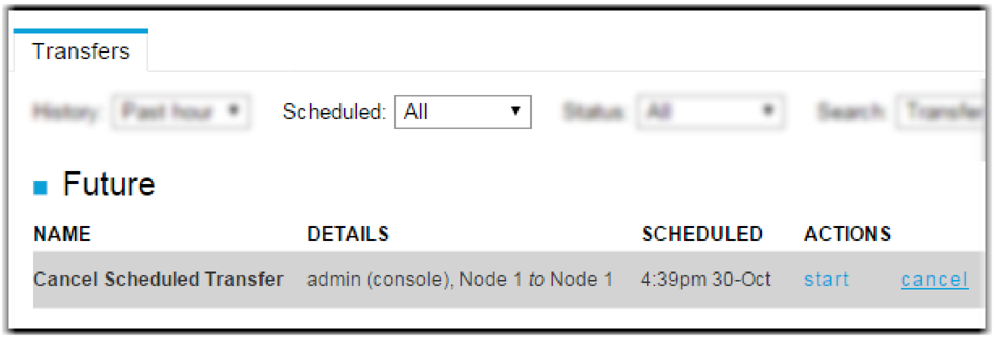
Timestamp Filtering	Select this option to exclude files modified in the designated number of seconds.
Resume policy	Specify a resume policy and the overwrite rule when the file exists on the destination.
File attributes	Check the option to preserve the file permissions on the destination.
Symlinks	Specify how to deal with symbolic links: follow, copy, copy and force, or skip. Leave this option blank if the source is on Windows. For all others, leaving it blank is the same as choosing "follow".
Source Archiving	Move source files to a designated directory after completing a transfer. The transfer's session details page will display the archive directory's filepath as the After transfer path. For more information on session details, see “Transfer Details” on page 43. **Note:** The After transfer path will only be visible in the session details of the Console that initiated the transfer. Another Console monitoring the same managed nodes will not have access to the After transfer path. **Note:** Rerunning the transfer may generate a "No such file or directory" error since the source files were moved to the archive directory.
Delete empty source subdirectories	This option becomes available if you selected Source Archiving. Select this option to delete any subdirectory that is emptied by the source archiving. **Note:** Console does not delete the top-most directory in the source path.
Source Deletion	Check the option to delete the transferred files from the source computer.
Exclude filter	Enter file-name pattern Console uses to determine what files to exclude from the transfer. You can use the following two symbols in the pattern: • `*`: Matchers zero to many characters in a string. For example, the `*.tmp` pattern matches `.tmp` and `abcde.tmp`. • `?`: Matches any one character except a / or `.` when preceded immediately by a / character. For example, the `t?p` pattern matches `tmp`, but not `temp`; and the `?exe` pattern matches `file.exe` but not `.exe.file`, because the filepath would be `/.exe.file`.
Include filter	Enter file-name pattern Console uses to determine what files to include in the transfer. Only files matching the filter are transferred.

	You can use the following two symbols in the pattern: • `*`: Matchers zero to many characters in a string. For example, the `*.tmp` pattern matches `.tmp` and `abcde.tmp`. • `?`: Matches any one character except a / or `.` when preceded immediately by a / character. For example, the `t?p` pattern matches `tmp`, but not `temp`; and the `?exe` pattern matches `file.exe` but not `.exe.file`.

Advanced

Initiator	Check this option to initiate transfers from the destination node (if possible). Console normally initiates transfers from the source node unless the source is an unmanaged node.
ascp version	Select the option to use **ascp4** for this transfer. The initiating node must have **ascp4** available. **Note:** Both nodes need to be running the same version of HST Server to use **ascp4**. Also, the `apply_local_docroot` parameter in `aspera.conf` is not currently supported. **Note: ascp4** is only available for HST Server 3.8 and later.
fasp datagram size (MTU)	Select the option and enter the datagram size in bytes.
Read block size	Check the option and enter the read block size in bytes.
Write block size	Check the option and enter the write block size in bytes.

Transfer Time

Transfer	Specify when to submit the transfer. **Note:** You can cancel scheduled simple transfers by going to **Activity > Transfers**. Click the **Scheduled** drop-down menu and select **All**. In the row for the transfer, click **Cancel**.  Transfers Scheduled: All Future NAME | DETAILS | SCHEDULED | ACTIONS Cancel Scheduled Transfer | admin (console), Node 1 *to* Node 1 | 4:39pm 30-Oct | start cancel

Smart Transfer Options

The following tables provide information on additional configurable settings that are available when creating smart transfers.

Connection

Fasp Port (UDP)	Specify the UDP port for FASP file transfers.

Fasp proxy	Enable transferring through a FASP proxy server, and specify the proxy host address, port, username, and password. This feature enables the source node to bypass restrictions to the destination node for this specific transfer by using a proxy.

Security

Content protection	Check the option to enable the content protection that encrypts the files on destination, using the entered password.
Transport encryption	Select aes-128 to transfer with this encryption method.

Transfer

Target rate	Specify the transfer target rate.
Minimum rate	Set the transfer minimum rate
Bandwidth policy	Choose a transfer policy among fixed/high/fair/low.
Retry policy	Check the option to enable the retry policy, as well as specify the number of attempts and the duration.

Notifications

Email address	To send status notifications for transfer events (start, success, or error), enter an email address and click Add. When the email address appears in the table, specify which email template to use for each transfer event.

File Handling

Timestamp Filtering	Select this option to exclude files modified in the designated number of seconds.
Resume policy	Specify a resume policy and the overwrite rule when the file exists on the destination.
File attributes	Check the option to preserve the file permissions on the destination.
Symlinks	Specify how to deal with symbolic links: follow, copy, copy and force, or skip. Leave this option blank if the source is on Windows. For all others, leaving it blank is the same as choosing "follow".
Source Archiving	Move source files to a designated directory after completing a transfer. The transfer's session details page will display the archive directory's filepath as the After transfer path. For more information on session details, see “Transfer Details” on page 43. **Note:** The After transfer path will only be visible in the session details of the Console that initiated the transfer. Another Console monitoring the same managed nodes will not have access to the After transfer path. **Note:** Rerunning the transfer may generate a "No such file or directory" error since the source files were moved to the archive directory.
Delete empty source subdirectories	This option becomes available if you selected Source Archiving. Select this option to delete any subdirectory that is emptied by the source archiving.

	Note: Console does not delete the top-most directory in the source path.
Source Deletion	Check the option to delete the transferred files from the source computer.
Exclude filter	Enter file-name pattern Console uses to determine what files to exclude from the transfer. You can use the following two symbols in the pattern: • `*`: Matchers zero to many characters in a string. For example, the `*.tmp` pattern matches `.tmp` and `abcde.tmp`. • `?`: Matches any one character except a `/` or `.` when preceded immediately by a `/` character. For example, the `t?p` pattern matches `tmp`, but not `temp`; and the `?exe` pattern matches `file.exe` but not `.exe.file`, because the filepath would be `/.exe.file`.
Include filter	Enter file-name pattern Console uses to determine what files to include in the transfer. Only files matching the filter are transferred. You can use the following two symbols in the pattern: • `*`: Matchers zero to many characters in a string. For example, the `*.tmp` pattern matches `.tmp` and `abcde.tmp`. • `?`: Matches any one character except a `/` or `.` when preceded immediately by a `/` character. For example, the `t?p` pattern matches `tmp`, but not `temp`; and the `?exe` pattern matches `file.exe` but not `.exe.file`.

Advanced

Initiator	Check this option to initiate transfers from the destination node (if possible). Console normally initiates transfers from the source node unless the source is an unmanaged node.
ascp version	Select the option to use **ascp4** for this transfer. The initiating node must have **ascp4** available. **Note:** Both nodes need to be running the same version of HST Server to use **ascp4**. Also, the `apply_local_docroot` parameter in `aspera.conf` is not currently supported. **Note: ascp4** is only available for HST Server 3.8 and later.
fasp datagram size (MTU)	Select the option and enter the datagram size in bytes.
Read block size	Check the option and enter the read block size in bytes.
Write block size	Check the option and enter the write block size in bytes.

Scheduling

Start	Click the calendar icon to select a date and time that serves as the starting basis for your recurring smart transfers. Based on the "Start" entry, Console will calculate the run time for the next occurrence (that matches the repeat rules). For example, if your start date is Friday, April 8, but your transfer is scheduled to run on Saturdays, then the first transfer will occur on Saturday, April 9.
Repeat every	Select the number of minutes, hours, days, weeks, or months to repeat this transfer. When *weeks* is selected, you can enable the requisite days of the week. When *months* is selected, you can specify whether to perform the transfer on a

	specific day of the month or on the "nth ___day" of the month (for example, 1st Sunday).
Until	Click the calendar icon to select a "do not go beyond" date and time. Your smart transfer will not repeat beyond this entry.
Time zone	Select your timezone from the drop-down list.

Important: When you have more than one destination, you can override the default smart transfer settings (with the exception of scheduling) shown in the More Options panel for each individual destination.

Specify Base for Source Path

When selecting the source for a simple or smart transfer, you have the option to select **Specify base for source path(s)** to specify a portion of the source path to *remove* to place the transferred files directly into the destination folder without its hierarchy of directories.

For example, a source computer has a **sent_files/project** directory containing these folders and files:

- `/shared_files/project/presentation`
- `/shared_files/project/video_footage/take1`
- `/shared_files/project/video_footage/take2`
- `/shared_files/project/video_footage/take3`

If **shared_files/project** directory is the source, by default, the transfer includes the `sent_files` directory and the entirety of its contents, including its hierarchy of directories. If the destination directory is specified as **/incoming**, your transferred files appear as follows on the destination computer:

```
docroot/incoming/shared_files/project/presentation
docroot/incoming/shared_files/project/video_footage/take1
docroot/incoming/shared_files/project/video_footage/take2
docroot/incoming/shared_files/project/video_footage/take3
```

By selecting **Specify base for source paths(s)**, the **project** folder can be excluded. Entering "/shared_files/project" in the field removes that part of the source path. Only the **presentation** and **video_footage** directories are transferred. The transferred files appear as follows on the destination computer:

```
docroot/incoming/presentation
docroot/incoming/video_footage/take1
docroot/incoming/video_footage/take2
docroot/incoming/video_footage/take3
```

If any files or folders selected for transfer fall outside the specified base path, they are omitted from the transfer. For example, if the specified path is **/shared_files/project/video_footage**, then **presentation** is not transferred at all because it is not in **video_footage**. Only **take1**, **take2**, and **take3** are transferred. The transferred files appear as follows on the destination computer:

```
docroot/incoming/take1
docroot/incoming/take2
docroot/incoming/take3
```

Tip: Specify base for source paths(s) can also be used to include *more* path depth than the default. If the source-base path is specified as **/shared_files**, then **project** and all files and folders in its folder hierarchy are included. Similarly, if the source-base path is specified as **/**, the entire source path and all fields and folders in its folder hierarchy are transferred.

Report References

Reference: Basic Report Organization Options

Field	Description
Client Address	Organize / summarize report by Client IP Address (client = initiator of the transfer)
Contact	Organize by the 'Contact' shown for a transfer. This might be a Console user name, a Faspex Server user name, SSH account, or customized value obtained from a transfer cookie. Examples: "admin (console)", "aspera (ssh)", "aspera (faspex)".
File	Display a detail row for every file in every transfer.
File Extension	Organize / summarize report by file extension.
Server Address	Organize / summarize report by Server IP Address.
Session	Display a row for every transfer session. A transfer session represents one attempt to transfer.
Transfer	Display a row for every transfer. A transfer may have multiple sessions if it took multiple attempts to finish.

Reference: Built-In Fields for Custom Field Rules

Built-In Fields Available for Creating Custom Field Rules (for Transfer-Level Fields)

Transfer Field	Description
Client Address	IP address of transfer initiator.
Client User	Client-side username. Null for all transfers, except for transfers initiated by the Console.
Contact	Contact assigned by Console. This can be a Console user name, a Faspex Server user name, SSH account, or customized value obtained from a transfer cookie. Examples: "admin console", "aspera ssh", "aspera faspex".
Cookie	Custom identifying text attached to a transfer session. This text is used by the Console to identify and name transfers.
Destination Address	IP address of transfer destination (use for general purpose).
Destination Path	The file path on the destination machine.
Destination User	If upload, dest_user is the server user. If download, dest_user is client user (NULL, unless initiated from Console). For everyday purposes, recommend using contact field instead.
Direction	The direction of the transfer from the perspective of the client. "Upload" if the transfer is a push; "Download" if the transfer is a pull.
Meta-tags	JSON hash used to tag transfers with additional data.
Faspex Metadata	Information provided by Faspex, encoded in the transfer cookie. See “Basic Report Example: Faspex Metadata” on page 179.
Server Address	IP address of the server.
Server User	SSH account specified when the transfer starts (should always be displayed).
Source Address	IP address of transfer source.

Transfer Field	Description
Source Paths	File paths on the source machine.
Source User	If upload, source_user is the client user. If download, source_user is server user.
Started Via	The name of the application (Aspera or custom) that is responsible for initiating the transfer (for example, aspera.scp, aspera.sync, etc.).
Token	Security token used for the transfer (note that this depends on whether or not the application that started the transfer is configured to use tokens).
Transfer Name	Human-readable name assigned to a transfer. This name may have been keyed in by the user or automatically set by an application.

Built-In Fields Available for Creating Custom Field Rules (for File-Level Fields)

Note: Setting up file-level custom fields is NOT recommended for customers that transfer many small files, as this will result in scaling issues.

File Field Name	Description
File Bytes Transferred	Total bytes successfully received over the network.
File Error Desc	Error message for the file, if any.
File Extension	Portion of the filename after the last period (.)
File Full Destination Path	File's full path from the destination's point-of-view.
File Full Source Path	File's full path from the source's point-of-view.
File Name	Name of the file, without its path (for example, `"my_file.txt"` rather than `"C:\temp\my_file.txt"`)
File Size	Size of the file in bytes.
File Status	Status of transfer or file (for example, "running," "completed," "canceled" or "error").

Using Matchers in Custom Rules

You can use regular expression (Regex) and JSON matchers (except in fields that expect a number) to trigger a rule based on a particular phrase.

For example, you can use a Regex matcher to set a custom field named "engagement_id" to text the ID from an URL in a cookie:

- **Built-in field**: `Cookie`
- **Operator**: `matching regular expression`
- **Expression**: `.*engagement\/(?<text>.+?)\/transferid`
- **"engagement_id" custom field value**: `<text>`.

In other words, if the cookie matches the regular expression, then set the custom field value to the value of `<text>`. So, if the cookie has the URL: `https://example.com/customer_engagement/1234/transferid`, the "engagement_id" custom field value is 1234.

Another example is using JSON matching to set a custom field (for example, "faspex_event") to the event name in Faspex metadata:

- **Built-in field**: `Faspex Metadata`
- **Operator**: `JSON match`
- **Expression**: `\{.*"Event":"(?<event>.+?)".*\)`

- **"faspex_event" custom field value**: <event>.

In other words, if the Faspex Metadata JSON matches the expression, then set the custom field value to the value of <event>. So, if the Faspex metadata is {"Event":"Summer","_pkg_uuid":"abde20e28db24bfdb513c0a3de3bb8ff", "pkg_name":"test"...}, the "faspex_event" custom field's value is Summer.

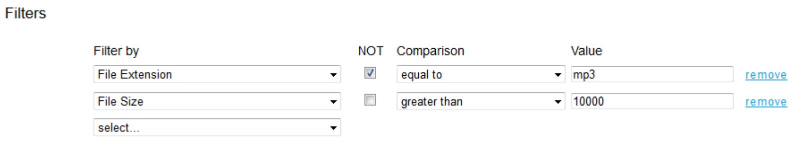
The expression is interpreted as follows:

Expression	Interpretation	Characters matched in the example
\{	Finds a left curly bracket character	{
.*	Followed by 0 or more characters until the next piece of the expression ("Event":")	
"Event":"	Followed by the first match of the exact text "Event":"	"EVENT":"
(?<event>.+?)	Followed by at least 1 character" (save these characters and call them <event>)	Summer
"	Followed by a quotation mark	"
.*	Followed by 0 or more characters until the next piece of the expression (\{)	,"_pkg_uuid":"abde20e28db24bfdb513c0a3de3bb8ff", "pkg_name":"test"...
\}	Followed by a right curly bracket character	}

Reference: Reporting Filters

Console provides built-in filters that allow you to specify conditions for limiting the data included in your report.

Column Heading	Description
Filter By	Select from a list of parameter names.
NOT	Appears as a checkbox, where unchecked represents "is" and checked represents "is not" (for example, file extension *is not* equal to tmp)
Comparison	Select from a list of operators (for example,, equal to, greater than, etc.).
Value	Input a parameter value to complete the filter expression.



Filters

Filter by	NOT	Comparison	Value	
File Extension	☑	equal to	mp3	remove
File Size	□	greater than	10000	remove
select...				

Important: Once you have added a filter, you may remove it by clicking the **Remove** hyperlink.

The following filter parameters are available within the **Filter By** drop-down list:

Parameter Name	Parameter Description
{Custom Field Names}	Displays custom fields that you have configured for the SQL database.

Parameter Name	Parameter Description
File Bytes Transferred	File bytes successfully received over the network by the destination.
File Bytes Written	Files bytes successfully received over the network by the destination, plus bytes skipped for data already present at the destination.
File Error Description	File's error message, if any.
File Extension	Portion of the filename after the last period (.)
File Fullpath	File's directory tree hierarchy.
File Name	Name of the file, without its path (for example, "my_file.txt," rather than "C:\temp\my_file.txt").
File Session Status	Status of file session (for example, "running," "completed," "canceled" or "error"), where a file session is one file in a transfer session. A file record may group together more than one file session record if, during a transfer session, one of the files fails or is interrupted. In the next transfer session (when the transfer is retried or a hot folder handles the next batch of files to arrive), then that particular file may be retried. This will result in another file session record being created.
File Size	Size of the file in bytes.
File Status	The file status will be the status of the last/most recent file session for the file (for example, "running," "completed," "canceled" or "error").
SSH Account	SSH account specified when the transfer starts.
Transfer Average Rate	Average transfer rate in bits per second.
Transfer Bytes Lost	Number of bytes sent by source for a particular file, but never received by destination, or never written to disk.
Transfer Bytes Transferred	Total bytes successfully received over the network by the destination.
Transfer Bytes Written	Total bytes successfully received over the network by the destination, plus bytes skipped for data already present at the destination.
Transfer Client Address	IP address of transfer initiator.
Transfer Contact	Contact assigned by Console. This can be a Console user name, a Faspex user name, SSH account, or customized value obtained from a transfer cookie. Examples: "admin (console)", "aspera (ssh)", "michael (faspex)"
Transfer Cookie	Custom identifying text attached to a transfer session. This text is used by the Console to identify and name transfers.
Transfer Destination Address	IP address of transfer destination.
Transfer Destination Path	The file path on the destination machine.
Transfer Error Description	Error message for transfer or file, if any.
Transfer Files Completed	Number of files successfully verified at destination (i.e., the number of files actually transferred plus the number of files that were already at destination).
Transfer Files Failed	Number of files that failed to transfer.
Transfer Name	Human-readable name assigned to a transfer. This name may have been keyed in by the user or automatically set by an application.

Parameter Name	Parameter Description
Transfer Server Address	IP address of transfer server.
Transfer Session Status	Indicates status of transfer session (for example, "running," "completed," "canceled" or "error"), where the transfer session represents one execution of ascp (i.e., one attempt to transfer). **Note:** When a transfer session is *interrupted or fails* and is configured to retry, a second transfer session will begin after the configured retry interval has elapsed.
Transfer Source Address	IP address of transfer source.
Transfer Source Paths	File paths on the source machine.
Transfer Status	A transfer will group together transfer sessions into a single item. The transfer status will be the status of the last/most recent transfer session for the transfer (for example, "running," "completed," "canceled" or "error").

Reference: SQL Variables for Advanced Reports

When creating your advanced report, you may utilize the SQL variables listed below. These variables also appear within Console's built-in, SQL script text help.

SQL Variable	Description
$TBL_FILES	Files table. One record in this table represents one file. At run time, this variable gets replaced with the SQL name of the table containing the file data (currently 'rpt_transfer_files'). Please note the following distinction: • A FILE record can have multiple associated TRANSFER SESSION FILE records (if a file took more than one attempt to transfer). • A FILE record has one and only one associated TRANSFER record ($TBL_TRANSFER_FILES.transfer_id = $TBL_TRANSFERS.id).
$TBL_TRANSFER_SESSIONS	Transfer sessions table. One record in this table represents one attempt to transfer data. If you start a transfer and it fails, then automatically retries and succeeds, there will be two records in this table, one for the initial attempt and one for the automatic retry. For hot folder transfers, each session represents one attempt to transfer a batch of files that are currently available. If new files become available while the first batch is in progress, these may be transferred in a subsequent session, resulting in an additional record in this table. At run time, this variable gets replaced with the SQL name of the table containing the transfer session data (currently 'rpt_transfer_sessions'). Please note the following distinction: • A TRANSFER SESSION record can have multiple TRANSFER SESSION FILE records (if the session attempted to transfer more than one file). • A TRANSFER SESSION record has one and only one associated TRANSFER record ($TBL_TRANSFER_SESSIONS.transfer_id = $TBL_TRANSFERS.id).
$TBL_TRANSFER_SESSION_FILES	Files within a transfer session. One record in this table represents one attempt to transfer a file. At run time, this variable gets

SQL Variable	Description
	replaced with the SQL name of the table containing the file session data (currently 'rpt_transfer_session_files'). Please note the following distinction: • A TRANSFER SESSION FILE record has one and only one associated FILE RECORD ($TBL_TRANSFER_SESSION_FILES.transfer_file_id = $TBL_FILES.id). • A TRANSFER SESSION FILE record has one and only one associated TRANSFER SESSION record ($TBL_TRANSFER_SESSION_FILES.transfer_session_id = $TBL_TRANSFER_SESSIONS.id).
$TBL_NODES	A table containing one record for each node, whether managed or unmanaged. At run time, this variable gets replaced with the SQL name of the table containing the node data (currently 'rpt_transfer_nodes').
$TBL_TRANSFERS	A TRANSFER groups together TRANSFER SESSIONS to tie together retry attempts and hot folder file batches. Related TRANSFER SESSIONS are grouped together so that no matter how many times the session was interrupted and retried, only a single record will be present in this table. At run time, this variable gets replaced with the SQL name of the table containing the transfer data (currently 'rpt_transfers'). Please note the following distinction: • A TRANSFER record can have multiple TRANSFER SESSION records (if multiple attempts or batches were required to transfer all the data). • A TRANSFER record can have multiple FILE records (if the transfer consisted of more than one file).
$FINAL_RESULT_TABLE	This is the table where you place your final results. The data displayed on reports comes directly from this table. At run time, this variable gets replaced with a name based on an auto-generated numeric id (for example, 'report_100_results').
$TMP_*TABLENAME*	If you need any temporary tables for intermediate record processing, give them names starting with "$TMP_" (for example, $TMP_UNIQUE_IP_ADDRESSES). At run time, these variables get replaced with a name based on an auto-generated numeric id (for example, 'report_100_temp_unique_ip_addresses').
$USER_ID	This is the login id of user requesting report. At run time, this variable gets replaced with the numeric id of the user requesting the report.
$REPORT_PERIOD_START	Report period start. The user running this report will be prompted for a value at request time. (Value is converted to UTC before substitution).
$REPORT_PERIOD_END	Report period end. The user running this report will be prompted for a value at request time. (Value is converted to UTC before substitution).
$*ANYTHING_ELSE*	Any $*NAME* that does not match one of the variables is presumed to be a custom variable whose value will be provided by the report requester. See "Editing Custom Variables" on page 80 for instructions on how to create and configure a custom variable.

Reference: Database Fields for Advanced Reports

When creating your advanced report, you may utilize the database fields listed below. These fields (and corresponding descriptions) also appear within Console's built-in, SQL script text help.

Note: The term "client" refers to the machine initiating a transfer request. The term "server" refers to the machine receiving the request. These terms do not describe the direction of the file transfer. As long as a machine is the transfer initiator, it does not matter whether the machine is sending a file or receiving a file.

Database Field	Description	Table
args_attempted	Number of items specifically selected by the user (either in GUI or command line).	$TBL_TRANSFER_SESSIONS
args_completed	Out of the number of arguments attempted, the number completed successfully.	$TBL_TRANSFER_SESSIONS
aspera_version	Aspera product version for the node machine.	$TBL_NODES
avg_loss_pct	Average packet loss over the network, which is calculated as a percentage.	$TBL_TRANSFER_SESSIONS
avg_rate	Average transfer rate in bits per second.	$TBL_TRANSFERS, $TBL_TRANSFER_SESSIONS
bytes_config	The number of contiguous bytes that have been transferred to the destination.	$TBL_TRANSFER_SESSION_FILES, $TBL_FILES
bytes_lost	Number of bytes sent by source for a particular file, but never received by destination, or never written to disk.	$TBL_TRANSFERS, $TBL_TRANSFER_SESSIONS, $TBL_TRANSFER_SESSION_FILES, $TBL_FILES
bytes_pretransfer	If the server is configured to do so, calculates size of the transfer before the transfer starts. On the server, this corresponds to the "pre-calculate job size" setting.	$TBL_TRANSFERS, $TBL_TRANSFER_SESSIONS
bytes_remaining	Total bytes waiting to be sent over the network to the destination.	$TBL_TRANSFERS, $TBL_TRANSFER_SESSIONS
bytes_transferred	Total bytes successfully received over the network by the destination.	$TBL_TRANSFERS, $TBL_TRANSFER_SESSIONS, $TBL_TRANSFER_SESSION_FILES, $TBL_FILES
bytes_written	Total bytes successfully received over the network by the destination, plus bytes skipped for data already present at the destination.	$TBL_TRANSFERS, $TBL_TRANSFER_SESSIONS, $TBL_TRANSFER_SESSION_FILES, $TBL_FILES
cipher	Encryption algorithm.	$TBL_TRANSFERS, $TBL_TRANSFER_SESSIONS
client_console_ip	The client's IP address from the perspective of the Aspera Console	$TBL_TRANSFER_SESSIONS

Database Field	Description	Table
	application (advanced / debugging field).	
client_err_code	Error code reported by the client.	$TBL_TRANSFER_SESSION_FILES, $TBL_TRANSFER_SESSIONS
client_err_desc	Error code description reported by the client.	$TBL_TRANSFER_SESSION_FILES, $TBL_TRANSFER_SESSIONS
client_external_fasp_port	The client's UDP port from the perspective of the server (advanced / debugging field).	$TBL_TRANSFER_SESSIONS
client_external_ip	The client's IP address from the perspective of the server (advanced / debugging field). **Note:** If the client is a managed node and the server is not, then this field is null.	$TBL_TRANSFER_SESSIONS
client_file_basename	File's basename from client's point-of-view.	$TBL_TRANSFER_SESSION_FILES, $TBL_FILES
client_file_extension	File's extension from client's point-of-view.	$TBL_TRANSFER_SESSION_FILES, $TBL_FILES
client_file_fullpath	File's full path from the client's point-of-view.	$TBL_TRANSFER_SESSION_FILES, $TBL_FILES
client_file_index	Arbitrary, unique number assigned to each file within a transfer session (on the client).	$TBL_TRANSFER_SESSION_FILES
client_ip	IP address of the transfer initiator (use for general purpose).	$TBL_TRANSFER_SESSIONS
client_node_id	ID number assigned to the client node.	$TBL_TRANSFER_SESSIONS
client_node_uuid	Universally, unique ID number assigned to the client node.	$TBL_TRANSFER_SESSIONS
client_status	Either the file status (running, completed, error) or the session status reported by the client. **Note:** In some cases, client and server can see different statuses (for example, canceled versus error).	$TBL_TRANSFER_SESSION_FILES, $TBL_TRANSFER_SESSIONS
client_user	Client-side username. Null for all transfers, except for transfers initiated by the Console.	$TBL_TRANSFERS, $TBL_TRANSFER_SESSIONS
contact	Contact assigned by Console. This can be a Console user name, a Faspex user name, SSH account, or customized value obtained from a transfer cookie. Examples: "admin	$TBL_TRANSFERS, $TBL_TRANSFER_SESSIONS

Database Field	Description	Table
	(console)", "aspera (ssh)", "michael (faspex)"	
cookie	Custom identifying text attached to a transfer session. This text is used by the Console to identify and name transfers.	$TBL_TRANSFERS, $TBL_TRANSFER_SESSIONS
dest_endpoint_id	ID number assigned to the destination endpoint.	$TBL_TRANSFER_SESSIONS
dest_file_basename	File's basename from destination's point-of-view.	$TBL_FILES
dest_file_extension	File's extension from destination's point-of-view.	$TBL_FILES
dest_file_fullpath	File's full path from the destination's point-of-view.	$TBL_FILES
dest_ip	IP address of transfer destination (use for general purpose).	$TBL_TRANSFER_SESSIONS
dest_node_id	ID number assigned to the destination node.	$TBL_TRANSFER_SESSIONS
dest_path	The file path on the destination machine.	$TBL_TRANSFERS, $TBL_TRANSFER_SESSIONS
dest_user	If upload, dest_user is the server user. If download, dest_user is client user (NULL, unless initiated from Console). For everyday purposes, recommend using contact field instead.	$TBL_TRANSFERS, $TBL_TRANSFER_SESSIONS
dirs_pretransfer	If the server is configured to do so, calculates number of directories to be transferred. Only calculated if "pre-calculate job size" setting is turned on.	$TBL_TRANSFERS, $TBL_TRANSFER_SESSIONS
dirscans_completed	Number of directory scans completed.	$TBL_TRANSFER_SESSIONS
err_desc	Error message for transfer or file, if any.	$TBL_TRANSFER_SESSION_FILES, $TBL_TRANSFER_SESSIONS
fallback_protocol	If the transfer has been configured to retry using the HTTP fallback protocol, then this field will report "http." If not, will be NULL.	$TBL_TRANSFER_SESSIONS
file_basename	Name of the file, without its path (for example, "my_file.txt," rather than "C:\temp\my_file.txt")	$TBL_TRANSFER_SESSION_FILES, $TBL_FILES
file_extension	Portion of the filename after the last period (.)	$TBL_TRANSFER_SESSION_FILES, $TBL_FILES
file_fullpath	Full path to the file.	$TBL_TRANSFER_SESSION_FILES, $TBL_FILES

Database Field	Description	Table
files_attempted	Number of files attempted to be sent over the network.	$TBL_TRANSFERS, $TBL_TRANSFER_SESSIONS
files_complete	Number of files successfully verified at destination, that is, the number of files actually transferred + number of files that were already at destination.	$TBL_TRANSFERS, $TBL_TRANSFER_SESSIONS
files_failed	Number of files that failed to transfer.	$TBL_TRANSFERS, $TBL_TRANSFER_SESSIONS
files_pretransfer	If the server is configured to do so, calculates number of files to be transferred. Only calculated if "pre-calculate job size" setting is turned on.	$TBL_TRANSFERS, $TBL_TRANSFER_SESSIONS
files_skipped	Number of files skipped.	$TBL_TRANSFERS, $TBL_TRANSFER_SESSIONS
filescans_completed	The number of file scans completed.	$TBL_TRANSFERS, $TBL_TRANSFER_SESSIONS
hostname	The local name of the node machine (which is only filled in for managed nodes). Note that a node machine will be called "localhost" if it hasn't been previously named.	$TBL_NODES
id	Unique integer ID assigned by the Console (used as an internal field).	$TBL_NODES, $TBL_TRANSFERS, $TBL_TRANSFER_SESSIONS, $TBL_TRANSFER_SESSION_FILES, $TBL_FILES
initiated_by_source	Identifies an "upload." If this field is equal to 1, then whoever started the transfer is uploading.	$TBL_TRANSFERS, $TBL_TRANSFER_SESSIONS, $TBL_TRANSFER_SESSION_FILES, $TBL_FILES
last_client_ip	The client's IP address from the last session of a multiple session transfer.	$TBL_TRANSFERS
last_client_node_id	ID number assigned to the client node during the last session of a multiple session transfer.	$TBL_TRANSFERS
last_dest_ip	The destination's IP address from the last session of a multiple session transfer.	$TBL_TRANSFERS
last_dest_node_id	ID number assigned to the destination node during the last session of a multiple session transfer.	$TBL_TRANSFERS
last_err_desc	Error description from the last session of a multiple session transfer.	$TBL_TRANSFERS, $TBL_FILES

Database Field	Description	Table
last_network_delay	The lag on the network (RTT, measured in milliseconds) from the last session of a multiple session transfer.	$TBL_TRANSFERS
last_restarted_at	The last date/time that the node machine was restarted (for managed node's only).	$TBL_NODES
last_retry_timeout	The number of seconds that the server waited to try again (after a failure), during the last session of a multiple transfer session.	$TBL_TRANSFERS
last_server_ip	The server's IP address from the last session of a multiple session transfer.	$TBL_TRANSFERS
last_server_node_id	ID number assigned to the server node during the last session of a multiple session transfer.	$TBL_TRANSFERS
last_source_ip	The source's IP address from the last session of a multiple session transfer.	$TBL_TRANSFERS
last_source_node_id	ID number assigned to the source node during the last session of a multiple session transfer.	$TBL_TRANSFERS
last_transfer_session_file_id	ID number assigned to the file during the last transfer session.	$TBL_FILES
last_transport	Transport mechanism ("fasp2" for Aspera protocol, "http" for fallback protocol) from the last session of a multiple session transfer.	$TBL_TRANSFERS
max_rate	Maximum transfer rate.	$TBL_TRANSFERS, $TBL_TRANSFER_SESSIONS
min_rate	Minimum transfer rate.	$TBL_TRANSFERS, $TBL_TRANSFER_SESSIONS
mkdirs_attempted	Number of directories that were attempted to be created at the destination.	$TBL_TRANSFER_SESSIONS
mkdirs_failed	Number of directories that failed to be created at the destination.	$TBL_TRANSFER_SESSIONS
mkdirs_passed	Number of directories that were created successfully at the destination.	$TBL_TRANSFER_SESSIONS
name cf	Human-readable name assigned to a transfer. This name may have been keyed in by the user or automatically set by an application.	$TBL_NODES, $TBL_TRANSFERS, $TBL_TRANSFER_SESSIONS

Database Field	Description	Table
network_delay	Lag on the network (RTT), which is measured in milliseconds.	$TBL_TRANSFER_SESSIONS
operation	Either upload or download from the perspective of the client (the initiator). Upload if pushing; download if pulling.	$TBL_TRANSFERS, $TBL_TRANSFER_SESSIONS
os	The node machine's Operating System.	$TBL_NODES
os_version	The version of the node machine's Operating System.	$TBL_NODES
paths_attempted	The total number of files and directories attempted.	$TBL_TRANSFER_SESSIONS
paths_excluded	The number of files and directories that were not transferred because of an exclusion rules.	$TBL_TRANSFER_SESSIONS
paths_failed	The number of files and directories that failed to transfer. A failure is counted if the sender was unable to read a source file or the destination was unable to write the file.	$TBL_TRANSFER_SESSIONS
paths_irreg	This is the total number of special files (for example, nodes, pipes, memory mapped files, page files or /proc files). These files are never transferred.	$TBL_TRANSFER_SESSIONS
pct_complete	Percent (%) of transfer that has been completed. NULL if the node is server is not configured to pre-calculate job size.	$TBL_TRANSFERS, $TBL_TRANSFER_SESSIONS
pretransfer_stats_changed	Between one attempt to the next (retries and sync), whether or not the size of the transfer has changed (grew or reduced in size).	$TBL_TRANSFERS
primary_address	The node's actual IP address (which has been keyed into the Console interface).	$TBL_NODES
priority	Normal or high (only valid when the policy if adaptive).	$TBL_TRANSFERS, $TBL_TRANSFER_SESSIONS
reported_by_both_sides	Database logger added information to the Console from both ends of the transfer (both source and destination are managed nodes and both are sending data back to the database).	$TBL_TRANSFERS
reported_by_server	If this field is equal to 1, then Console received data from the server node. If this field is equal	$TBL_TRANSFER_SESSION_FILES, $TBL_FILES

Database Field	Description	Table
	to 0, then the server was not a managed node or failed to log.	
reported_policy	High, fixed, adaptive or trickle.	$TBL_TRANSFER_SESSIONS
reported_priority	Normal or high (only valid when the policy if adaptive).	$TBL_TRANSFER_SESSIONS
retry_timeout	After a transfer fails, the number of seconds the server will wait before trying again.	$TBL_TRANSFER_SESSIONS
seconds_remaining	Seconds remaining for the file transfer.	$TBL_TRANSFERS, $TBL_TRANSFER_SESSIONS
server_console_ip	Internal IP address of the server (inputted into the nodes page inside Console). Note that this field is primarily used for testing.	$TBL_TRANSFER_SESSIONS
server_err_code	Error code reported by the server.	$TBL_TRANSFER_SESSION_FILES, $TBL_TRANSFER_SESSIONS
server_err_desc	Error code description reported by the server.	$TBL_TRANSFER_SESSION_FILES, $TBL_TRANSFER_SESSIONS
server_external_fasp_port	External fasp (UDP) port of the server. Note that this field is primarily used for testing.	$TBL_TRANSFER_SESSIONS
server_external_ip	External IP address of the server. Note that this field is primarily used for testing.	$TBL_TRANSFER_SESSIONS
server_file_basename	File's basename from server's point-of-view.	$TBL_TRANSFER_SESSION_FILES, $TBL_FILES
server_file_extension	File's extension from server's point-of-view.	$TBL_TRANSFER_SESSION_FILES, $TBL_FILES
server_file_fullpath	File's full path from the server's point-of-view.	$TBL_TRANSFER_SESSION_FILES, $TBL_FILES
server_file_index	Arbitrary, unique number assigned to each file within a transfer session (on the server)	$TBL_TRANSFER_SESSION_FILES
server_ip	IP address of transfer server.	$TBL_TRANSFER_SESSIONS
server_node_id	ID assigned to the server node.	$TBL_TRANSFER_SESSIONS
server_node_uuid	Universally, unique ID assigned to the server node.	$TBL_TRANSFER_SESSIONS
server_status	Either the file status (running, completed, error) or the session status reported by the server. Note that in some cases, client and server can see different statuses (for example, canceled versus error).	$TBL_TRANSFER_SESSION_FILES, $TBL_TRANSFER_SESSIONS

Database Field	Description	Table
server_user	SSH account specified when the transfer starts (should always be displayed).	$TBL_TRANSFERS, $TBL_TRANSFER_SESSIONS
session_count	Number of sessions required for the transfer. **Note:** Hot folders can span many sessions.	$TBL_TRANSFERS
session_file_count	Number of sessions required to send a particular file.	$TBL_FILES
session_id	ID assigned to transfer session.	$TBL_TRANSFER_SESSION_FILES, $TBL_TRANSFER_SESSIONS
size	Size of the file in bytes.	$TBL_TRANSFER_SESSION_FILES, $TBL_FILES
size_changed	The change in file size from one transfer attempt to another.	$TBL_FILES
soap_active_sessions	Number of transfer sessions running on the node.	$TBL_NODES
source_endpoint_id	ID assigned to the source endpoint.	$TBL_TRANSFER_SESSIONS
source_file_basename	File's basename from source's point-of-view.	$TBL_FILES
source_file_extension	File's extension from source's point-of-view.	$TBL_FILES
source_file_fullpath	File's full path from the source's point-of-view.	$TBL_FILES
source_ip	IP address of transfer source.	$TBL_TRANSFER_SESSIONS
source_node_id	ID assigned to the source node.	$TBL_TRANSFER_SESSIONS
source_paths	File paths on the source machine.	$TBL_TRANSFERS, $TBL_TRANSFER_SESSIONS
source_paths_changed	Between one transfer attempt to the next, whether or not the file source paths have changed.	$TBL_TRANSFERS
source_user	If upload, source_user is the client user. If download, source_user is server user.	$TBL_TRANSFERS, $TBL_TRANSFER_SESSIONS
ssh_port	Node machine's SSH port.	$TBL_NODES
ssh_tunnel_port	Node machine's SSH tunnel port.	$TBL_NODES
start_byte	Displays the point at which data from the file started transferring to the destination (relevant if some of the file has already been transferred). If the file has already been transferred to the	$TBL_TRANSFER_SESSION_FILES, $TBL_FILES

Database Field	Description	Table
	destination, then the start byte equals the total file size.	
started_at	Date and time that a transfer or file started.	$TBL_TRANSFERS, $TBL_TRANSFER_SESSIONS, $TBL_TRANSFER_SESSION_FILES, $TBL_FILES
started_via	The name of the application (Aspera or custom) that is responsible for initiating the transfer (for example, aspera.scp, aspera.sync, etc.).	$TBL_TRANSFERS, $TBL_TRANSFER_SESSIONS
status	Status of transfer or file (for example, "running," "completed," "canceled" or "error").	$TBL_TRANSFERS, $TBL_TRANSFER_SESSIONS, $TBL_TRANSFER_SESSION_FILES, $TBL_FILES
stopped_at	Date and time that a transfer or file stopped (value is blank if transfer or file is still active).	$TBL_TRANSFERS, $TBL_TRANSFER_SESSIONS, $TBL_TRANSFER_SESSION_FILES, $TBL_FILES
target_rate	Target transfer rate.	$TBL_TRANSFERS, $TBL_TRANSFER_SESSIONS
tmp_actual_rate	Reserved for future use.	$TBL_TRANSFERS, $TBL_TRANSFER_SESSIONS
tmp_actual_rate_calculated_at	Reserved for future use.	$TBL_TRANSFERS, $TBL_TRANSFER_SESSIONS
tmp_loss_pct	Reserved for future use.	$TBL_TRANSFER_SESSIONS
token	Security token used for the transfer (note that this depends on whether or not the application that started the transfer is configured to use tokens).	$TBL_TRANSFERS, $TBL_TRANSFER_SESSIONS
transfer_file_id	Corresponds to the file ID field	$TBL_TRANSFER_SESSION_FILES
transfer_id	Corresponds to ID field.	$TBL_TRANSFER_SESSIONS, $TBL_FILES
transfer_session_id	Corresponds to transfer session ID field	$TBL_TRANSFER_SESSION_FILES
transfer_uuid	Universally, unique ID that is used to identify the transfer as a whole. May contain multiple sessions and is generated by application that started the transfer. Generally only populated by transfers started by Console.	$TBL_TRANSFERS, $TBL_TRANSFER_SESSIONS
transport	fasp2 for Aspera protocol, "http" for fallback protocol	$TBL_TRANSFER_SESSIONS
type	Managed or unmanaged node.	$TBL_NODES

Database Field	Description	Table
udp_port	Node machine's UDP port.	$TBL_NODES
use_ssh_tunnel	Set up an SSH tunnel for database logging (for managed nodes only).	$TBL_NODES
usecs	Length of the transfer session (in milliseconds). Not authoritative (use only for transfer sessions and transfers).	$TBL_TRANSFERS, $TBL_TRANSFER_SESSIONS, $TBL_TRANSFER_SESSION_FILES, $TBL_FILES
uuid	Universally, unique identifier that is generated on the node when installing Aspera software (for managed nodes only)	$TBL_NODES

Important: If you have configured custom fields, they will be prefixed with "cf_". Custom fields are utilized in the $TBL_FILES and $TBL_TRANSFER tables. Please note that if you would like to add additional custom fields, you may do so via the **Configuration** > **Custom Fields**. For instructions on setting up a custom field, see "Creating Custom Fields" on page 81.

Advanced Report Usage Notes

Advanced Report Usage Notes: Avoid Duplicating Identical Records

Console's security filtering prioritizes speed over the cost of potentially returning duplicate records. It is up to the report writer to remove duplicate records returned when querying report tables directly.

For example, a user unaware of Console internals might expect the following to always return no more than a single record:

```
SELECT ts.*
FROM
    $TBL_TRANSFER_SESSIONS ts
WHERE
    ts.session_id='ed0a9b4039bb40dfa86690ff7e1f6fa2'
;
```

However, depending on the user's group memberships and permissions, the above could return multiple identical records. To correct this, use **SELECT DISTINCT**. For example:

```
SELECT DISTINCT ts.*
FROM
    $TBL_TRANSFER_SESSIONS ts
WHERE
    ts.session_id='ed0a9b4039bb40dfa86690ff7e1f6fa2'
;
```

Be aware that this means you cannot directly perform aggregate computations--such as **SUM**, **AVERAGE**, or **COUNT**--on the reporting tables. For example, in the following, **total_bytes_transferred** could count some sessions multiple times:

```
SELECT DISTINCT
    ts.contact
    , SUM(ts.bytes_transferred) AS total_bytes_transferred
FROM
    $TBL_TRANSFER_SESSIONS ts
WHERE
    ...
;
```

Instead, first extract just the data of interest to a temporary table, then summarize from there:

```
# Create holding table for filtered raw data
CREATE TABLE $TMP_FILTERED_TRANSFER_SESSIONS (
    `id` INT(11) NOT NULL AUTO_INCREMENT PRIMARY KEY
    , `contact` VARCHAR(255)
    , `bytes_transferred` BIGINT(20)
    );

# Extract relevant data (very important to include ts.id)
INSERT INTO $TMP_FILTERED_TRANSFER_SESSIONS
SELECT DISTINCT
    ts.id
    , ts.contact
    , ts.bytes_transferred
FROM $TBL_TRANSFER_SESSIONS ts
WHERE
    ts.started_at < '$REPORT_PERIOD_END'
    AND (
        ts.stopped_at >= '$REPORT_PERIOD_START'
        OR ts.stopped_at IS NULL
    );

# Summarize by contact
CREATE TABLE $FINAL_RESULT_TABLE
SELECT
    fts.contact
    , SUM(fts.bytes_transferred) AS total_bytes_transferred
FROM
    $TMP_FILTERED_TRANSFER_SESSIONS fts
GROUP BY
    fts.contact
ORDER BY
    fts.contact
;
```

Advanced Report Usage Notes: Avoid Duplicating Redundant Records

Transfers between two managed nodes create two records per file, one in **$TBL_TRANSFER_SESSION_FILES** and one in **$TBL_FILES**.

If both source and destination are managed nodes, then both sides log to the database. These records will not be identical--the record logged by the server reports the server-side path, while the record logged by the client reports the client-side path. Sometimes other fields, such as **err_desc**, may differ as well.

There are several fields in the canonical tables supplied specifically to address this issue:

Field Name	Description	Tables
reported_by_both_sides	0 if transfer was only logged by one side. 1 if transfer was logged by both server and client.	**$TBL_TRANSFERS**
reported_by_server	0 if the file record was logged by the client. 1 if the file record was logged by the server.	**$TBL_FILES** **$TBL_TRANSFER_SESSION_FILES**
initiated_by_source	0 if transfer is a pull (client is the destination). 1 if transfer is a push (client is the source).	**$TBL_TRANSFERS** **$TBL_TRANSFER_SESSIONS $TBL_FILES** **$TBL_TRANSFER_SESSION_FILES**

To ensure that each file is present only once in a result set, we need to use the above fields to give precedence to the record from one side or the other.

Note: The previous caveat about record duplication ("Advanced Report Usage Notes: Avoid Duplicating Identical Records" on page 166) also applies (i.e. the file record reported by the server node could itself be returned multiple times, as well as the record reported by the client node).

Note: Certain edge cases cause a problem even when using the above filter. For example, if both nodes start reporting a transfer session and one node loses its connection to the database, then **reported_by_both_sides** will equal 1, but not all of the file records will have two records in the file tables.

The following SQL example, taken from the built-in **Activity Summary By Contact** report, gives the *destination-side* file record precedence in cases where both sides logged the transfer.

```
#==================================================
# Set variables to hold report datetime parameters
# (all datetimes are converted to UTC)
#==================================================
SET @report_period_start = '$REPORT_PERIOD_START';
SET @report_period_end = '$REPORT_PERIOD_END';
#===================================================
# PRE-FILTER RECORD IDS
# Initially retrieve just the id columns from
# base tables (improves performance by avoiding
# queries with more than one join)
#===================================================
#---------------------------------------------------
# Create tables to hold the prefiltered record IDs
#---------------------------------------------------
CREATE TABLE $TMP_TRANSFER_IDS (
    id INT NOT NULL PRIMARY KEY
    , reported_by_both_sides TINYINT(1) NOT NULL DEFAULT 0
    );
CREATE TABLE $TMP_TRANSFER_SESSION_IDS (
    id INT NOT NULL PRIMARY KEY
    , reported_by_both_sides TINYINT(1) NOT NULL DEFAULT 0
    );
CREATE TABLE $TMP_FILE_SESSION_IDS (
    id INT NOT NULL PRIMARY KEY
    , transfer_session_id INT NOT NULL
    );
#---------------------------------------------------
# Retrieve IDs
#---------------------------------------------------
#---------------------------------------------------
# Transfers
#---------------------------------------------------
INSERT INTO $TMP_TRANSFER_IDS
SELECT DISTINCT
    t.id
    , t.reported_by_both_sides
FROM $TBL_TRANSFERS t
WHERE
    (t.started_at < @report_period_end
        AND (t.stopped_at >= @report_period_start
            OR t.stopped_at IS NULL
        )
    )
;
#---------------------------------------------------
# Transfer Sessions
# (copy over 'reported_by_both_sides'
# from transfers)
#---------------------------------------------------

INSERT INTO $TMP_TRANSFER_SESSION_IDS
SELECT DISTINCT
    ts.id
    , t.reported_by_both_sides
FROM $TBL_TRANSFER_SESSIONS ts
    JOIN $TMP_TRANSFER_IDS t
        ON ts.transfer_id = t.id
WHERE
    (ts.started_at < @report_period_end
        AND (ts.stopped_at >= @report_period_start
            OR ts.stopped_at IS NULL
        )
    )
;
#---------------------------------------------------
```

```
# File Sessions (choose destination-side
# info if both sides logged to db)
#-----------------------------------------------------
INSERT INTO $TMP_FILE_SESSION_IDS
SELECT DISTINCT
    fs.id
    , fs.transfer_session_id
FROM $TBL_TRANSFER_SESSION_FILES fs
    JOIN $TMP_TRANSFER_SESSION_IDS ts
        ON fs.transfer_session_id = ts.id
WHERE
    (fs.started_at < @report_period_end
        AND (fs.stopped_at >= @report_period_start
            OR fs.stopped_at IS NULL)
        )
        AND (ts.reported_by_both_sides=0
            OR (
                (fs.reported_by_server=1
                    AND fs.initiated_by_source=1)
            OR (fs.reported_by_server=0
                AND fs.initiated_by_source=0)
        )
    )
;
CREATE INDEX idx_transfer_session_id
    ON $TMP_FILE_SESSION_IDS (transfer_session_id);
```

Advanced Report Usage Notes: Filter on Raw Values

Filtering on computed values in most cases prevents MySQL from being able to take advantage of indexes. For example, the following will force a scan of every record in **TBL_TRANSFER_SESSIONS**, because MySQL has to perform the **CONVERT( )** on **ts.started_at** for every record:

```
SELECT DISTINCT ts.*
FROM
    $TBL_TRANSFER_SESSIONS ts
WHERE
    CONVERT(ts.started_at, DATE) = DATE(NOW())
;
```

Instead, compute the correct criteria to compare the raw value against:

```
SET @todays_date = DATE(NOW());
SET @tomorrows_date = DATE_ADD(@todays_date, INTERVAL 1 DAY);
SELECT DISTINCT ts.*
FROM
    $TBL_TRANSFER_SESSIONS ts
WHERE
    ts.started_at >= @todays_date
    AND ts.started_at < @tomorrows_date
;
```

Note: Even the above will only give expected results if you are in GMT time zone, as **NOW( )** will return UTC time.

The builtin report variables **$REPORT_PERIOD_START** and **$REPORT_PERIOD_END** contain datetimes converted from local time zone of input into UTC and are usually a better choice for date filtering (unless recipient is fine with UTC-based filtering).

Advanced Report Usage Notes: Filter Strings by Using "Begins With"

If possible, filter strings by matching “begins with” rather than “contains” or “ends with”. If that's not possible, consider creating a custom field. For example, the following will not be able to use the index on **ts.contact**:

```
SELECT DISTINCT ts.*
FROM
    $TBL_TRANSFER_SESSIONS ts
WHERE
    ts.contact LIKE '%Euro2012_Livex%'
;
```

If you know all the possible ways the string could begin, you could enumerate them like this:

```
SELECT DISTINCT ts.*
FROM
    $TBL_TRANSFER_SESSIONS ts
WHERE
    ts.contact LIKE 'AA_Euro2012_Livex%'
    OR ts.contact LIKE 'BB_Euro2012_Livex%'
    OR ts.contact LIKE 'CC_Euro2012_Livex%'
;
```

If the report is only to be run for a small date range and there are few transfer sessions then you may not need to worry about this. If you expect to be running over large date ranges and large numbers of sessions, then you should create a custom field that detects the presence of the match string and then copies it to the custom field -- you could then filter on the custom field instead.

Advanced Report Usage Notes: Always Include a Date Filter

To avoid creating a report that might try to work on the entire database you should always include some kind of date filter.

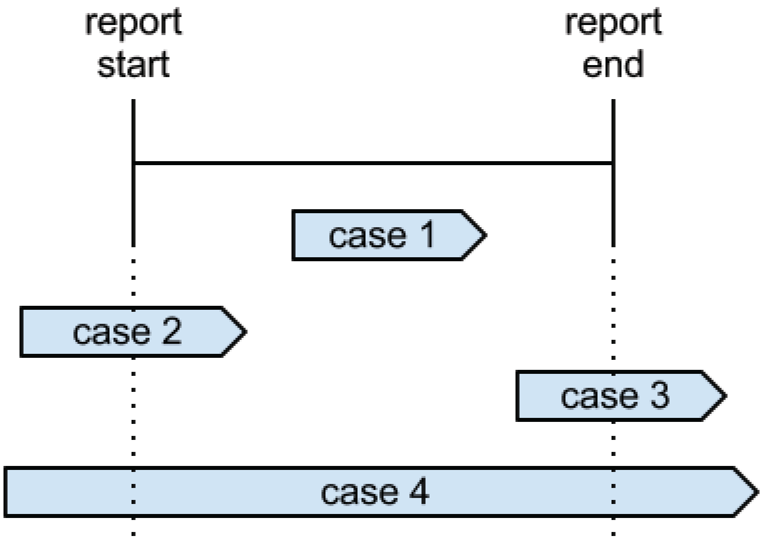
The recommended option is to use the built-in report variables **$REPORT_PERIOD_START** and **$REPORT_PERIOD_END** to filter data. If an advanced report contains these variables, the web UI will include date pickers when the user runs the report. A standard filter to find all transfers that were active at any time during the report period would look like the following:

```
SELECT DISTINCT
    ts.id
    , ts.contact
    , ts.bytes_transferred

FROM
    $TBL_TRANSFER_SESSIONS ts

WHERE
    ts.started_at < '$REPORT_PERIOD_END'
    AND (
        ts.stopped_at >= '$REPORT_PERIOD_START'
        OR ts.stopped_at IS NULL
    );
```

Note the clause **`OR ts.stopped_at IS NULL`**. Without this, the report would exclude any transfers that were still running at the time the report was run. Depending on the intended purpose of the report, you might need to prorate data for transfers that were active for only part of the reporting period, such as cases 2, 3, and 4 in the following:



As an alternative, you can avoid the use of **$REPORT_PERIOD_START** and **$REPORT_PERIOD_END** if you are creating a report that always looks at the last X hours:

```
SET @min_start = DATE_SUB(NOW(), INTERVAL 24 HOUR);
CREATE TABLE $FINAL_RESULT_TABLE
SELECT DISTINCT ts.*
FROM
    $TBL_TRANSFER_SESSIONS ts
WHERE
    ts.started_at >= @min_start
;
```

Note: As of Console 1.6 there is a bug in the report engine that causes the creation of Excel/CSV files to fail if you do not reference **$REPORT_PERIOD_START** and **$REPORT_PERIOD_END** at all. To work around this, include a dummy reference in the report SQL. For example:

```
SET @dummy = '$REPORT_PERIOD_START';
```

When running the report, users are then asked for report period dates, but they will be ignored.

Advanced Report Usage Notes: Always Name Your Computed or Aggregated Columns

Always name your computed or aggregate columns, and avoid names that might be reserved words. In particular, do not call a final result column "name", "count", "id", and so on.

INCORRECT:

```
CREATE TABLE $FINAL_RESULT_TABLE
SELECT
    fts.contact
    , COUNT(*)
    , SUM(fts.bytes_transferred)
...
```

CORRECT:

```
CREATE TABLE $FINAL_RESULT_TABLE
SELECT
    fts.contact
    , COUNT(*) AS session_count
    , SUM(fts.bytes_transferred) AS total_bytes_transferred
...
```

Advanced Report Usage Notes: Avoid Joining Reporting Views

MySQL often mis-optimizes queries that join reporting views directly to each other. The fact that the views can show the same record multiple times can cause a geometric explosion in the number of temporary records inspected.

EXAMPLE:

```
# Find all sessions that contained file "foo.txt"
# List both session info and file info
# ASSUMES NO SESSIONS WERE BETWEEN TWO MANAGED NODES
SET @report_period_start = '$REPORT_PERIOD_START';
SET @report_period_end = '$REPORT_PERIOD_END';
CREATE TABLE $FINAL_RESULT_TABLE
SELECT DISTINCT
    ts.session_id
    , ts.source_ip
    , ts.dest_ip
    , ts.started_at
    , ts.stopped_at
    , ts.status
    , tsf.file_fullpath
    , tsf.size
    , tsf.started_at AS file_started_at
    , tsf.stopped_at AS file_stopped_at
    , tsf.status AS file_status
FROM
    $TBL_TRANSFER_SESSIONS ts
    JOIN $TBL_TRANSFER_SESSION_FILES tsf
        ON ts.id = tsf.transfer_session_id
WHERE
    tsf.started_at < @report_period_end
    AND (
        tsf.stopped_at >= @report_period_start
        OR tsf.stopped_at IS NULL
    )
    AND tsf.file_basename = "foo.txt"
ORDER BY
    ts.started_at
    , tsf.started_at
;
```

Although the above report uses **SELECT DISTINCT**, contains no aggregate functions such as **COUNT** and **SUM**, and generates a correct final result (unless any of the transfer sessions were between two managed nodes), it is potentially slow. For greater speed (and to prevent query misoptimization from MySQL), it is better to decompose the above query into smaller steps, and join your temporary tables to the report views instead of joining the report views together directly.

Note: In order to avoid complexity in the SQL, the example below assumes no sessions were between two managed nodes. Therefore, the code for dealing with this has been left out (see “Advanced Report Usage Notes: Avoid Duplicating Redundant Records” on page 167).

EXAMPLE

```
SET @report_period_start = '$REPORT_PERIOD_START';
SET @report_period_end = '$REPORT_PERIOD_END';

#-------------------------------------------------
# Create tables to prefilter base table record ids
#-------------------------------------------------

CREATE TABLE $TMP_TRANSFER_SESSION_IDS (
    id INT NOT NULL PRIMARY KEY
    );
CREATE TABLE $TMP_TRANSFER_SESSION_FILE_IDS (
    id INT NOT NULL PRIMARY KEY
    , transfer_session_id INT NOT NULL
    );

#-------------------------------------------------
# Create table to hold all desired fields from
# transfer_sessions
#-------------------------------------------------

CREATE TABLE $TMP_TRANSFER_SESSION_DATA (
    `id` INT(11) NOT NULL AUTO_INCREMENT PRIMARY KEY
    , `session_id` VARCHAR(36)
    , `source_ip` VARCHAR(255)
    , `dest_ip` VARCHAR(255)
    , `started_at` DATETIME
    , `stopped_at` DATETIME
    , `status` VARCHAR(255)
    );

#-------------------------------------------------
# Create table to hold all desired fields from
# transfer_session_files
#-------------------------------------------------

CREATE TABLE $TMP_TRANSFER_SESSION_FILE_DATA (
`id` INT(11) NOT NULL AUTO_INCREMENT PRIMARY KEY
    , `transfer_session_id` INT(11)
    , `started_at` DATETIME
    , `stopped_at` DATETIME
    , `status` VARCHAR(255)
    , `file_fullpath` TEXT
    , `size` BIGINT(20)
    );
```

```
#========================================
# PRE-FILTER BASE TABLE IDS
#========================================

#------------------------------------------------
# For this report, we know we are
# filtering on file name and can use
# the index on that column, so it is
# faster to find the records from
# transfer_session_files first
#------------------------------------------------

#------------------------------------------------
# Transfer Session Files
#------------------------------------------------

INSERT INTO $TMP_TRANSFER_SESSION_FILE_IDS
SELECT DISTINCT
    tsf.id
    , tsf.transfer_session_id
FROM $TBL_TRANSFER_SESSION_FILES tsf
WHERE
    tsf.started_at < @report_period_end
    AND (
        tsf.stopped_at >= @report_period_start
        OR tsf.stopped_at IS NULL
    )
    AND tsf.file_basename = "foo.txt"
;

#------------------------------------------------
# Create an index on the join field -
# for speed, we wait until table is
# populated instead of defining the index
# during initial creation of table
#------------------------------------------------

CREATE INDEX idx_transfer_session_id ON
$TMP_TRANSFER_SESSION_FILE_IDS (transfer_session_id);

#-------------------
# Transfer Sessions
#-------------------

INSERT INTO $TMP_TRANSFER_SESSION_IDS
SELECT DISTINCT ts.id
FROM $TBL_TRANSFER_SESSIONS ts
    JOIN $TMP_TRANSFER_SESSION_FILE_IDS tsf
        ON ts.id = tsf.transfer_session_id
WHERE
    ts.started_at < @report_period_end
    AND (
        ts.stopped_at >= @report_period_start
        OR ts.stopped_at IS NULL
    )
;

#------------------------------------------------
# Remove transfer file sessions that don't have an
# associated transfer_session record
# (normally not supposed to happen, but we want
# to protect against bad data that might be
# caused by system crash, logger errors, console
# purge errors, Canonicalizer shutdown, etc.)
# For this particular report, this is not needed
# since the final join will weed out such records,
# but it is a good habit to maintain, this report
# could be modified later into one where
# it would make a difference.
#------------------------------------------------

DELETE tsf.*
FROM $TMP_TRANSFER_SESSION_FILE_IDS tsf
    LEFT JOIN $TMP_TRANSFER_SESSION_IDS ts
        ON tsf.transfer_session_id = ts.id
WHERE ts.id IS NULL;

#========================
# Get all desired fields
#========================
#------------------------
# transfer_session_files
#------------------------

INSERT INTO $TMP_TRANSFER_SESSION_FILE_DATA (
    id
    , `transfer_session_id`
    , `started_at`
    , `stopped_at`
    , `status`
    , `file_fullpath`
    , `size`
    )

SELECT DISTINCT
    tsf.id
    , tsf.transfer_session_id
    , tsf.started_at
    , tsf.stopped_at
    , tsf.status
    , tsf.file_fullpath
    , tsf.size
FROM
    $TMP_TRANSFER_SESSION_FILE_IDS tsf_ids
    STRAIGHT_JOIN $TBL_TRANSFER_SESSION_FILES tsf
        ON tsf_ids.id = tsf.id
;

#-------------------------
# Add index to speed joins
#-------------------------

CREATE INDEX idx_transfer_session_id ON
$TMP_TRANSFER_SESSION_FILE_DATA (`transfer_session_id`);

#-------------------
# transfer_sessions
#-------------------

INSERT INTO $TMP_TRANSFER_SESSION_DATA (
    id
    , `session_id`
    , `source_ip`
    , `dest_ip`
    , `started_at`
    , `stopped_at`
    , `status`
    )
SELECT DISTINCT
    ts.id
    , ts.session_id
    , ts.source_ip
    , ts.dest_ip
    , ts.started_at
    , ts.stopped_at
    , ts.status
FROM
    $TMP_TRANSFER_SESSION_IDS ts_ids
    STRAIGHT_JOIN $TBL_TRANSFER_SESSIONS ts
        ON ts_ids.id = ts.id
;

#==========================
# Create final result table
#==========================

CREATE TABLE $FINAL_RESULT_TABLE
SELECT
    ts.session_id
    , ts.source_ip
    , ts.dest_ip
    , ts.started_at
    , ts.stopped_at
    , ts.status
    , tsf.file_fullpath
    , tsf.size
    , tsf.started_at AS file_started_at
    , tsf.stopped_at AS file_stopped_at
    , tsf.status AS file_status
FROM
    $TMP_TRANSFER_SESSION_DATA ts
    JOIN $TMP_TRANSFER_SESSION_FILE_DATA tsf
```

```
        ON ts.id = tsf.transfer_session_id
ORDER BY
    ts.started_at
    , tsf.started_at
;
```

Example Reports

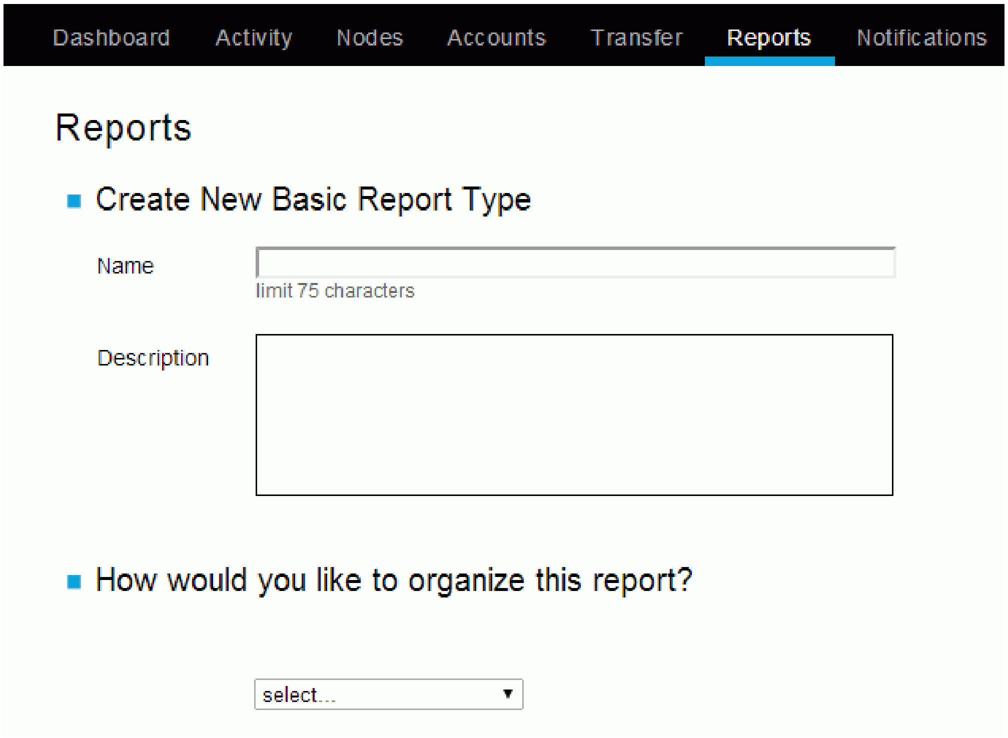
Basic Report Example: Faspex User Activity

The following example demonstrates the process of creating a new, **basic** report (following the instructions described in "Creating a Basic Report" on page 76) for Faspex users. In our example, we will generate a report that displays transfer activity by *Faspex users* only. The example report, once generated, will display total bytes transferred by each Faspex Server user, along with file and transfer-level detail (where a **transfer** groups together transfer sessions into a single item).

1. Go to the *Manage Report Types* page.

   Select **Reports** from the Console menu, and then click the **Manage Report Types** button. On the *Manage Report Types* screen, click the **New Basic** button.
2. Configure your basic report to display file- and transfer-level details, organized by Faspex Users.

   On the *Create New Report Type* page (for **basic** reports), enter the following information:



Field	Description
Name	Basic Faspex User Report
Description	Basic Faspex Server User report, which includes total bytes per Faspex User, as well as file- and transfer-level details.
How would you like to organize this report?	Select "Contact," "Transfer," and "File" as the fields by which to organize this report. In doing so, the report will be grouped by the following fields: • **Contact** (Contact assigned by Console. This can be a Console user name, a Faspex Server user name, SSH account, or customized value obtained

Field	Description
	from a transfer cookie. Examples: "admin (console)", "aspera (ssh)", "michael (faspex)".) • **Transfer** (Human-readable name assigned to a transfer. A transfer represents one or multiple executions of *ascp* (that is, one or multiple attempts to transfer).) • **File** (File's name)
Columns to include	Select the following basic fields to include as columns: • bytes transferred • average rate • files completed • files failed • started at • stopped at • status • error description • source address • destination address **Note:** When you select a field, its definition will appear in the box below.
Sort	Select the following fields to sort data inside your groups: • Sort your contact groups by contact name. • Sort your transfer groups by the time that the transfer started. • Sort your file groups by file name. Select ascending order for all fields.
Filters	To narrow down the report so that only Faspex Users are displayed, specify the *Transfer Contact* field as ending with the value *(faspex)*.

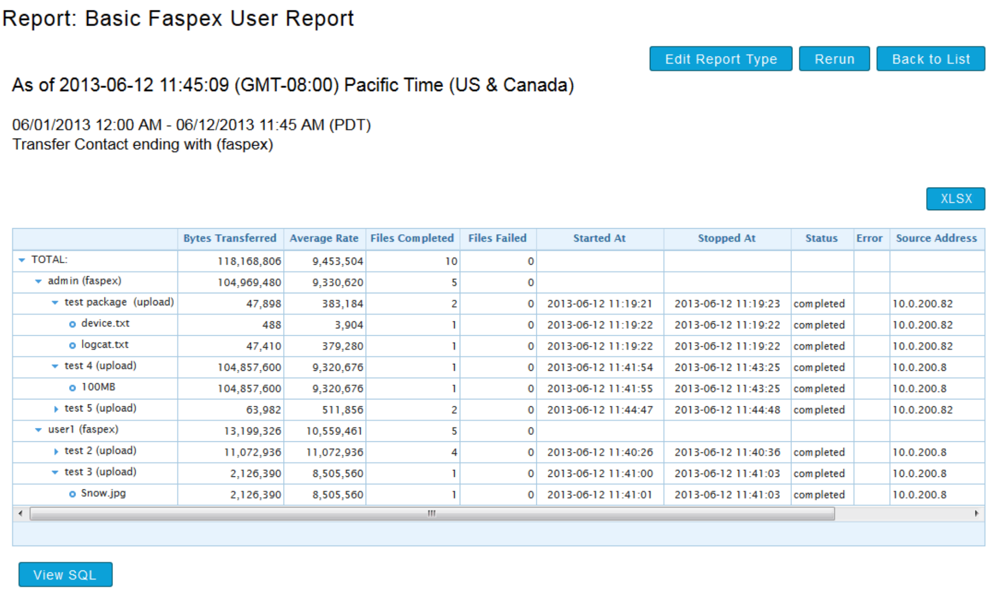
3. Save, finalize run settings, and run your report.

   Next, click the **Create and Run** button. Confirm the following settings on next page:

   - Title is as described above.
   - Report is scheduled to *Run Now*.
   - Report period is *Month to date* and time zone is *Pacific*.
   - Sorting is as described above.
   - Filter is as described above.
   - XLSX file format is checked.

   Once confirmed, click the **Run Report** button.
4. View your Web and XLSX reports.

   After clicking the **Run Report** button, the page updates to display the report queuing and then running. Once generated, the Web version of your basic report appears as shown below.



Report: Basic Faspex User Report

As of 2013-06-12 11:45:09 (GMT-08:00) Pacific Time (US & Canada)

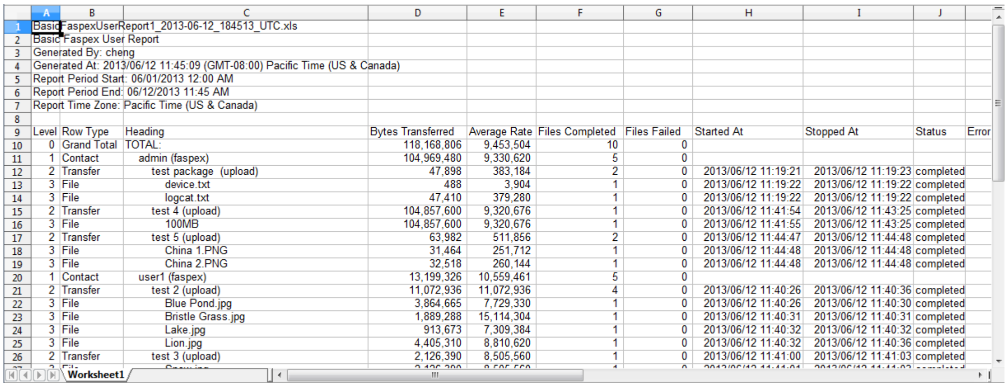
06/01/2013 12:00 AM - 06/12/2013 11:45 AM (PDT)
Transfer Contact ending with (faspex)

	Bytes Transferred	Average Rate	Files Completed	Files Failed	Started At	Stopped At	Status	Error	Source Address
TOTAL:	118,168,806	9,453,504	10	0					
admin (faspex)	104,969,480	9,330,620	5	0					
test package (upload)	47,898	383,184	2	0	2013-06-12 11:19:21	2013-06-12 11:19:23	completed		10.0.200.82
device.txt	488	3,904	1	0	2013-06-12 11:19:22	2013-06-12 11:19:22	completed		10.0.200.82
logcat.txt	47,410	379,280	1	0	2013-06-12 11:19:22	2013-06-12 11:19:22	completed		10.0.200.82
test 4 (upload)	104,857,600	9,320,676	1	0	2013-06-12 11:41:54	2013-06-12 11:43:25	completed		10.0.200.8
100MB	104,857,600	9,320,676	1	0	2013-06-12 11:41:55	2013-06-12 11:43:25	completed		10.0.200.8
test 5 (upload)	63,982	511,856	2	0	2013-06-12 11:44:47	2013-06-12 11:44:48	completed		10.0.200.82
user1 (faspex)	13,199,326	10,559,461	5	0					
test 2 (upload)	11,072,936	11,072,936	4	0	2013-06-12 11:40:26	2013-06-12 11:40:36	completed		10.0.200.8
test 3 (upload)	2,126,390	8,505,560	1	0	2013-06-12 11:41:00	2013-06-12 11:41:03	completed		10.0.200.8
Snow.jpg	2,126,390	8,505,560	1	0	2013-06-12 11:41:01	2013-06-12 11:41:03	completed		10.0.200.8

As you can see, the report's data is grouped and sorted in the following manner:

- Faspex Users
- Transfers (per Faspex User), which are sorted by the time they started
- File name (per Transfer)

In addition, all data columns appear as selected on the *Create Advanced Report Type* page. To download the Excel version of the report for use in other applications, click the **XLSX** button.



BasicFaspexUserReport1_2013-06-12_184513_UTC.xls
Basic Faspex User Report
Generated By: cheng
Generated At: 2013/06/12 11:45:09 (GMT-08:00) Pacific Time (US & Canada)
Report Period Start: 06/01/2013 12:00 AM
Report Period End: 06/12/2013 11:45 AM
Report Time Zone: Pacific Time (US & Canada)

Level	Row Type	Heading	Bytes Transferred	Average Rate	Files Completed	Files Failed	Started At	Stopped At	Status	Error
0	Grand Total	TOTAL:	118,168,806	9,453,504	10	0				
1	Contact	admin (faspex)	104,969,480	9,330,620	5	0				
2	Transfer	test package (upload)	47,898	383,184	2	0	2013/06/12 11:19:21	2013/06/12 11:19:23	completed	
3	File	device.txt	488	3,904	1	0	2013/06/12 11:19:22	2013/06/12 11:19:22	completed	
3	File	logcat.txt	47,410	379,280	1	0	2013/06/12 11:19:22	2013/06/12 11:19:22	completed	
2	Transfer	test 4 (upload)	104,857,600	9,320,676	1	0	2013/06/12 11:41:54	2013/06/12 11:43:25	completed	
3	File	100MB	104,857,600	9,320,676	1	0	2013/06/12 11:41:55	2013/06/12 11:43:25	completed	
2	Transfer	test 5 (upload)	63,982	511,856	2	0	2013/06/12 11:44:47	2013/06/12 11:44:48	completed	
3	File	China 1.PNG	31,464	251,712	1	0	2013/06/12 11:44:48	2013/06/12 11:44:48	completed	
3	File	China 2.PNG	32,518	260,144	1	0	2013/06/12 11:44:48	2013/06/12 11:44:48	completed	
1	Contact	user1 (faspex)	13,199,326	10,559,461	5	0				
2	Transfer	test 2 (upload)	11,072,936	11,072,936	4	0	2013/06/12 11:40:26	2013/06/12 11:40:36	completed	
3	File	Blue Pond.jpg	3,864,665	7,729,330	1	0	2013/06/12 11:40:26	2013/06/12 11:40:30	completed	
3	File	Bristle Grass.jpg	1,889,288	15,114,304	1	0	2013/06/12 11:40:31	2013/06/12 11:40:31	completed	
3	File	Lake.jpg	913,673	7,309,384	1	0	2013/06/12 11:40:32	2013/06/12 11:40:32	completed	
3	File	Lion.jpg	4,405,310	8,810,620	1	0	2013/06/12 11:40:32	2013/06/12 11:40:36	completed	
2	Transfer	test 3 (upload)	2,126,390	8,505,560	1	0	2013/06/12 11:41:00	2013/06/12 11:41:03	completed	

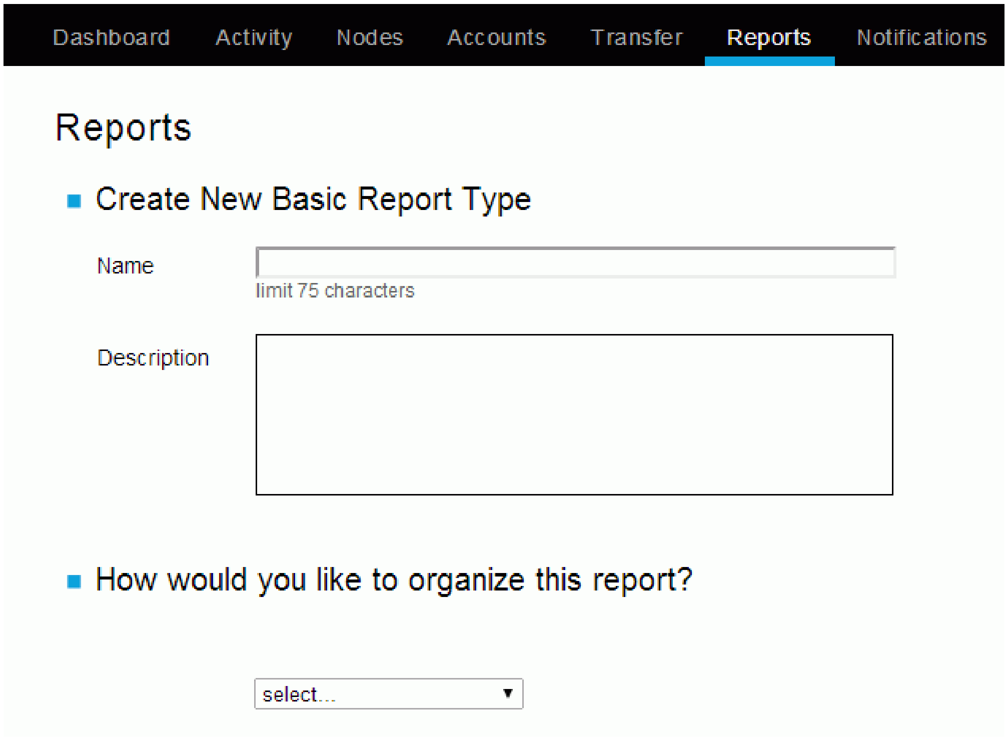
Basic Report Example: Hot Folder Activity

The following example demonstrates the process of creating a new, **basic** report (following the instructions described in the topic "Creating a Basic Report" on page 76) for Hot Folder transfers. In our example, we will generate a report that displays transfer activity for *Hot Folders* that have been set up within HST Server, Point_to_Point and Client. The example report, once generated, will display Hot Folder transfer start time, end time and the number of files transferred.

1. Go to the *Manage Report Types* page

   Select **Reports** from the Console menu, and then click the **Manage Report Types** button. On the *Manage Report Types* screen, click the **New Basic** button.
2. Configure your basic report to display file- and transfer-level details, organized by Faspex Users

   On the *Create New Report Type* page (for **basic** reports), enter the following information:



Field	Description
Name	Basic Hot Folder Transfer Report
Description	Basic Hot Folder Transfer report, which includes start time, stop time and the number of files that were transferred.
How would you like to organize this report?	Select "Transfer" as the field by which to organize this report. In doing so, the report will be grouped the human-readable name that has been assigned to each hot folder transfer. A transfer represents one or multiple executions of *ascp* (that is., one or multiple attempts to transfer).
Columns to include	Select the following basic fields to include as columns: • started at • stopped at • files completed • average rate **Note:** When you select a field, its definition appears in the box below.
Sort	Select the "transfer name" field (in ascending order) to sort data inside your group.
Filters	In this example, we must set a filter that checks the value of the transfer cookie. When files are transferred using Hot Folders, the transfer cookie contains the following information: `aspera.sync2:` Thus, the filter must be set to only include transfers that have a transfer cookie starting with the value *aspera.sync2:*.

3. Save, finalize run settings, and run your report.

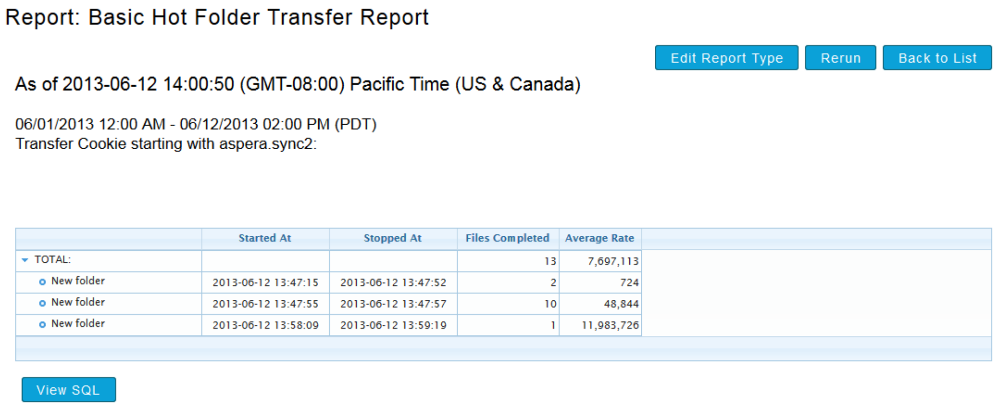
   Next, click the **Create and Run** button. Confirm the following settings on next page:

- Title is as described above
- Report is scheduled to *Run Now*
- Report period is *Month to date* and time zone is *Pacific*
- Sorting is as described above
- Filter is as described above

Once confirmed, click the **Run Report** button.

4. View your Web report.

   After clicking the **Run Report** button, the page will update to display the report queuing and then running. Once generated, the Web version of your basic report appears as shown below.



Report: Basic Hot Folder Transfer Report

As of 2013-06-12 14:00:50 (GMT-08:00) Pacific Time (US & Canada)

06/01/2013 12:00 AM - 06/12/2013 02:00 PM (PDT)
Transfer Cookie starting with aspera.sync2:

	Started At	Stopped At	Files Completed	Average Rate
TOTAL:			13	7,697,113
New folder	2013-06-12 13:47:15	2013-06-12 13:47:52	2	724
New folder	2013-06-12 13:47:55	2013-06-12 13:47:57	10	48,844
New folder	2013-06-12 13:58:09	2013-06-12 13:59:19	1	11,983,726

Basic Report Example: Faspex Metadata

The following example demonstrates the process of creating a new, **basic** report (following the instructions described in the topic "Creating a Basic Report" on page 76) for Faspex metadata. In our example, we will generate a report that displays the metadata that is entered into a "Create New Package" form within Faspex, which is accomplished by creating a new, custom field called "Event" within Console.

Note: This example assumes that the "event" (metadata) field has already been set up on the Faspex node. When creating a new Faspex package, Faspex users can select from a predefined (drop-down) list of events, which populates the database for this custom field.

The example report, once generated, will display the purpose (or "Event") of the Faspex package, as well as file-level detail, transfer-level detail (where a **transfer** groups together transfer sessions into a single item), and which Faspex user sent the package.

1. Set up a Console database custom field for the metadata.

   Within Console, select **Configuration** from the main menu, and then the **Custom Fields** tab. Create a new, custom field with the following attributes:

   - **Level**: Select "transfer"
   - **Name**: Enter the name "event"
   - **Start Date**: Enter "2011-01-01"
   - **Description**: Since this custom field is for the metadata report, enter the description "Faspex Metadata report demo"

   For more information on custom fields, see "Creating Custom Fields" on page 81.
2. On the next page, click the **Back to Custom Fields** tab or the **Custom Fields** tab. Locate the entry for the field you just created ("event" in this case), and click **recalculate**
3. Create a custom rule for the **Faspex Metadata** transfer field.

   - **Built-in field**: `Faspex Metadata`
   - **Operator**: `JSON match`

- **Expression**: `\{.*"Event":"(?<event>.+?)".*\)`
- **"faspex_event" custom field value**: `<event>`.

In other words, if the Faspex Metadata JSON matches the expression, then set the custom field value to the value of `<event>`. So, if the Faspex metadata is `{"Event":"Summer","_pkg_uuid":"abde20e28db24bfdb513c0a3de3bb8ff", "pkg_name":"test"...}`, the "faspex_event" custom field's value is `Summer`.

The expression is interpreted as follows:

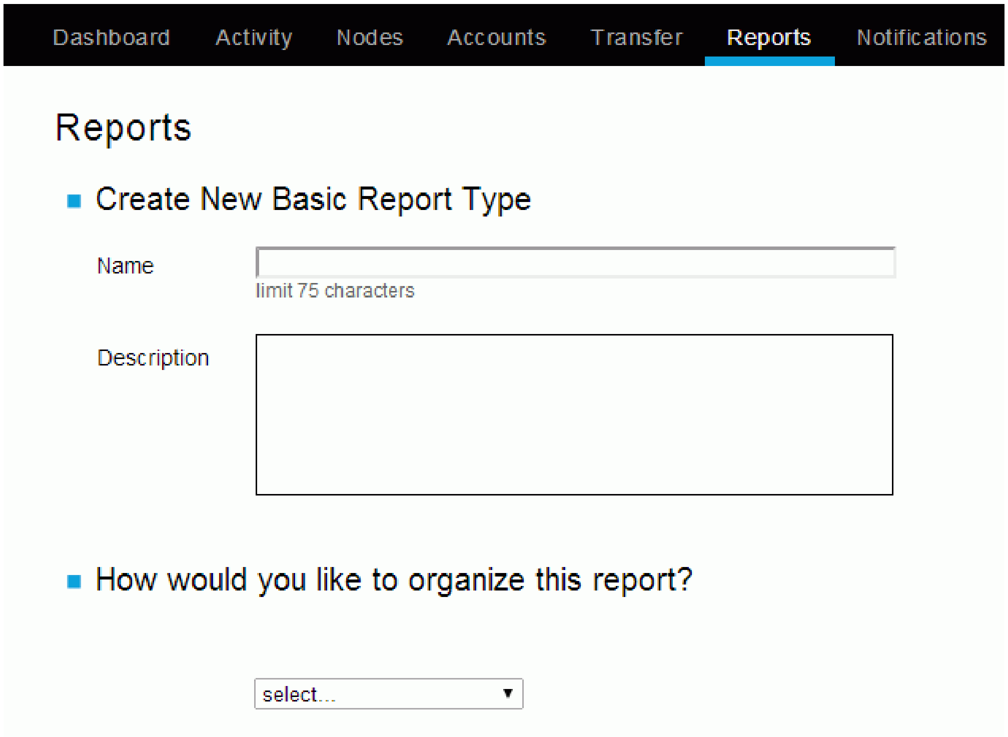
Expression	Interpretation	Characters matched in the example
`\{`	Finds a left curly bracket character	`{`
`.*`	Followed by 0 or more characters until the next piece of the expression (`"Event":"`)	
`"Event":"`	Followed by the first match of the exact text `"Event":"`	`"EVENT":"`
`(?<event>.+?)`	Followed by at least 1 character" (save these characters and call them `<event>`)	`Summer`
"	Followed by a quotation mark	"
`.*`	Followed by 0 or more characters until the next piece of the expression (`\{`)	`,"_pkg_uuid":"abde20e28db24bfdb513c0a3de3bb8ff", "pkg_name":"test"...`
`\}`	Followed by a right curly bracket character	`}`

4. Go to the *Manage Report Types* page.

   Select **Reports** from the Console menu, and then click the **Manage Report Types** button. On the *Manage Report Types* screen, click the **New Basic** button.

5. Configure your basic report to display contact, file-level, and transfer-level details, organized by Faspex metadata (the "event").

   On the *Create New Report Type* page (for **basic** reports), enter the following information:



Reports

Create New Basic Report Type

Name

limit 75 characters

Description

How would you like to organize this report?

select...

Field	Description
Name	Faspex meta data report
Description	Based on the custom field "event." Includes metadata, contact, file-level, and transfer-level details.
How would you like to organize this report?	Select "Event" (which is a custom field), "Contact," "Transfer" and "File" as the fields by which to organize this report. In doing so, the report will be grouped by the following: • **Event** (Based on a transfer-level rule that states if the conditions match the regular expression, then set the "event" custom field value to the Faspex metadata value.) • **Contact** (Contact assigned by Console. This can be a Console user name, a Faspex Server user name, SSH account, or customized value obtained from a transfer cookie. Examples: "admin (console)", "aspera (ssh)", "michael (faspex)".) • **Transfer** (Human-readable name assigned to a transfer. A transfer represents one or multiple executions of *ascp* (i.e., one or multiple attempts to transfer).) • **File** (File's name)
Columns to include	Select the following basic fields to include as columns: • started at • stopped at • bytes transferred • status • average rate • cookie **Note:** When you select a field, its definition will appear in the box below.

Field	Description
Sort	Select the following fields to sort data inside your groups: • Sort your metadata groups by event/metadata name • Sort your contact groups by contact name • Sort your transfer groups by transfer name • Sort your file groups by file name Select ascending order for all fields.
Filters	Filter the report so that only fields with metadata appear (that is, event *is not* NULL) and only data from *Faspex Users* is displayed (that is, transfer contact contains the value *faspex*).

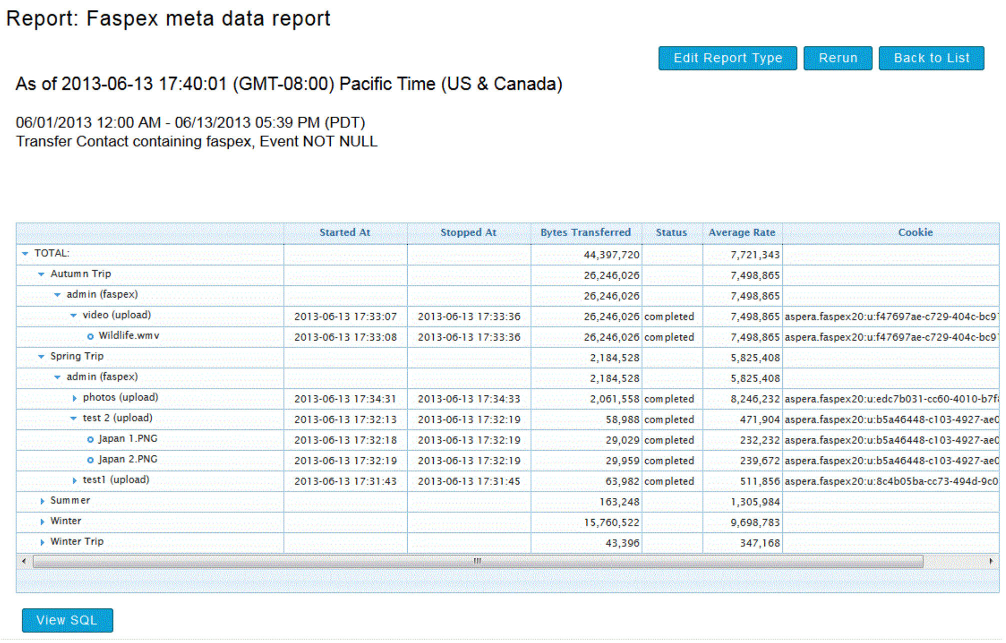
6. Save, finalize run settings and run your report.

   Next, click the **Create and Run** button. Confirm the following settings on next page:

   - Title is as described above.
   - Report is scheduled to *Run Now*.
   - Report period is *Month to date* and time zone is *Pacific*.
   - Sorting is as described above.
   - Filter is as described above.

   Once confirmed, click the **Run Report** button.
7. View your Web report.

   After clicking the **Run Report** button, the page will update to display the report queuing and then running. Once generated, the Web version of your basic report will appear as shown below.



Report: Faspex meta data report

As of 2013-06-13 17:40:01 (GMT-08:00) Pacific Time (US & Canada)

06/01/2013 12:00 AM - 06/13/2013 05:39 PM (PDT)
Transfer Contact containing faspex, Event NOT NULL

	Started At	Stopped At	Bytes Transferred	Status	Average Rate	Cookie
TOTAL:			44,397,720		7,721,343	
Autumn Trip			26,246,026		7,498,865	
admin (faspex)			26,246,026		7,498,865	
video (upload)	2013-06-13 17:33:07	2013-06-13 17:33:36	26,246,026	completed	7,498,865	aspera.faspex20:u:f47697ae-c729-404c-bc9
Wildlife.wmv	2013-06-13 17:33:08	2013-06-13 17:33:36	26,246,026	completed	7,498,865	aspera.faspex20:u:f47697ae-c729-404c-bc9
Spring Trip			2,184,528		5,825,408	
admin (faspex)			2,184,528		5,825,408	
photos (upload)	2013-06-13 17:34:31	2013-06-13 17:34:33	2,061,558	completed	8,246,232	aspera.faspex20:u:edc7b031-cc60-4010-b7f
test 2 (upload)	2013-06-13 17:32:13	2013-06-13 17:32:19	58,988	completed	471,904	aspera.faspex20:u:b5a46448-c103-4927-ae0
Japan 1.PNG	2013-06-13 17:32:18	2013-06-13 17:32:19	29,029	completed	232,232	aspera.faspex20:u:b5a46448-c103-4927-ae0
Japan 2.PNG	2013-06-13 17:32:19	2013-06-13 17:32:19	29,959	completed	239,672	aspera.faspex20:u:b5a46448-c103-4927-ae0
test1 (upload)	2013-06-13 17:31:43	2013-06-13 17:31:45	63,982	completed	511,856	aspera.faspex20:u:8c4b05ba-cc73-494d-9c0
Summer			163,248		1,305,984	
Winter			15,760,522		9,698,783	
Winter Trip			43,396		347,168	

As you can see, the report's data is grouped and sorted in the following manner:

- Metadata
- Faspex Users that selected the corresponding event/metadata
- Transfers (per Faspex User), which are sorted by the time they started
- File name (per Transfer)

In addition, all data columns appear as selected on the *Create Basic Report Type* page.

Advanced Report Example: Transfer Sessions with High Packet Loss

The following example demonstrates the process of creating a new, **advanced** report (following the instructions described in the topic "Creating an Advanced Report" on page 77) for transfers with high packet loss. In our example, we will generate a report that displays a list of all transfers that have high packet loss, where high loss is user specified. The report includes transfers that started before the report period start, as well as ones that ended after the report period end, as long as part of the transfer fell within the reporting period. Note that the data is not prorated, meaning that the "bytes transferred," "files complete" and other values show totals for the entire transfer, even if part of the transfer took place outside the reporting period.

1. Go to the *Manage Report Types* page.

   Select **Reports** from the Console menu, and then click the **Manage Report Types** button. On the *Manage Report Types* screen, click the **New Advanced** button.
2. Input your advanced report's name and description.

   On the *Create New Advanced Report Type* page, enter the following information:

Field	Description
Name	Transfer Sessions with High Packet loss
Description	Displays a list of all transfers that have high packet loss.
3. Create your SQL script.

   Important: For assistance on SQL variables and a fields reference guide, please click the *Help* link.

```
CREATE TABLE $FINAL_RESULT_TABLE

SELECT DISTINCT -- prevents duplicate rows (that is, overlapping permissions)
  ts.name
  , ts.contact
  , ts.bytes_transferred
  , ts.bytes_lost
  , TRUNCATE((ts.bytes_lost)*100/(ts.bytes_transferred + ts.bytes_lost), 1) AS `packet loss
%`
  , ts.source_ip AS `from`
  , ts.dest_ip AS `to`
  , ts.started_at
  , ts.stopped_at
  , ts.status
  , ts.files_complete
  , ts.files_failed
  , ts.files_skipped

FROM
  $TBL_TRANSFER_SESSIONS ts

WHERE
 ((ts.bytes_lost * 100) /(ts.bytes_lost + ts.bytes_transferred)) >= $PACKET_LOSS /* Custom/
configurable variable */
  AND
   ts.started_at < '$REPORT_PERIOD_END'
   AND (
    ts.stopped_at >= '$REPORT_PERIOD_START'
    OR ts.stopped_at IS NULL
  )

ORDER BY
  5 DESC
  , 8
;
```

   Important: For demonstration purposes, we have created a configurable/custom variable called $PACKET_LOSS in the SQL script text above. You may, alternatively, utilize the built-in SQL database field *avg_loss_pct*, to display the average packet loss over the network (as a percentage). Please see the *Help* link in the application for details.

4. Save, finalize run settings and run your report.

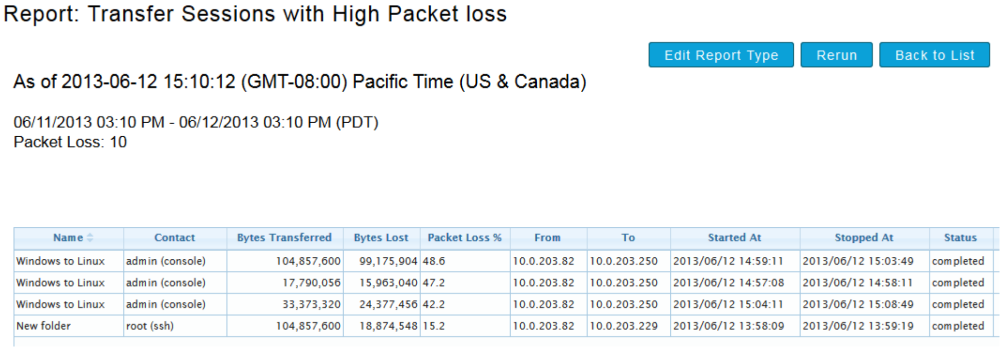
   Next, click the **Create and Run** button. Confirm the following settings on next page:

   - Title is as described above.
   - Report is scheduled to *Run Now*.
   - Report period is *Last 24 hours* and time zone is *Pacific*.

   Once confirmed, click the **Run Report** button.

5. View your Web report.

   After clicking the **Run Report** button, the page will update to display the report queuing and then running. Once generated, the Web version of your basic report will appear as shown below.



Report: Transfer Sessions with High Packet loss

As of 2013-06-12 15:10:12 (GMT-08:00) Pacific Time (US & Canada)

06/11/2013 03:10 PM - 06/12/2013 03:10 PM (PDT)
Packet Loss: 10

Name	Contact	Bytes Transferred	Bytes Lost	Packet Loss %	From	To	Started At	Stopped At	Status
Windows to Linux	admin (console)	104,857,600	99,175,904	48.6	10.0.203.82	10.0.203.250	2013/06/12 14:59:11	2013/06/12 15:03:49	completed
Windows to Linux	admin (console)	17,790,056	15,963,040	47.2	10.0.203.82	10.0.203.250	2013/06/12 14:57:08	2013/06/12 14:58:11	completed
Windows to Linux	admin (console)	33,373,320	24,377,456	42.2	10.0.203.82	10.0.203.250	2013/06/12 15:04:11	2013/06/12 15:08:49	completed
New folder	root (ssh)	104,857,600	18,874,548	15.2	10.0.203.82	10.0.203.229	2013/06/12 13:58:09	2013/06/12 13:59:19	completed

Aspera Ecosystem Security Best Practices

Your Aspera applications can be configured to maximize system and content security. The following sections describe the recommended settings and practices that best protect your content when using IBM Aspera High-Speed Transfer Server and IBM aspera High-Speed Transfer Endpoint, IBM Aspera Faspex,IBM Aspera Shares, and IBM Aspera Console.

Contents

Securing the Systems that Run Aspera Software

The systems that run Aspera software can be secured by keeping them up to date, by applying security fixes, and by configuring them using the recommended settings.

Updates

Aspera continually improves the built-in security of its products, as do the producers of third-party components used by Aspera, such as Apache, Nginx, and OpenSSH. One of the first lines of defense is keeping your products up to date to ensure that you are using versions with the latest security upgrades:

- Keep your operating system up to date.
- Keep your Aspera products up to date.
- If using, keep OpenSSH up to date. The server security instructions require that OpenSSH 4.4 or newer (Aspera recommends 5.2 or newer) is installed on your system in order to use the `Match` directive. `Match` allows you to selectively override certain configuration options when specific criteria (based on user, group, hostname, or address) are met.
- If you are using the HSTS web UI, keep Apache serverIIS up to date.

Security Fixes

Rarely, security vulnerabilities are detected in the operating systems and third-party components that are used by Aspera. Aspera publishes security bulletins immediately that describe the affected products and recommended remediation steps.

Security Configuration

Recommended security settings vary depending on the products you are using and how they interact. See the following subsections for your Aspera products.

HSTS

1. Configure your SSH Server.

   Aspera recommends that you:

   - Open TCP/33001 and keep TCP/22 open until users are notified that they should switch to TCP/33001.
   - Once users are notified, block TCP/22 and allow traffic only on TCP/33001.

   The following steps open TCP/33001 and block TCP/22.

   a) Open the SSH configuration file.

   ```
   C:\Program Files\Aspera\\etc\sshd_config
   ```

   ```
   /etc/ssh/sshd_config
   ```

   If you do not have an existing configuration for OpenSSH, or need to update an existing one, Aspera recommends the following reference: https://wiki.mozilla.org/Security/Guidelines/OpenSSH.

   b) Change the SSH port from TCP/22 to TCP/33001.

   Add TCP/33001 and comment out TCP/22 to match the following example:

   ```
   #Port 22
   Port 33001
   ```

   HSTS admins must also update the `SshPort` value in the `<WEB...>` section of `aspera.conf`.

   Once this setting takes effect:

   - Aspera clients must set the TCP port to 33001 when creating connections in the GUI or specify **`-P 33001`** for command line transfers.
   - Server administrators should use `ssh -p 33001` to access the server through SSH.

   c) Disable non-admin SSH tunneling.

   SSH tunneling can be used to circumvent firewalls and access sensitive areas of your company's network. Add the following lines to the end of `sshd_config` (or modify them if they already exist) to disable SSH tunneling:

   ```
   AllowTcpForwarding no
   Match Group Administrators
   AllowTcpForwarding yes
   ```

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

   Depending on your `sshd_config` file, you might have additional instances of `AllowTCPForwarding` that are set to the default `Yes`. Review your `sshd_config` file for other instances and disable if necessary.

Disabling TCP forwarding does not improve security unless users are also denied shell access, because with shell access they can still install their own forwarders. Aspera recommends assigning users to aspshell, described in the following section.

d) Disable password authentication and enable public key authentication.

Public key authentication provides a stronger authentication method than passwords, and can prevent brute-force SSH attacks if all password-based authentication methods are disabled.

Important: Before proceeding:

- Create a public key and associate it with a transfer user, otherwise clients have no way of connecting to the server.
- Configure at least one non-root, non-transfer user with a public key to use to manage the server. This is because in the following steps, root login is disabled and transfer users are restricted to aspshell, which does not allow interactive login. This user and public key is what you use to access and manage the server as an administrator.

Add or uncomment `PubkeyAuthentication yes` and comment out `PasswordAuthentication yes`:

```
PubkeyAuthentication yes
#PasswordAuthentication yes
PasswordAuthentication no
```

Note: If you choose to leave password authentication enabled, be sure to advise account creators to use strong passwords and set `PermitEmptyPasswords` to "no".

```
PermitEmptyPasswords no
```

e) Disable root login.

⚠ **CAUTION:** This step disables root access. Make sure that you have at least one user account with sudo privileges before continuing, otherwise you may not have access to administer your server.

Comment out `PermitRootLogin yes` and add `PermitRootLogin No`:

```
#PermitRootLogin yes
PermitRootLogin no
```

f) Restart the SSH server to apply new settings. Restarting your SSH server does not affect currently connected users.

Click **Start > Control Panel > Administrative Tools > Services**. Locate the `OpenSSH Service` and click **Restart**.

```
# systemctl restart sshd.service
```

or for Linux systems that use **`init.d`**:

```
# service sshd restart
```

g) Review your logs periodically for attacks.

For information on identifying attacks, see IBM Aspera IBM Aspera High-Speed Transfer Server Admin Guide: Securing Your SSH Server.

2. For Aspera servers on Windows in an Active Directory Domain, create the Active Directory user account to use as the Aspera service account before installing your Aspera server software. This ensures that the correct security settings are applied to the user. If you create the Active Directory service account user after installation, see the following knowledge base article for instructions on how to configure security policies for the acount.

https://support.asperasoft.com/hc/en-us/articles/216125388-OpenSSH-in-Active-Directory-Environments

3. Configure your server's firewall to permit inbound access to only Aspera-required ports.

   Aspera requires inbound access on the following ports:

   - For SSH connections that are used to set up connections, TCP/33001.
   - For FASP transfers, UDP/33001 (or a range, see below).
   - If you use HTTP and HTTPS fallback with HSTS, TCP/8080 and TCP/8443. If you only use HTTPS, only open TCP/8443.
   - If your clients access the HSTS web UI, TCP/80 (for HTTP) or TCP/443 (for HTTPS).

4. For HSTS, require strong TLS connections to the web server.

   TLS 1.0 and TLS 1.1 are vulnerable to attack. Run the following command to require that the client's SSL security protocol be TLS version 1.2 or higher:

   ```
   > /opt/aspera/bin/asconfigurator -x "set_server_data;ssl_protocol,tlsv1.2"
   ```

5. If Aspera Node D is exposed to internet traffic, run it behind a reverse proxy.

   If your Aspera server must expose Aspera Node D to the internet, such as when setting it up as a IBM Aspera on Cloud (AoC) node, Aspera strongly recommends protecting it with a reverse proxy. Normally, Aspera Node D runs on port 9092, but nodes that are added to AoC must have Aspera Node D run on port 443, the standard HTTPS port for secure browser access. Configuring a reverse proxy in front of Aspera Node D provides additional protection (such as against DOS attacks) and resource handling for requests to the node's 443 port.

   The following instructions describe how to set up Nginx as a reverse proxy and require that you have valid, CA-signed SSL certificates in `.pem` format for the server. Other reverse proxies might be supported on your server.

   a) Set up a system user with Node API credentials on your server.

   b) Download and install Nginx.

   c) Configure the HTTPS port for Aspera Node D.

   ```
   # asconfigurator -x "set_server_data;https_port,9092"
   ```

   d) Open the Nginx configuration file in a text editor.

   Open `C:\nginx\conf\nginx.conf` and ensure the following `include` directive is present in the `http` section. If it is not present, add it to the file:

   ```
   http {
   …
   include /etc/nginx/conf.d/*.confC:\nginx\*.conf;
   }
   ```

   e) Create a file named `aspera_node_proxy.conf` and save it in the following location:

   `C:\nginx\conf\sites-enabled\aspera_node_proxy.conf`

   Create the `sites-enabled` folder if it does not exist.

   f) Paste the following content into `aspera_node_proxy.conf`:

   ```
   #
   # Aspera configuration - reverse proxy for asperanoded
   #
   server {
           listen 443;
           server_name your.servername.com;
           ssl_certificate "C:/Program Files/Aspera/Enterprise Server/etc/
   aspera_server_cert.pem";
           ssl_certificate_key "C:/Program Files/Aspera/Enterprise Server/etc/
   aspera_server_key.pem";

           ssl on;
           ssl_session_cache builtin:1000 shared:SSL:10m;
           ssl_protocols TLSv1.2;
           ssl_ciphers EECDH+AESGCM:EDH+AESGCM:AES256+EECDH:AES256+EDH;
           ssl_prefer_server_ciphers on;
   ```

```
        access_log          C:\Logs\nginx\node-api.access.log;

        location / {
            proxy_pass https://127.0.0.1:9092;
            proxy_read_timeout 60;
            proxy_redirect https://127.0.0.1:9092 https://your.servername.com;

            proxy_set_header Host               $host:$server_port;
            proxy_set_header X-Real-IP          $remote_addr;
            proxy_set_header X-Forwarded-For    $proxy_add_x_forwarded_for;
            proxy_set_header X-Forwarded-Proto  $scheme;
        }
}
```

Note: Configure SSL ciphers as required. The preceding sample is not configured for backwards compatibility, and the recommended list of secure ciphers might change. Aspera recommends reviewing and staying current with the list provided in https://cipherli.st/.

In this configuration, Nginx listens externally on port 443, not 9092. Replace *your.servername.com* with your server's domain name.

g) Restart Aspera Node D.

```
# systemctl restart asperanoded
```

or for Linux systems that use **init.d**:

```
# service asperanoded restart
```

```
> net stop asperanoded
> net start asperanoded
```

h) Restart Nginx.

```
> nginx -s reload
```

i) Run Nginx as a Windows service.

When you install Nginx on a Windows OS, it is installed as an application that runs only when the user who installed the application is logged in. However, for HSTS nodes that are added to AoC, Aspera recommends running Nginx as a service so that its function is not tied to a specific user.

For instructions, see https://ibm.ibmaspera.com/helpcenter/admin/nodes/configuring-an-aspera-transfer-server-as-a-node-for-aspera-on-cloud.

6. Install Aspera FASP Proxy in a DMZ to isolate your HSTS from the Internet.

   For more information, see IBM Aspera FASP Proxy Admin Guide

Faspex and Shares

1. Configure your Faspex or Shares server firewall to allow inbound access to TCP/443, the default HTTPS port.
2. Faspex and Shares transfer nodes should be configured as described for HSTS.

   The transfer user that is used by Faspex and Shares (usually `xfer`) must be configured on the node to only allow transfers with a token:

```
> asconfigurator -x "set_user_data;user_name,xfer;authorization_transfer_in_value,token"
> asconfigurator -x "set_user_data;user_name,xfer;authorization_transfer_out_value,token"
```

   Set the token encryption key to a string of at least 20 characters:

```
> asconfigurator -x "set_user_data;user_name,xfer;token_encryption_key,token_string"
```

   Do not use UUIDs for this key because they might not be generated using cryptographically secure methods.

Console

Configure the firewall of the computer on which Console is installed to only allow Aspera-required connections to the following ports:

- For HTTP or HTTPS access for the web UI, inbound TCP/80 or TCP/443.
- For SSH connections, outbound TCP/33001 to managed nodes.
- For Node API connections, outbound TCP/9092 to managed nodes.
- For connections to legacy nodes (those running HSTS older than 3.4.6), outbound TCP/40001 and inbound TCP/4406. For security and reliability, Asepra strongly recommends upgrading all nodes to the latest version.

Securing the Aspera Applications

Your Aspera products can be configured to limit the extent to which users can connect and interact with the servers. The instructions for Shares 1.9.x and Shares 2.x are slightly different; see the section for your version.

HSTS

1. Restrict user permissions with **aspshell**.

   By default, all system users can establish a FASP connection and are only restricted by file permissions. Restrict the user's file operations by assigning them to use **aspshell**, which permits only the following operations:

   - Running Aspera uploads and downloads to or from this computer.
   - Establishing connections between Aspera clients and servers.
   - Browsing, listing, creating, renaming, or deleting contents.

   These instructions explain one way to change a user account or active directory user account so that it uses the **aspshell**; there may be other ways to do so on your system.

   Windows users are assigned to **aspshell** automatically when you configure the user in the GUI and specify a non-empty docroot. If you do not specify a docroot or configure users from the command line, you must manually set the users' shell as **aspshell.exe** in `C:\Program Files\Aspera\\etc\passwd`.

   Run the following command to change the user login shell to **aspshell**:

   ```
   > sudo usermod -s /bin/aspshell username
   ```

   Confirm that the user's shell updated by running the following command and looking for `/bin/aspshell` at the end of the output:

   ```
   > grep username /etc/passwd
   username:x:501:501:...:/home/username:/bin/aspshell
   ```

   Note: If you use OpenSSH, sssd, and Active Directory for authentication: To make `aspshell` the default shell for all domain users, first set up a local account for server administration because this change affects all domain users. Then open `/etc/sssd/sssd.conf` and change `default_shell` from `/bin/bash` to `/bin/aspshell`.
2. Restrict Aspera transfer users to a limited part of the server's file system or bucket in object storage.

   a) For on-premises servers, set a default docroot to an empty folder, then set a docroot for each user:

   ```
   > asconfigurator -x "set_node_data;absolute,docroot"
   > asconfigurator -x "set_user_data;user_name,username;absolute,docroot"
   ```

   Replace *username* with the username and *docroot* with the directory path to which the user should have access.

b) For cloud-based servers, set a default restriction to an empty folder, then set a restriction for each user:

```
> asconfigurator -x "set_node_data;file_restriction,|storage_path"
> asconfigurator -x "set_user_data;user_name,username;file_restriction,|storage_path"
```

Replace *username* with the username and *storage_path* with the path to which the user has access. Restriction syntax is specific to the storage:

Storage Type	**Format Example**
local storage	`file:////*file:///c%3A/Documents/*`
S3 and IBM Cloud Object Storage	`s3://*`
Swift storage	`swift//*`
Azure storage	`azu://*`
Azure Files	`azure-files://*`
Google Cloud Storage	`gs://*`
Hadoop (HDFS)	`hdfs://*`

The "|" is a delimiter, and you can add additional restrictions. For example, to restrict the system user `xfer` to `s3://s3.amazonaws.com/bucket_xyz/folder_a/*` and not allow access to key files, run the following command:

```
> asconfigurator -x "set_user_data;user_name,xfer;file_restriction,|s3://s3.amazonaws.com/
bucket_xyz/folder_a/*|!*.key"
```

3. Restrict users' read, write, and browse permissions.

   Users are given read, write, and browse permissions to their docroot by default. Change the global default to deny these permissions:

```
> asconfigurator -x "set_node_data;read_allowed,false;write_allowed,false;dir_allowed,false"
```

   Run the following commands to enable permissions per user, as required:

```
> asconfigurator -x "set_user_data;user_name,username;read_allowed,false"
> asconfigurator -x "set_user_data;user_name,username;write_allowed,false"
> asconfigurator -x "set_user_data;user_name,username;dir_allowed,false"
```

4. Limit transfer permissions to certain users.

   Set the default transfer permissions for all users to deny:

```
> asconfigurator -x "set_node_data;authorization_transfer_in_value,deny"
> asconfigurator -x "set_node_data;authorization_transfer_out_value,deny"
```

   Allow transfers for specific users by running the following commands for each user:

```
> asconfigurator -x "set_user_data;user_name,username;authorization_transfer_in_value,allow"
> asconfigurator -x "set_user_data;user_name,username;authorization_transfer_out_value,allow"
```

   Note: For a user that is used by Shares or Faspex (usually `xfer`), allow transfers only with a token by setting `authorization_transfer_{in|out}_value` to `token`.

5. Encrypt transfer authorization tokens.

   When a client requests a transfer from a server through an Aspera web application, an authorization token is generated. Set the encryption key of the token for each user or group on the server:

```
> asconfigurator -x "set_user_data;user_name,username;token_encryption_key,token_string"
> asconfigurator -x "set_group_data;group_name,groupname;token_encryption_key,token_string"
```

The token string should be at least 20 random characters.

Note: This is not used to encrypt transfer data, only the authorization token.

6. Require encryption of content in transit.

   Your server can be configured to reject transfers that are not encrypted, or that are not encrypted with a strong enough cipher. Aspera recommends setting an encryption cipher of at least AES-128. AES-192 and AES-256 are also supported but result in slower transfers. Run the following command to require encryption:

   ```
   > asconfigurator -x "set_node_data;transfer_encryption_allowed_cipher,aes-128"
   ```

   By default, your server is configured to transfer (as a client) using AES-128 encryption. If you require higher encryption, change this value by running the following command:

   ```
   > asconfigurator -x "set_client_data;transport_cipher,value"
   ```

   You can also specify the encryption level in the command line by using `-c cipher` with **ascp** and **async** transfers. **ascp4** transfers use AES-128 encryption.

7. Configure SSH fingerprinting for HSTS.

   For transfers initiated by a web application (such as Faspex, Shares, or Console), the client browser sends the transfer request to the web application server over an HTTPS connection. The web application requests a transfer token from the target server. The transfer is executed over a UDP connection directly between the client and the target server and is authorized by the transfer token. Prior to initiating the transfer, the client can verify the server's authenticity to prevent server impersonation and man-in-the-middle (MITM) attacks.

   To verify the authenticity of the transfer server, the web application passes the client a trusted SSH host key fingerprint of the transfer server. The client confirms the server's authenticity by comparing the server's fingerprint with the trusted fingerprint. In order to do this, the host key fingerprint must be set in the server's `aspera.conf`.

   Note: Server SSL certificate validation (HTTPS) is enforced if a fingerprint is specified in `aspera.conf` and HTTP fallback is enabled. If the transfer "falls back" to HTTP and the server has a self-signed certificate, validation fails. The client requires a properly signed certificate.

   If you set the host key path, the fingerprint is automatically extracted from the key file and you do not extract it manually.

   Retreiving and setting the host key fingerprint:

   a) Retrieve the server's SHA-1 fingerprint.

      ```
      > cat /etc/ssh/ssh_host_rsa_key.pub | awk '{print $2}' | base64 - | sha1sum
      ```

      On the server, run a local **ascp** transfer. The transfer does not need to complete successfully in order for the remote host-key fingerprint to appear in the log.

      ```
      > ascp source_file username@localhost:destination
      ```

      Open `C:\Program Files\Aspera\\var\log\aspera-scp-transfer.log`. Search for "remote host-key fingerprint". The line appears similar to the following, in which 19f7cf4d495234ng4342ha062f5d98b5a9d665 is the SHA-1 fingerprint:

      ```
      2017-12-08 12:04:53.024 [1888-0000264c] LOG [asssh] remote host-key fingerprint
      19f7cf4d495234ng4342ha062f5d98b5a9d665
      ```

   b) Set the SSH host key fingerprint in `aspera.conf`.

      ```
      > asconfigurator -x "set_server_data;ssh_host_key_fingerprint,fingerprint"
      ```

This command creates a line similar to the following example of the `<server>` section of `aspera.conf`:

```
<ssh_host_key_fingerprint>7qdOwebGGeDeN7Wv+2dP3HmWfP3
</ssh_host_key_fingerprint>
```

c) Restart the node service to activate your changes.

Go to **Control Panel > Administrative Tools > Services**, click **Aspera NodeD**, and click **Restart**.Run the following commands to restart `asperanoded`:

```
> systemctl restart asperanoded
```

or for Linux systems that use **`init.d`**:

```
> service asperanoded restart
```

Setting the host key path: To set the SSH host key path instead of the fingerprint, from which the fingerprint will be extracted automatically, run the following command:

```
# asconfigurator -x "set_server_data;ssh_host_key_path,ssh_key_filepath"
```

This command creates a line similar to the following in the `<server>` section of `aspera.conf`:

```
<ssh_host_key_path>/etc/ssh/ssh_host_rsa_key.pub
</ssh_host_key_path>
```

Restart the node service to activate your changes, as described for "Retreiving and setting the host key fingerprint".

8. Install properly signed SSL certificates.

   Though your Aspera server automatically generates self-signed certificates, Aspera recommends installing valid, signed certificates. These are required for some applications.

Faspex

Many of the settings for Faspex are the same as for HSTS, including SSH server configuration, firewall settings, and signed SSL certificate installation. The following recommendations augment or are additional to the recommendations described for HSTS.

1. Restrict transfers by all users except "faspex".

   If your system is a dedicated Faspex server - the HSTS installed as part of your Faspex installation is used only for Faspex transfers - prohibit transfers by all users except "faspex". If you have not already, deny transfers globally by default:

```
> asconfigurator -x "set_node_data;authorization_transfer_in_value,deny"
> asconfigurator -x "set_node_data;authorization_transfer_out_value,deny"
```

   Allow transfers for "faspex" by running the following commands:

```
> asconfigurator -x "set_user_data;user_name,faspex;authorization_transfer_in_value,token"
> asconfigurator -x "set_user_data;user_name,faspex;authorization_transfer_out_value,token"
```

2. Configure the Nginx server to allow only strong TLS.

   The default configuration of Faspex has TLS 1.0, 1.1 and 1.2 enabled. Older browsers require the older and less secure versions (TLS 1.0 and 1.1). You can disable support for older browsers by removing TLS 1.0 and TLS 1.1 from the configuration.

   a) Open the Nginx configuration file on the Shares server for editing:

   `C:\Program Files\Common Files\Aspera\Common\apache\conf\extra/httpd-ssl.conf`

   b) Locate the following line:

```
SSLProtocol ALL -SSLv2 -SSLv3
```

   c) Replace the line with the following and save your change:

```
SSLProtocol TLSv1.2
```

   d) Restart Apache to activate your change:

```
> asctl apache:restart
```

3. Limit admin logins to those from known IP addresses.

   Faspex admins have the ability to execute post-processing scripts on the server. If an admin account is compromised, this capability can be a serious threat to your server's security. You can add additional protection by allowing admin logins from only specific IP addresses.

   a) In the Faspex UI, go to **Accounts** and select the admin account.

   b) In the **Permissions** section, locate the **Allowed IP addresses for login** field and enter the IP addresses or IP address range to allow.

   c) Click **Save** to activate your changes.

4. Configure Faspex account security settings.

   Go to **Server > Configuration > Security** and set the following global default configurations in the **Faspex accounts** section, then edit configurations for individual users, as needed:

   a) Set a non-zero session timeout.

   b) Lock users out after five failed login attempts within five minutes.

   c) Enable **Prevent concurrent login**.

   d) Set a password expiration interval of 30 days.

   e) Prevent reuse of the last three passwords and require strong passwords.

   f) Set **Keep user directory private** to **Yes**.

   g) Disable **Allow all users to send to all other Faspex users**.

   h) Disable **Users can see global distribution lists**.

   i) Disable **Ignore invalid recipients**.

   j) Disable **Allow users to change their email address**.

   Stay in **Server > Configuration > Security** for the next step.

5. Configure Faspex account registration settings.

   In **Server > Configuration > Security**, set the following configurations in the **Registrations** section:

   a) Set **Self-registration** to **None**.

   When self-registration is enabled, it can be used to find out whether a certain account exists on the server. That is, if you attempt to self-register a duplicate account, you receive a prompt stating that the user already exists.

   b) Select **Require external users to register**.

   By requiring external users to register, you can better track their Faspex activity.

   Stay in **Server > Configuration > Security** for the next step.

6. Configure outside email address settings.

   In **Server > Configuration > Security**, set the following global default configurations in the **Outside email addresses** section, then edit configurations for individual users, as needed:

   a) Disable **Allow inviting external senders**.

   b) Enable **Invitation link expires** and set an expiration policy.

   c) Disable **Allow public submission URLs**.

   d) Disable **Allow sending to external email addresses**.

   e) Set a package link expiration.

f) Disable **Allow external packages to Faspex users**.

Stay in **Server > Configuration > Security** for the next step.

7. Configure Faspex encryption.

   In **Server > Configuration > Security**, set the following configurations in the **Encryption** section:

   a) Enable **Encrypt transfers**.

   b) If possible in your work flow, set **Use encryption-at-rest** to **Always**.

   See the next section, "Securing Content in your Workflow," for information about encryption at rest.

   c) Disable **Allow dropboxes to have their own encryption settings**.
8. Click **Update** when you have completed updating settings on the **Security** page to activate your changes.
9. Hide your server's IP address from email notifications.

   If Faspex is configured to identify itself by IP address (rather than by domain name), then the URLs in your notification emails contain your IP address (for example, "https://10.0.0.1/aspera/faspex"). Configure an alternate IP address or domain name for users who are external to your organization.

   a) Go to **Server > Configuration > Web Server**.

   b) Select **Enable alternate address** then click **Add alternate address**.

   c) Enter the address name and description, and select **Show in emails**.

   d) Click **Update** to activate your change.

   e) Customize your email notification templates to use the alternate address.

   Go to **Server > Notifications**.

Shares

The Shares server and its nodes should be secured as described for HSTS, including SSH server configuration, firewall settings, and valid, signed SSL certificate installation. You can also secure the Shares application and its network of nodes by restricting user permissions. Set the following settings globally, then edit the settings for specific users and groups.

1. Configure Shares security settings.

   On the **Admin** page, click **User Security** and set the following:

   a) Set a non-zero session timeout.

   b) Require strong passwords.

   c) Set a password expiration interval of 30 days.

   d) Lock users out after five failed login attempts within five minutes.

   e) Do not allow self registration by setting **Self Registration** to **None**.
2. When setting up the email server (**Admin > SMTP**), select **Use TLS if available**.
3. Configure the Nginx server to allow only strong TLS.

   The default configuration of Shares has TLS 1.0, 1.1 and 1.2 enabled. Older browsers require the older and less secure versions (TLS 1.0 and 1.1). You can disable support for older browsers by removing TLS 1.0 and TLS 1.1 from the configuration.

   a) Open the Nginx configuration file on the Shares server for editing:

   `C:\Shares\nginx\conf\nginx.conf`

   b) Delete TLSv1 and TLSv1.1 from the following line:

   ```
   ssl_protocols TLSv1 TLSv1.1 TLSv1.2;
   ```
4. Configure secure transfer settings.

   Go to **System Settings > Transfers** and set the following:

   a) Require a minimum Connect version of 3.6.1.

   b) For **Encryption**, select **AES-128**.

c) If possible in your workflow, set **Encryption at Rest** to **Required**.

See the next section, "Securing Content in your Workflow," for information about encryption at rest.

5. Go to **System Settings > Web Server** and select **Enable SSL/TLS**.

   This setting requires that the Shares server has a valid, signed SSL certificate.
6. When adding new users to Shares, disable **API Login** if users do not need to use the Shares API.

   The Shares API is used by clients connecting through IBM Aspera Drive and IBM Aspera Command-Line Interface
7. When adding nodes to Shares, select **Use SSL** (required by Shares) and **Verify SSL** (requires that the node has a valid, signed SSL certificate).
8. When authorizing a user or group to a share (***share_name*** **> Authorizations**), set the minimum permissions required based on their Shares use.

Shares 2.x

The Shares 2.x server and its nodes should be secured as described for HSTS, including configuring the SSH server, firewall settings, and installing valid, signed SSL certificates. You can also secure the Share application and its network of nodes by restricting user permissions. Set the following settings globally and then edit the settings for specific users, groups, and administrators.

1. Configure Shares security settings.

   Go to **System Administration > Configuration > User Security** and set the following:

   a) Set a non-zero session timeout.

   b) Set an access token lifetime of 8 hours.

   c) Enable refreshing of expired access tokens, with a lifetime of 7 days.

   Go to **System Administration > Configuration > Local User Security** and set the following:

   a) Require strong passwords.

   b) Set a password expiration interval of 30 days.

   c) Lock users out after five failed login attempts within five minutes.

   d) Prevent reuse of the last three passwords and require strong passwords.
2. When setting up the email server (**System Administration > Configuration > SMTP**), select **Use TLS if available**.
3. Configure secure transfer settings.

   Go to **System Administration > Configuration > Transfers** and set the following:

   a) Require a minimum Connect version of 3.6.1.

   b) For **Encryption**, select **AES-128** (or higher, if needed).

   c) If possible in your workflow, set **Encryption at Rest** to **Yes**.

   See the next section, "Securing Content in your Workflow," for information about encryption at rest.
4. Go to **System Administration > Configuration > Web Server** and select **Enable SSL/TLS**.

   This setting requires that the Shares server has a valid, signed SSL certificate.
5. Configure the Nginx server to allow only strong TLS.

   The default configuration of Shares has TLS 1.0, 1.1 and 1.2 enabled. Older browsers require the older and less secure versions (TLS 1.0 and 1.1). You can disable support for older browsers by removing TLS 1.0 and TLS 1.1 from the configuration.

   a) Open the Nginx configuration file on the Shares server for editing:

   `C:\Shares\nginx\conf\nginx.conf`

   b) Delete TLSv1 and TLSv1.1 from the following line:

   ```
   ssl_protocols TLSv1 TLSv1.1 TLSv1.2;
   ```

6. When adding nodes to Shares, select **Use SSL** (required by Shares) and **Verify SSL** (requires that the node has a valid, signed SSL certificate).
7. When authorizing a user or group to a share, set the minimum permissions required based on their Shares use.

Console

Console nodes should be secured as described for HSTS, including SSH server configuration, firewall settings, and valid, signed SSL certificate installation. If possible for your workflow, limit Console and its nodes to your internal network.

You can also secure the Console application and its network of nodes by restricting user permissions:

1. Configure secure Console defaults.

   Go to **Configuration > Defaults** and set the following:

   a) In the drop-down menu for **Default SSH encryption**, select a default SSH encryption algorithm of at least AES-128 for non-Console nodes.

   b) For **Transport Encryption**, select **AES-128**.

   c) Disable **Smart Transfer Sharing**.

   d) Set a non-zero session timeout.

   e) Lock users out after five failed login attempts within five minutes.

   f) Enable **Prevent concurrent login**.

   g) Enable **Suppress logging of transfer tokens** to prevent tokens from being written to the Console database.

   h) Set a password expiration interval of 30 days.

   i) Prevent reuse of the last three passwords and require strong passwords.
2. When setting up the email server (**Notifications > Email Server**), select **Use TLS if available**.
3. Restrict Console users' permissions.

   a) When creating a new user (**Accounts > Users > New User**), disable user login until their permissions are set by clearing **Active (allow user to log in)**. Click **permissions** and enable only the permissions that the user requires. Once permissions are configured, allow the user to login by going to **Accounts > Users**, clicking the user, and selecting **Active (allow user to log in)**.

   b) Assign users to Console Groups with only the required transfer paths and permissions allowed.

   Create a group (**Accounts > Groups > New Group**) and restrict the group's transfers by clicking **Add Transfer Path**. Assign specific endpoints to the group's transfer path, rather than **Any**, which grants permission to transfer to all nodes. Limit the direction of the path, if the group's workflow allows.
4. When adding managed and unmanaged nodes, set the SSH port to 33001 and ensure SSH connections are encrypted with AES-128 or higher.
5. When adding a managed cluster, select **Use HTTPS to connect to node** and **Require signed SSL certificate**.
6. When adding SSH endpoints, use SSH public key authentication rather than password authentication.

   The key file on the node should not be a shared key; it should be a "private" key in the specified user account.

Securing Content in your Workflow

1. If your workflow allows, enable server-side encryption-at-rest (EAR).

   When files are uploaded from an Aspera client to the Aspera server, server-side encryption-at-rest (EAR) saves files on disk in an encrypted state. When downloaded from the server, server-side EAR first decrypts files automatically, and then the transferred files are written to the client's disk in an unencrypted state. Server-side EAR provides the following advantages:

- It protects files against attackers who might gain access to server-side storage. This is important primarily when using NAS storage or cloud storage, where the storage can be accessed directly (and not just through the computer running HSTS).
- It is especially suited for cases where the server is used as a temporary location, such as when one client uploads a file and another client downloads it.
- Server-side EAR can be used together with client-side EAR. When used together, content is doubly encrypted.
- Server-side EAR doesn't create an "envelope" as client-side EAR does. The transferred file stays the same size as the original file. The server stores the metadata necessary for server-side EAR separately in a file of the same name with the file extension `.aspera-meta`. By contrast, client-side EAR creates a envelope file containing both the encrypted contents of the file and the encryption metadata, and it also changes the name of the file by adding the file extension `.aspera-env`.)
- It works with both regular transfers (FASP) and HTTP fallback transfers.

Limitations and Other Considerations

- Server-side EAR is not designed for cases where files need to move in an encrypted state between multiple computers. For that purpose, client-side EAR is more suitable: files are encrypted when they first leave the client, then stay encrypted as they move between other computers, and are decrypted when they reach the final destination and the passphrase is available. See Step 4 of this section for more information on client-side encryption.
- Do not mix server-side EAR and non-EAR files in transfers, which can happen if server-side EAR is enabled after the server is in use or if multiple users have access to the same area of the file system but have different EAR configurations. Doing so can cause problems for clients by overwriting files when downloading or uploading and corrupting metadata.
- Server-side EAR does not work with multi-session transfers (using **`ascp -C`** or node API `multi_session` set to greater than 1) or Watch Folders (versions prior to 3.8.0 that do not support URI docroots).

To enable server-side EAR:

a) Set users' docroots in URI format (local docroots are prepended with `file:///`).

```
> asconfigurator -x "set_user_data;user_name,username;absolute,file:///path"
```

b) Set the server-side EAR password.

Set a different EAR password for each user:

```
> asconfigurator -x
"set_user_data;user_name,username;transfer_encryption_content_protection_secret,passphrase"
```

Important: If the EAR password is lost or `aspera.conf` is compromised, you cannot access the data on the server.

c) Require content protection and strong passwords.

These settings cause server-side EAR to fail if a password is not given or if a password is not strong enough. For example, the following **`asconfigurator`** command adds both these options for all users (global):

```
> asconfigurator -x "set_node_data;transfer_encryption_content_protection_required,true"
> asconfigurator -x "set_node_data;transfer_encryption_content_protection_strong_pass_required,true"
```

2. Never use "shared" user accounts.

   Configure each user as their own Aspera transfer user. Sharing Aspera transfer user account credentials with multiple users limits user accountability (you cannot determine which of the users sharing the account performed an action).

3. Use passphrase-protected private keys.

   The **`ssh-keygen`** tool can protect an existing key or create a new key that is passphrase protected.

If you cannot use private key authentication and use password authentication, use strong passwords and change them periodically.

4. If your workflow allows, require client-side encryption-at-rest (EAR).

   Aspera clients can set their transfers to encrypt content in transit and on the server, and the server can be configured to require client-side EAR. You can combine client-side and server-side EAR, in which case files are doubly encrypted on the server. Client-side encryption-at-rest is not supported for **ascp4** or **async** transfers.

   Client configuration

   The client specifies a password and the files are uploaded to the server with a `.aspera-env` extension. Anyone downloading these `.aspera-env` files must have the password to decrypt them. Users can enable client-side EAR in the GUI or on the **ascp** command line.

   GUI: Go to **Connections > *connection_name* > Security**. Select **Encrypt uploaded files with a password** and set the password. Select **Decrypt password-protected files downloaded** and enter the password.

   Ascp command line: Set the encryption and decryption password as the environment variable ASPERA_SCP_FILEPASS. For uploads (`--mode=send`), use `--file-crypt=encrypt`. For downloads (`--mode=recv`), use `--file-crypt=decrypt`.

   Note: When a transfer to HSTS falls back to HTTP or HTTPS, client-side EAR is no longer supported. If HTTP fallback occurs while uploading, then the files are NOT encrypted. If HTTP fallback occurs while downloading, then the files remain encrypted.

   Server configuration

   To configure the server to require client-side EAR and to require strong content protection passwords, run the following commands:

   ```
   > asconfigurator -x "set_node_data;transfer_encryption_content_protection_required,true"
   > asconfigurator -x "set_node_data;transfer_encryption_content_protection_strong_pass_required,true"
   ```

   Note: These commands set the global configuration. Depending on your work flow, you might want to require client-side EAR and strong passwords for only specific users.

5. For particularly sensitive content, do not store unecrypted content on any computer with network access.

   HSTS, HSTE, and Desktop Client include the **`asprotect`** and **`asunprotect`** command-line tools that can be used to encrypt and decrypt files. Use an external drive to physically move encrypted files between a network-connected computer and an unconnected computer on which the files can be unencrypted.

   - To encrypt a file before moving it to a computer with network access, run the following commands to set the encryption password and encrypt the file:

     ```
     > setexport ASPERA_SCP_FILEPASS=password
     > /opt/aspera/bin/asprotect -o filename.aspera-env filename
     ```

   - To download client-side-encrypted files without decrypting them immediately, run the transfer without decryption enabled (clear **Decrypt password-protected files downloaded** in the GUI or do not specify `--file-crypt=decrypt` on the **ascp** command line).
   - To decrypt encrypted files, run the following commands to set the encryption password and decrypt the file:

     ```
     > setexport ASPERA_SCP_FILEPASS=password
     > /opt/aspera/bin/asprotect -o filename filename.aspera-env
     ```

IBM®